University of Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: Computational Engineering	Spring/ Autumn semester, 2021...... Open

Author: Megh Raj Upreti

Programme Coordinator: Aksel Hiorth

Supervisors: Aksel Hiorth(Internal) and Kjartan Kloster Osmundsen(External)

Title of master's thesis: Predicting appointment attendance at an outpatient clinic

Credits: 30

Keywords: Hospital Missed Appointments, Show, no-show, Classification models, Machine learning, C_kontaktAvsluttkodeNavn, Binary Classification, Precision and Recall	Number of pages: 63 Pages + supplemental material/other: 4 pages Stavanger 15.06.2021 date/year

[This page is intentionally left blank]

Abstract

Background and aim: No-show, refers to patients who do not show up in the scheduled appointment, which is a serious problem for many years in the hospital. Also, the no-show rates differ in countries because of the different care systems and medical facilities. This study aims to understand the characteristics of the patient who miss the appointments, analyze their data, and come up with suitable prediction models.

Methods: The study consists of primary research. The main research is based on quantitative data that aims to analyze Helse vest data and compare different binary classifiers and choose the best model that fits the hospital's business objective. CRISP-DM Methodology was preferred for the data mining process. The dataset consists of 61 columns and for data privacy reasons Age and date were not provided. Also, an SMS reminder sent was not available to use for the analysis due to GDPR.

Results: In the data analysis part, 9.44% missed the appointments(Ikke møtt/Ingen beskjed), 10.57% cancel the appointment and 79.97% came for the appointment(Ordinært avsluttet). The result for analysis shows that young people aged 20-29 missed most appointments.

Regarding days, Monday was the top-performing day for scheduling appointments. Thursday is the most missed appointment day and November is the most missed appointment month whereas least July is the least missed appointment. Precision was highest for Random Forest with 89.56% correctly predicted as a show and the values were actual show, whereas recall was highest for GaussianNB which was 80.87% which means out of all show, 80.87% was predicted show.

Conclusion: Patients aged 20-29 missed most appointments. Random Forest is better among all in terms of distinguishing between shows and no shows. However, solving this issue fully is challenging. In a nutshell, it would be well served to develop an understanding of the situation and business objectives under which each evaluation metric should be utilized.

Keywords

Hospital Missed Appointments, Show, no-show, Classification models, Machine learning, C_kontaktAvsluttkodeNavn, Binary Classification, Precision and Recall

Acknowledgments

This is the thesis of my master’s degree in Computational Engineering at the University of Stavanger, which has transpired over 4.5 months during the spring of 2021. The thesis is in the field of Missed Appointments and presents exploratory data analysis of medical data and performing prediction on binary classifiers on the same data.

I would like to express my gratitude to my external supervisor Kjartan Kloster Osmundsen for all information, comments, feedbacks, and engagement he has contributed through the learning process of this master thesis. I would also like to thank my internal supervisor at the University of Stavanger, Professor Aksel Hiorth, for comments, motivation, and feedback during the process.

I would like to give special thanks to Helse vest and other members involved in data collection, guidance, and for providing data and as it will not be possible to perform analysis and predictions without data access. Therefore, I appreciate Helse vest willingness to provide me with data.

Furthermore, I would like to thank my wife Rijan Thapa for her full support in my studies and for giving wonderful tips for motivation. Finally, is my goal reached. I will forever be grateful for your love. I am also grateful for all the information and support Helse Vest has provided during this whole period. Always, I would like to thank my Family: Jaya Prasad Upreti, Kamala Upreti, and Yubraj Upreti, and Pratima Dahal.

Meg Raj Upreti

Contents

Table of Figures

Listings:

List of tables:

List of Acronyms:

GDPR: General Data Protection Regulation
SMS: Short Message Service
AUC: Area Under Curve
TP: True Positive
TN: True Negative
FP: False Positive
FN: False negative
RF: Random Forest
DT: Decision Tree
KNN: K Nearest Neighbor
ML: Machine Learning
CRISP-DM: Cross Industry Standard Process for Data Mining
ROC: Receiver Operating Characteristic
MCC: Matthews Correlation Coefficient
PR: Precision-Recall
CSV: Comma Separated Values
ETL: Extract Transform Load
SMOTE: Synthetic Minority Oversampling Technique

1 Introduction

1.1 Background and motivation

Patients who do not present in the scheduled appointment which means no-shows are considered as a critical issue in health care settings. This will consequently lead to a waste of resources, money, and time(Healthwatch Lincolnshire, 2014; Hasvold and Wootton, 2011, Dantas et al., 2018). Missed appointments are an issue that has health and economic consequences and concerns children, and juveniles(Triemstra & Lowery, 2018). Nang et al. (2014) stated that they have an impact on longer appointment times leading to dissatisfied patients due to poor management and eventually loss of revenue for the hospital. In this report, they mentioned that 36% of patients forget the appointments and 27% feel good and just stayed at home and did not present in the appointment. They also used SMS reminders before the appointment, but one out of four patients were absentees. This has done tremendous waste of resources and time. There are many ways to see the side effects of missed appointments in this sector and described below.

First, health expenditures are skyrocketing and facing health-related spending (Bhattacharya et al., 2014). In another research conducted by Bech(2005), there are two types of costs involved when patients missed appointments, namely, social costs and financial costs. Social costs refer to unused facilities like ward capacity, and time whereas Financial cost means less amount reimbursed due to lack of patients because patient missed the appointment.

Secondly, Stubbs et al.(2012) present the research on the reduced productivity of health care providers because of workflow, and efficiency in the clinics. This has resulted in dissatisfaction with higher waiting times and perception of an overall decrease in the quality of the healthcare system(Dantas et al., 2018). Moreover, no-show delays clinical care and the resources could have used in improving services and quality(NHS Digital, 2018)

Thirdly, due to increasing no-shows, waiting times get increased for other patients as well and they do not get proper treatment on time(Bhattacharya et al., 2014).

The pivotal part is that patients who miss appointments have a negative impact on their health because they are the ones who need treatment. DiMatteo (2000) in his research states that if any patients avoid the appointment they will suffer at last anyway.

My motivation for this thesis is to check the data quality, improve the data quality, prepare the data for analysis, modeling and evaluating the selected models and finally deliver the best model.

1.2 Definitions

There are several definitions used in the literature to describe the phenomenon in which patients do not show appointments at scheduled dates and times. Attending an appointment means “attending an appointment that had been prearranged” (Guy et al.,2012). Dantas et al.(2018) defined missed appointments as broken appointments or no-show appointments. According to Pesata et al. (1999) missed appointments referred to patients who “do not attend their scheduled visits”, or “fail to appear for their visit”. Turkcan et al. (2013) refer to missed appointments and cancelled appointments are different.

No show is the event when a patient does not come for a previously scheduled appointment or cancels with minimum time where the appointment slot will be empty, and none get the appointment (Hanauer & Huang, 2014).

As discussed, various terms are equivalent to the no-show definition and have been used in many conditions and domains, but the meaning remains the same.

1.3 Related Work

1.3.1 Show and No-show rates

To delineate the issue of no-show, there will be mentioned some research conducted for no-show in outpatient care.

Dantas et al. (2018) concluded from 105 papers that the average no-show rate across all studies was found to be 23.0%, and further analysis revealed that this rate was highest in the African continent (43.0%) and lowest in Oceania (13.2%). In Asia, there is a contribution of 25.1% of no show and 19.3% in Europe. Its reach is global. They also identified patient characteristics that were more frequently associated with no-show behaviour: Young adults, poor patients, place of the house is far from the clinic; no private insurance. Furthermore, patients with mental health problems, those taking psychiatric medication and/or making use of tobacco, drugs, and/or alcohol were also frequently found to be more likely to miss their appointments. They also find out that indicate that primary care and psychiatric care are the core research area that is most explored regarding no-show in appointment scheduling. Dantas et al. (2018) depict the median no-show rate over ten years period has decreased over time, while keeping continents in analysis. The box plot is portrayed in Figure 1.

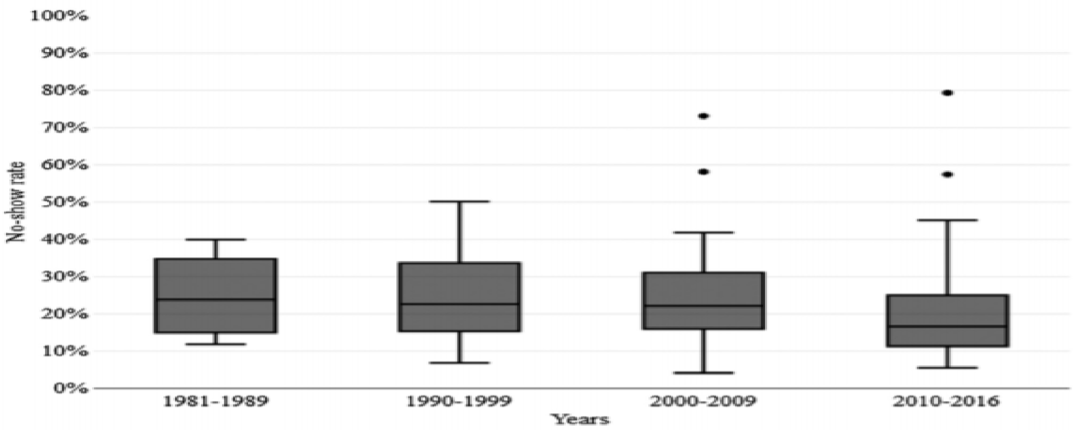


Figure 1: Box plots of No-show rates over 10 years period
Source: Adapted from Dantas et al. (2018:415)

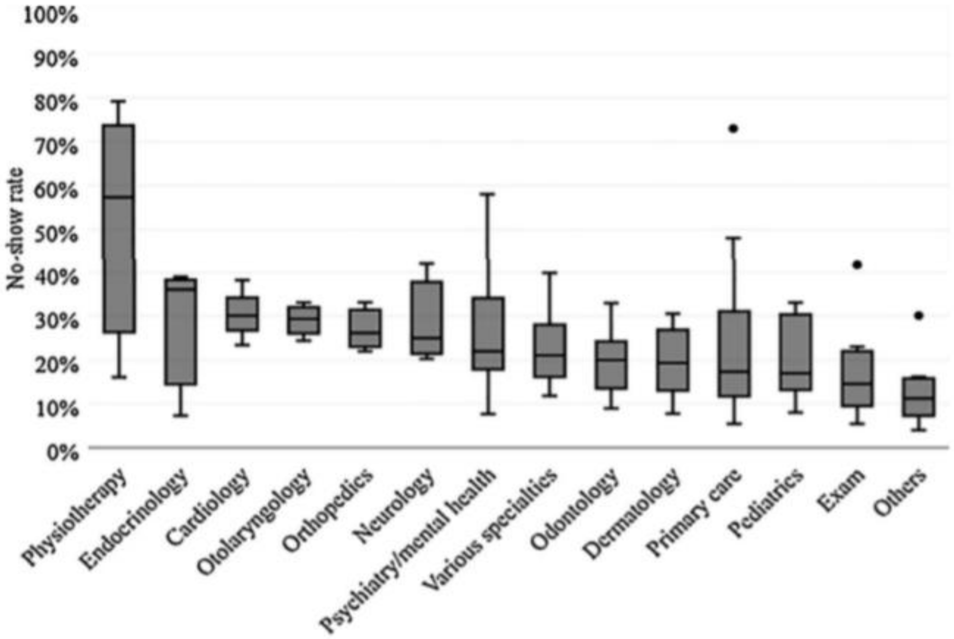


Figure 2: Box Plots of medical departments concerning no-show
Source: Adapted from Dantas et al. (2018:415)

In figure 2, Physiotherapy holds the highest median no-show rates (57.3%), followed by endocrinology and cardiology. In contrast, exams and others had the lowest median no-show rates. The interesting part here is Psychiatry/mental health and primary care was most studied dealing with no-show rates.

NHS Digital(2018) investigated attendance and show rates in NHS hospitals, England in 2017-2018. The study gave some interesting results such as The number of outpatient appointments has doubled since 2007-08, surging from 66.6 million to 119.4 million in 2017-18. Furthermore, the number of patient attendance has skyrocketed significantly from 54.4 million in 2007-2008 to93.5 million in 2017-18. They also provided a note that patients aged 60-79 accounted for 30% of all attendance.

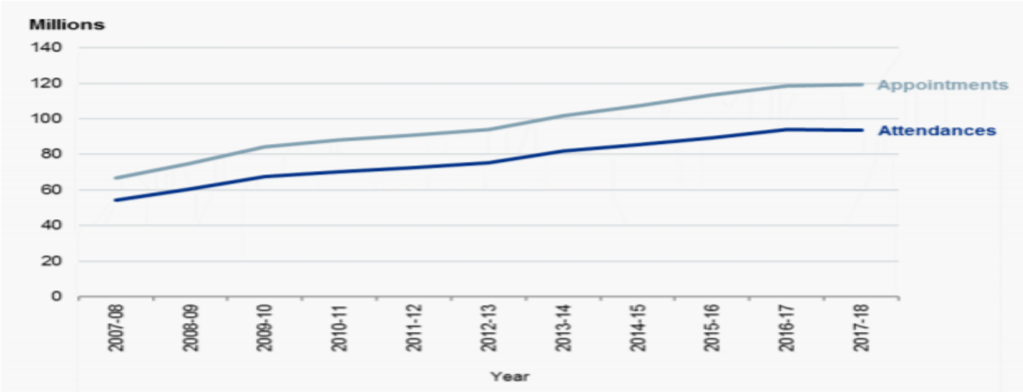


Figure 3: Comparison of attendance and appointment over 10 years period
Source: Adapted from NHS Digital(2018)

1.3.2 Exploratory Data Analysis and Machine learning models

Numerous journal papers and articles from other studies have been studied and investigated about no-shows in hospitals to reduce the no-show rate. Analysts have used exploratory data analysis and model the data to evaluate the models.

Most of the studies have found that age is inversely proportional to the probability of no-show(Menendez et al.,2015; Peng et al.,2016). Young adults were likely to miss most appointments. Insured patients were highly likely to show at the appointments than those who pay their medical fees(Menendez et al.,2015; Karter et al.,2004). Another research conducted showed that gender is not a statistical predictor of missed appointments, however, some studies reported that men missed more appointments than women(Peng et al.,2016; Torres et al.,2015). Day of the week and month of the appointment and appointment time were not a predictor of missed appointments according to many research papers(Daggy et al; Torres et al.,2015). But some research studies found that most non-attendance occurs on Mondays(Kheirkhah,2016; Torres et al.,2015).

Kurasawa et al.(2016) used a logistic regression model to predict no-shows for diabetes patients. The value of AUC for the best predictor was 0.958. Precision was 0.757, the recall was 0.659 and F-measure was 0.704. Similar research was conducted by Elvira et al.(2018) and they used the Gradient Boosting algorithm for no-show prediction. The model evaluated AUC as 0.74. Mohammadi et al.(2018) studied three ML models to predict the no-show of the next scheduled medical appointment. Naïve Bayes held the highest accuracy of 82%. The AUC for logistic regression, naïve Bayes, and Multilayer perceptron are 0.81, 0.86, and 0.66, respectively.

On the other hand, three ML algorithms were identified, Random Forest, Decision Tree, and Naïve Bayes, and important features were selected for training the models. Random forest outperformed the other two models achieving the AUC of 0.697.

1.4 Objectives

The first objective in this thesis will be to check and evaluate the quality of the medical data used for the case study. The quality will be evaluated by looking at data types, check for missing values in the rows, check for null values, removing unnecessary symbols and words from the dataset. Good data quality plays a crucial role in data analysis.

The second objective is to explore and analyze the dataset. This is achieved by using necessary columns and rows. This part is crucial to understand the data. The number of columns, rows, the shape of data, properties of data, etc. can be obtained during this phase. By exploring the dataset, I found out young adults missed most appointments, and the most missed days were Monday.

The third objective is to use the models available. I have used 6 binary classifier models. Although the target variables have 3 classes, I have reduced them to 2 classes to make it binary.

The fourth objective is to evaluate the used models. Confusion matrix, Precision, Recall, AUC ROC Curve, Precision-Recall curve, error rate, MCC are the metrics that will be implemented in the thesis**.**

The fifth objective is to write conclusions, Future work, and limitations of the thesis report.

1.5 Dissertation organization

The rest of the thesis incorporates four chapters and structured as follows:

In chapter 2, I focus on background theories of several machine learning models such as Decision Trees, Random Forest, Logistic Regression, Gradient Boosting, GaussianNB, and K nearest Neighbours. Briefly, I emphasized evaluation metrics such as Confusion Matrix, Area under ROC curve, Precision-Recall curve.

Chapter 3 focuses on the experimental setup and methodology followed throughout the thesis. I have implemented and used the CRISP-DM methodology. Concepts of this data mining process are explained. Data cleaning, data preparation is briefly explained in this chapter.

Chapter 4 consists of results and discussion. Accuracy score, analysis, and discussion of ROC Curve, PR curve, Matthews correlation curve, etc are discussed in detail.

Finally, chapter 5 comprises Conclusions. Summarizing achievements, limitations and future work are illuminated in this chapter.

2 Background Theory

This thesis uses machine learning models for model building and evaluating the models in the evaluation phase. The task in the thesis is to classify the target variables by using multiple classification models by predicting the class labels. A classification issue will be encountered when an instance of a class needs to be assigned into a certain fixed class based on several observed attributes related to that object. Classification problems are huge in industries and markets. For instance, Stock market prediction, Weather forecasting, Medical diagnosis, Speech recognition (Moghadassi et al.,2016; khan et al,2009). This thesis will focus on supervised learning classification models meaning

2.1 Machine learning models

One of the most used machine learning works, which involves predicting a target variable in the previously unseen data, is a classification that falls under supervised learning(Mohamed et al.,2015). I have implemented different classification algorithms in this thesis. Since the problem is to classify the target variable, unsupervised learning is ignored. The classification model aims to predict a target class in our case it is ” c_kontaktAvsluttkodeNavn” by building and comparing several classification models based on a training dataset, and then testing the model in test data(Witten, et al., 2011). I have used several machine learning models(Classification Models) such as Decision Trees, Random Forest, Logistic Regression, Gradient Boosting, K Nearest Neighbours, Naïve Bayes.

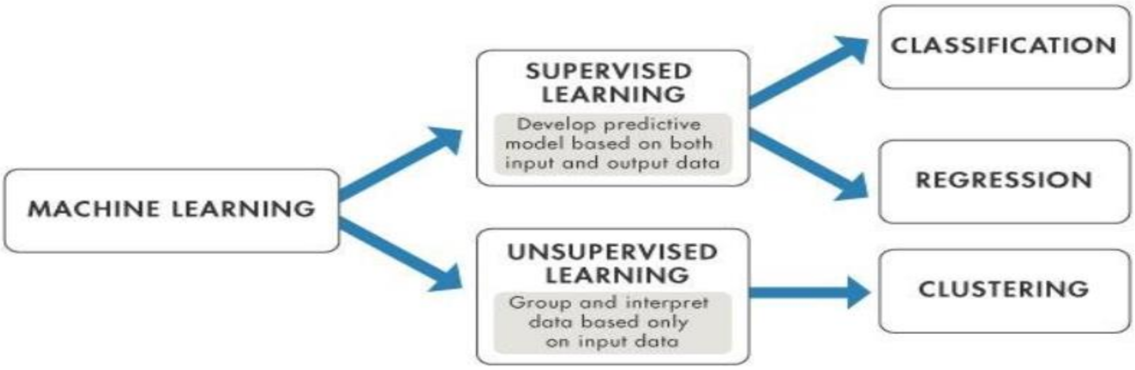

MACHINE LEARNING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
UNSUPERVISED LEARNING
Group and interpret data based only on input data
CLASSIFICATION
REGRESSION
CLUSTERING

Figure 4. Supervised Learning versus Unsupervised Learning (Bunker & Thabtah 2019).

2.1.1 Decision Tree

Decision trees are the most popular machine learning algorithm for classification problems. It is easy to use and interpret, handles categorical features of data, works on binary and multiclass classification problems. This algorithm does not have to scale and normalize the data. Therefore, a decision tree requires less effort for data pre-processing during data preparation. However, I have scaled the data throughout the data modeling process. The drawback of this model is that they are unstable, a small change in data leads to a change in the structure of the optimal decision tree. This can be remedied by replacing single trees with multiple trees. The code below is the instantiate of the decision tree.

```
model3 = tree.DecisionTreeClassifier(max_depth = 13)
```

("Classification and regression - Spark 3.1.2 Documentation", 2021).

2.1.2 Random Forest

Random forest is the most popular family of a classification systems. It is the ensemble of decision trees. It is better than a decision tree because it combines many decision trees resulting in the risk of overfitting. Random Forest supports and handles categorical and numerical variables well. The algorithm works randomly in the training dataset, in such a way each decision tree is different. Due to this randomness, there are chances of model bias slightly. From each decision tree, the random forest gets the class vote, and then it is converted into a majority vote by taking an average of all class votes obtained from each decision tree. The most crucial parameter for improving the performance of this model is and max_depth.

The code below is the instantiate of Random Forest.

```
model4 = ensemble.RandomForestClassifier(n_estimators=1000, max_depth=5)
```

Max_Depth: This parameter is so powerful and expressive. On the other hand, increasing max_depth takes a longer time to train the data and prone to overfitting("An Implementation and Explanation of the Random Forest in Python", 2021).

2.1.3 Logistic Regression

Logistic regression is a classification model suitable to predict categorical responses. It predicts the probability of outcomes. Using family parameters while selecting binomial or multinomial logistic regression otherwise, the spark will automatically find the correct variant and classify the parameter.

For binary classification problems, the algorithm outputs a binary logistic regression model. Given a new data point, denoted by x, the model makes predictions by applying the logistic function:
$f(z)= 1\div(1+e-z)$, this function is called as Sigmoid function[3].
where $f(z)$: output between 0 to 1.
    z: Input to the function.
    e: base of natural log
The sigmoid function is used to map predictions to probabilities.

The code below is the instantiate of Logistic regression.

```
model1 = LogisticRegression(solver='saga', max_iter=100)
```

max_iter: Number of iterations for solvers to converge.
Solver: saga is used for the larger dataset.
("Logistic Regression — ML Glossary documentation", 2021)

2.1.4 Gradient Boosting

Gradient boosting is a general-purpose model and the most efficient machine learning algorithm used for classification.

The key idea of the algorithm is iterative minimization of target loss function by training each time one more estimator to the sequence. In this implementation, decision trees are taken as estimators. Boosting means being strong. By using this model, weak learners become strong learners. Predictions of the final ensemble model are the weighted sum of all predictions from initial trees. All the weighted sum of predictions of trees are calculated making this model better than decision trees.

Models are fit using loss function and gradient descent optimization algorithm. This is how this name of the model comes to exist ("Understanding Gradient Boosting Machines", 2021).

The code below is the instantiate of Gradient Boosting.

```
model5 = ensemble.GradientBoostingClassifier(n_estimators=100, learning_rate=1.0, max_depth=1, random_state=0)
```

Parameters included:
n_estimators - The number of trees or estimators in the model.
learning_rate – learning rate of model
max_depth - This parameter is so powerful and expressive. On the other hand, increasing max_depth takes a longer time to train the data and prone to overfitting

2.1.5 K Nearest Neighbour

K Nearest Neighbour is a simple and most preferred algorithm that classifies data points based on the points that have more similar characteristics with other data points. It uses test data to guess which unclassified will be classified to some data points. The benefit of using this classifier is: it is easy to use and interpret, quick processing time, and straightforward. However, it has some drawbacks such as finding optimal value of K, fewer quality data gives less accuracy to the model. This model is suitable for recognition systems, classification systems, and so on.

The idea behind KNN is that firstly, data pints need to be in feature vectors. This model then finds the distance between the values of these points and this is possible using Euclidean distance ("K-Nearest Neighbors (KNN) Algorithm for Machine Learning", 2021).

This is calculated as:

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

Figure 5: Euclidean Distance formula

Adapted from: "K-Nearest Neighbors (KNN) Algorithm for Machine Learning", 2021

KNN finds the Euclidean distance between each data point and the test data. After that, it finds the probability of calculated points that is being like the test data and classifies it based on which points share the highest probabilities ("K-Nearest Neighbors (KNN) Algorithm for Machine Learning", 2021).

The code below is the instantiate of K Nearest Neighbor.

```
model2 = KNeighborsClassifier(n_neighbors=4)
```

2.1.6 Gaussian Naïve Bayes

Gaussian Naïve Bayes is a classification algorithm that uses Bayes' theorem to classify data points. The Bayesian theorem describes the probability of an event will occur if you have prior knowledge of a condition related to the specific event.

Naïve refers to data points independent of one another. Naive Bayes classifiers use the probabilities of certain events being true and other events being true to make predictions about new data points. Therefore, this is a unique algorithm in the classification domain. Certain advantages include ease to build, use, train, and ignore useless variables. However, disadvantages include that it assumes data points as independent and does not work well with smaller data sets.

This algorithm is popular in spam detection, missing appointments, and facial recognition ("Naive Bayes Classifiers for Machine Learning", 2021).

Let us consider two events A and B. The formula to calculate the probability of different events occurring is:

$$P(B|A) = \frac{p(A|B)p(B)}{p(A)}$$

Figure 6: Formula of Bayes Theorem

Source: Adapted from "Naive Bayes Classifiers for Machine Learning", 2021

Where,
P(B|A) = is the probability that event B will occur if event A is true.
p(A|B) = is the probability that event A will occur if event B is true.
P(A), p(B) = is the probability of events A and B that occur independently of each other.
The code below is the instantiate of Gaussian Naïve Bayes Classifier

```
model6 = GaussianNB(var_smoothing=1e-9)
```

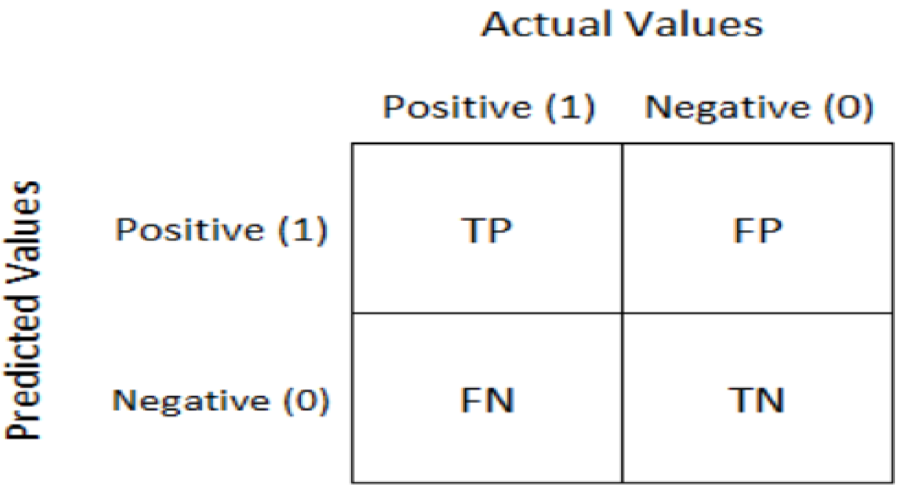
2.2 Model Evaluation

Models are typically evaluated by Confusion matrix, AUC, Precision-Recall curve.

The data is split into training and testing data before evaluation. From this thesis, training data is 70%, and testing data is 30%. While evaluating the models, data must be balanced and scaled properly. An imbalanced dataset makes the models appear impressive while they are not real. Some of the models such as Logistic Regression must be scaled otherwise the result will be worthless.

2.2.1 Confusion Matrix

After building the model, it is essential to evaluate the model. This is when the confusion matrix comes into play. It measures the performance of models for classification problems. The target variable I have used in the thesis has a binary class. Therefore, I will have a 2*2 Confusion Matrix. It is a table with four combinations of Actual and Predicted values.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 7: 2*2 Confusion Matrix for Model Evaluation.
Source: Adapted from Understanding Confusion Matrix, 2021.
TP stands for True Positive.
FP stands for False Positive.
FN stands for False Negative.
TN stands for True Negative.
Having the values of all these values will help us to calculate other important elements of evaluation such as precision, recall, F1-score, and AUC-ROC Curve("Understanding Confusion Matrix", 2021).

Understanding all these evaluation metrics is important and should be inclined to business problems and objectives.
Let me provide examples of the confusion matrix metrics.
For instance, in Hospital missed appointments:

True Positive:
Elucidation: Actual is positive and predicted is positive.
It is a predicted show and it is a show("Understanding Confusion Matrix", 2021).

True Negative:
Elucidation: Actual is negative and predicted is negative.
It is a predicted no-show and it is a no-show.

False Positive:

Elucidation: Actual is Negative and predicted is Positive.
It is a predicted show, but it is a no-show.

False Negative:
Elucidation: Actual is Positive and predicted is Negative.
It is a predicted no-show and it shows.

Recall:
Out of all positive classes, how much we predicted correctly is called Recall. The recall comes to play when False Negative trumps False Positive. It is also called Sensitivity or True Positive Rate.
It is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Precision: Out of all classes predicted positive, how many are positive is called Precision. Precision is a useful metric when False Positive is a higher concern than False Negative. It is also called a True Negative rate or Specificity.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Accuracy:
Out of all classes, how many predicted correctly is called Accuracy.

$$\text{Accuracy} = \frac{TP+TN}{TP + FP+TN+FN}$$

F1-score: It represents the balance between precision and recall. It is also an important metric in the Machine learning algorithm. Usually, to evaluate the performance of algorithms, F1-score need to be checked. Precision and recall both need to be combined and checked and F1-score does that. It is the harmonic mean of precision and recall.

$$\text{F1-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

("Understanding Confusion Matrix", 2021).

Matthews correlation coefficient:
It is also an evaluation method for binary class classification problems. It is a powerful evaluation metric that provides a single value as a result and when both positive and negative classes are equally important for the study of models. MCC provides more information about the models rather than that of accuracy score and f1-score. This metric is mostly applicable in bioinformatics and medical fields. The coefficient score lies between -1 to +1. The number that lies close to 1 is better and the number that lies around -1 is considered not a good score. It is the most informative among all the single evaluation metrics discussed so far. This coefficient is calculated from the confusion matrix as well. The formula for the calculation of Matthews correlation coefficient is as follows:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

It is the correlation coefficient between the observed and predicted binary classifications. MCC displays a high score if the binary classifier correctly predicted most of the positive samples and most of the negative samples and if most of its positive predictions and most of its negative predictions are correct. If the model cannot correctly predict most of the positive and negative samples, then the score will be low (Chicco, Tötsch & Jurman, 2021).

Error Rate:

When it comes to incorrect classification error rate comes into play. The error rate has a score from 0 to 1. Rate near to 0 is best and rate near 1 is worst. It is calculated as the total number of two incorrect predictions divided by total samples. The two incorrect predictions are FP and FN. It can also be obtained by subtracting 1 with the accuracy score of the model.

2.2.2 Area Under Curve

An excellent model has an AUC near 1 which implies it has a good measure of separability and a poor model has an AUC near 0 which implies it has the worst measure of separability. This metric demonstrates how much the model can distinguish between classes. ROC is plotted with TPR against FPR. This metric is desirable because of the scale-invariant and classification threshold invariant. The threshold value for AUC is 0.5, which means it has no discrimination power to distinguish between positive class and negative class.

True Positive Rate(Recall) and True Negative Rate are inversely proportional to each other. Similarly, when the threshold value gets decreased, positive values increased which will eventually increase the TPR and decrease TNR.
Also, when the threshold value gets decreased, negative values increased which will eventually increase TNR and decrease TPR("Classification: ROC Curve and AUC | Machine Learning Crash Course", 2021).

2.2.3 Precision-Recall Curve

This is a special type of curve that considers Precision and Recall. This is very different from than AUC curve and uses a limited Data science community. When there is a higher case of data imbalance, this type of curve is very useful to look at it. Since our data is an imbalanced dataset, I must calculate and plot this curve as well. As stated before, true negatives samples are not of big concern as true positives samples. Therefore, a suitable metric that is not as determined by the number of true negatives should be used. A suitable metric for this type of situation is the Area under the Precision-Recall(PR) curve. As the name suggests, it is a curve plotted between precision and recall. The recall is positioned on the horizontal axis and precision is placed on the vertical axis. The perfect PR score is 1 and the minimum score is 0. The perfect score of 1 implies that all positive predicted values are correct, and all positive values are detected.

For different thresholds, the plot shows the trade-off between precision and recall. A high area under the PR curve represents both high recall and high precision, where high precision means low false-positive rate, and high recall means low false-negative rate. High scores for both show that the classifier is returning high precision, and high recall. This curve is more useful in binary classification problems. The curve will be different than the AUC curve because this curve used precision and recall("Precision-Recall — sci-kit-learn 0.24.2 documentation", 2021).

3 Experimental setup

3.1 CRISP-DM Methodology

The experiment setup I have followed is a standard framework for the data science project, CRISP-DM(Cross Industry Standard Process for Data Mining). This framework addresses business and data understanding, data preparation, model building, evaluation, and deployment. (Wirth & Hipp, n.d.)

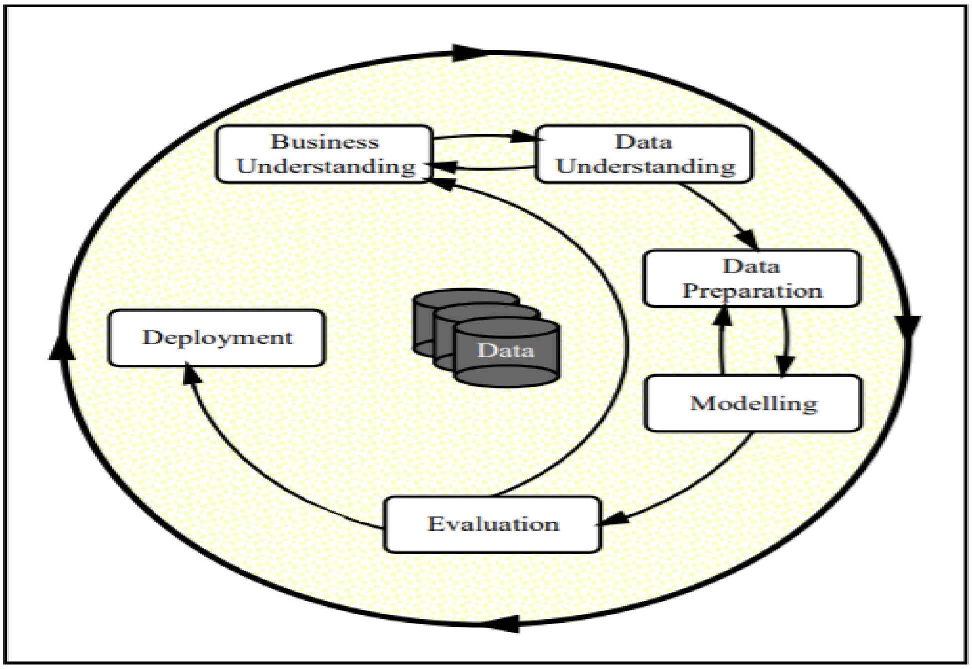


Figure 8: Phases of the CRISP-DM methodology
Source: Adapted from (Wirth & Hipp, n.d.)

From figure 8, the different steps are iterative, meaning this process can be improved over time. This thesis has followed this standard framework. Most of the thesis is about data preparation and modeling.

The phases of CRISP-DM methodology are described in brief below:

3.1.1 Business Understanding

This is the initial phase of this methodology. This phase focus on project aims and objectives from a business perspective. After the objectives, it will be implemented to create this knowledge for data mining problems. For my master thesis, the project aims to find the characteristics of people who miss appointments, explore the data, find a suitable classifier. This level of understanding is necessary to understand the problem.

3.1.2 Data understanding

This phase starts with data collection methods. This is where data quality issues can be recognized. More information about data can be studied during this phase of the data mining process. I have been provided data from the Helse vest. Data collection was done from the analyst of the hospital. I was advised about data quality issues such as missing values, empty rows, extra strings, etc.

3.1.3 Data Preparation

This is the most crucial phase of the data mining process. Complete data preparation has an overall impact while modeling the dataset. Data cleaning, reformatting, changing data types, combining data for analysis, data transformation such as using pipeline, OneHotcodEncoder. This phase takes a long time while analyzing data science projects.

3.1.4 Modeling

Different modeling techniques are used in this phase. Appropriate parameters for specific models can be used to get optimal results from the model. There is a close relationship between data preparation and modeling. Better the data preparation more accurate results can be obtained during the modeling and evaluation phase.

3.1.5 Evaluation

Before deploying the model, the evaluation phase needs to be thoroughly studied and analyzed. Model evaluation should match the business aim and objectives. It also checks if any objectives are still not satisfied.

3.1.6 Deployment

After the model is created and evaluated deployment comes to take place. Users have an immense role in this phase as after deployment users give feedback and it can be improved over time. The models deployed need to be user-friendly and the company that deploys need to improve the model based on feedback as well. (Wirth & Hipp, n.d.)

3.2 Data Quality

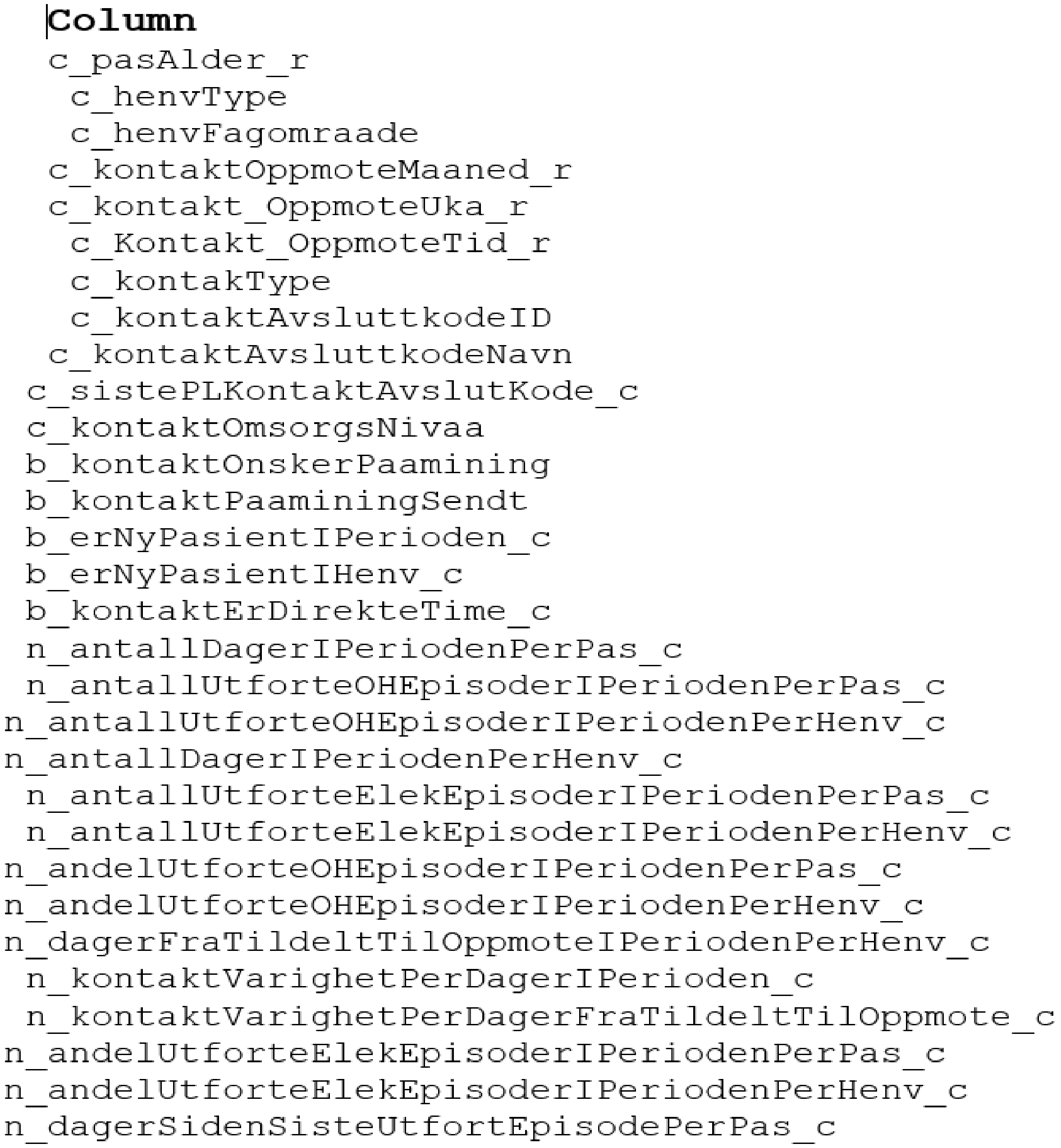
In this sub-chapter quality of the dataset will be studied and visualized. The column names are in the Norwegian language. Some of the column names are difficult for native Norwegians to understand as well. Despite that, it has been converted to English. The data is lacking dates, SMS reminders due to GDPR issues which is critical for data confidentiality. In this topic, an overview of the dataset, its description will be elaborated.

3.2.1 Summary of Dataset

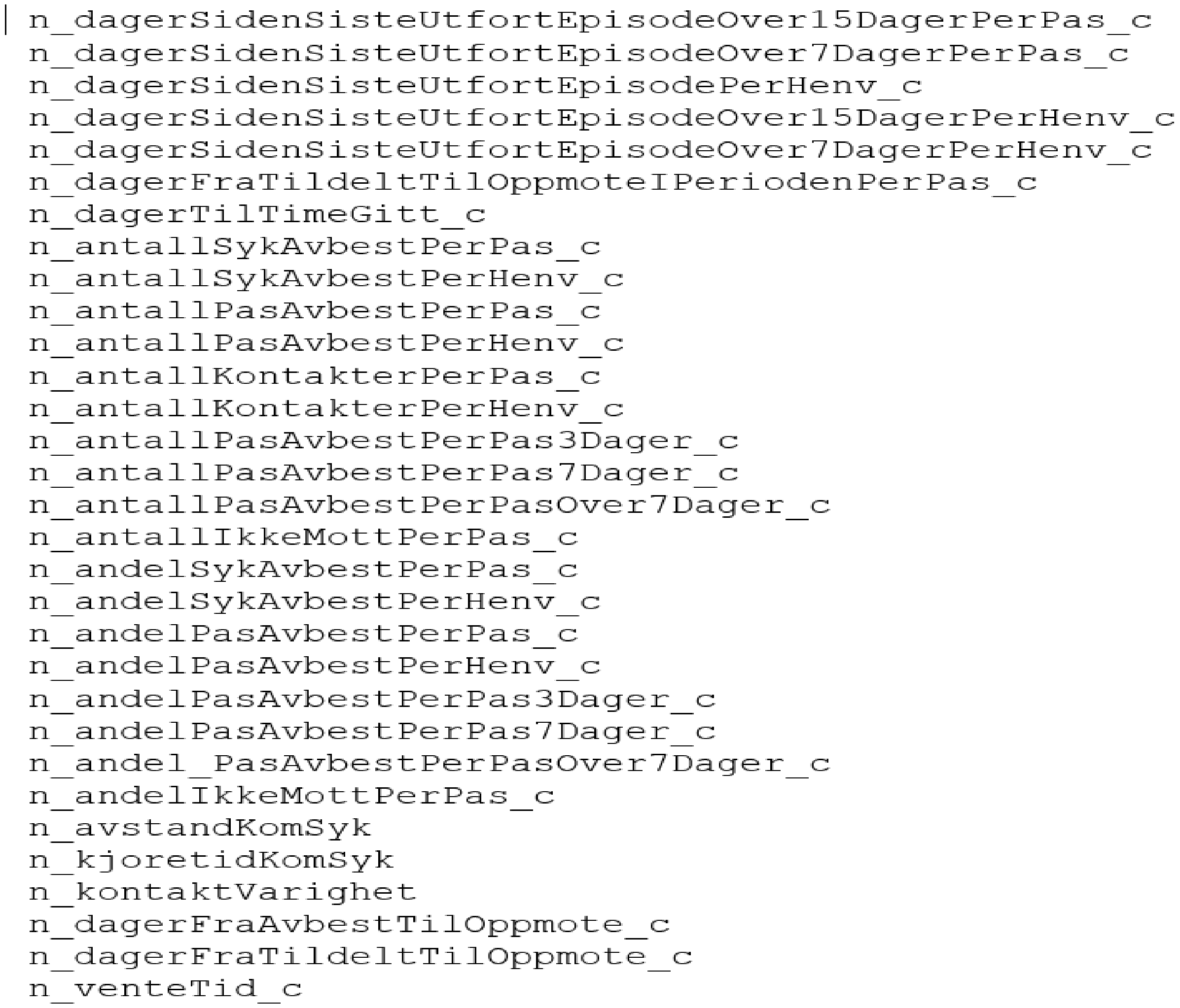
The dataset used in this thesis is provided by Helse Vest. It contains medical data about appointments. The dataset is about 6.59 MB of data. It is a CSV file. It has 61 columns and 31858 rows. Due to privacy reasons: gender, SMS messages, the date is not included in the dataset. This also gives less opportunity to compare data on different dates, whether SMS encourages the patient to be present on an appointment and gender to know which type of age group and which gender miss appointments. Overall, finding characteristics of a patient who misses appointments would be difficult in this case.

3.2.1.1 Attributes

The dataset consists of 61 columns.



```
  Column
  c_pasAlder_r
   c_henvType
   c_henvFagomraade
  c_kontaktOppmoteMaaned_r
  c_kontakt_OppmoteUka_r
   c_Kontakt_OppmoteTid_r
   c_kontakType
   c_kontaktAvsluttkodeID
  c_kontaktAvsluttkodeNavn
 c_sistePLKontaktAvslutKode_c
 c_kontaktOmsorgsNivaa
 b_kontaktOnskerPaamining
 b_kontaktPaaminingSendt
 b_erNyPasientIPerioden_c
 b_erNyPasientIHenv_c
 b_kontaktErDirekteTime_c
 n_antallDagerIPeriodenPerPas_c
 n_antallUtforteOHEpisoderIPeriodenPerPas_c
n_antallUtforteOHEpisoderIPeriodenPerHenv_c
n_antallDagerIPeriodenPerHenv_c
 n_antallUtforteElekEpisoderIPeriodenPerPas_c
 n_antallUtforteElekEpisoderIPeriodenPerHenv_c
n_andelUtforteOHEpisoderIPeriodenPerPas_c
n_andelUtforteOHEpisoderIPeriodenPerHenv_c
n_dagerFraTildeltTilOppmoteIPeriodenPerHenv_c
 n_kontaktVarighetPerDagerIPerioden_c
 n_kontaktVarighetPerDagerFraTildeltTilOppmote_c
n_andelUtforteElekEpisoderIPeriodenPerPas_c
n_andelUtforteElekEpisoderIPeriodenPerHenv_c
n_dagerSidenSisteUtfortEpisodePerPas_c
```



```
n_dagerSidenSisteUtfortEpisodeOver15DagerPerPas_c
n_dagerSidenSisteUtfortEpisodeOver7DagerPerPas_c
n_dagerSidenSisteUtfortEpisodePerHenv_c
n_dagerSidenSisteUtfortEpisodeOver15DagerPerHenv_c
n_dagerSidenSisteUtfortEpisodeOver7DagerPerHenv_c
n_dagerFraTildeltTilOppmoteIPeriodenPerPas_c
n_dagerTilTimeGitt_c
n_antallSykAvbestPerPas_c
n_antallSykAvbestPerHenv_c
n_antallPasAvbestPerPas_c
n_antallPasAvbestPerHenv_c
n_antallKontakterPerPas_c
n_antallKontakterPerHenv_c
n_antallPasAvbestPerPas3Dager_c
n_antallPasAvbestPerPas7Dager_c
n_antallPasAvbestPerPasOver7Dager_c
n_antallIkkeMottPerPas_c
n_andelSykAvbestPerPas_c
n_andelSykAvbestPerHenv_c
n_andelPasAvbestPerPas_c
n_andelPasAvbestPerHenv_c
n_andelPasAvbestPerPas3Dager_c
n_andelPasAvbestPerPas7Dager_c
n_andel_PasAvbestPerPasOver7Dager_c
n_andelIkkeMottPerPas_c
n_avstandKomSyk
n_kjoretidKomSyk
n_kontaktVarighet
n_dagerFraAvbestTilOppmote_c
n_dagerFraTildeltTilOppmote_c
n_venteTid_c
```

Table 1: Overview of Attributes name

The description is written from Helse vest in an excel file. Although, only relevant and useful columns have been translated into English and some of the columns are not even described.

I will describe some columns that are useful for the data analysis part. Also, some of them are converted into English.

1. c_henvType – Referral type

2. c_kontakt_OppmoteTid_r - Scheduled appointment time, hour

3. c_kontakt_OppmoteUka_r - Scheduled appointment time, weekday

4. c_kontaktAvsluttkodeNavn – Appointment exit code in text (Ikke møtt/ingen beskjed = No-show, Pasientønsket avbestilling = Cancelled by patient,   Ordinært avsluttet = Appointment conducted). This is a target variable.

5. c_kontaktOmsorgsNivaa – care level of Scheduled appointment

6. c_kontaktOppmoteMaaned_r – Appointment time, month

7. c_kontaktType – Appointment type

8. c_pasAlder_r – Age group 10 years of difference.

c_kontaktAvsluttkodeNavn is target Variable. This variable has three instances namely: Ikke møtt/ingen beskjed = No-show, Pasientønsket avbestilling = Cancelled by patient, Ordinært avsluttet = Appointment conducted.

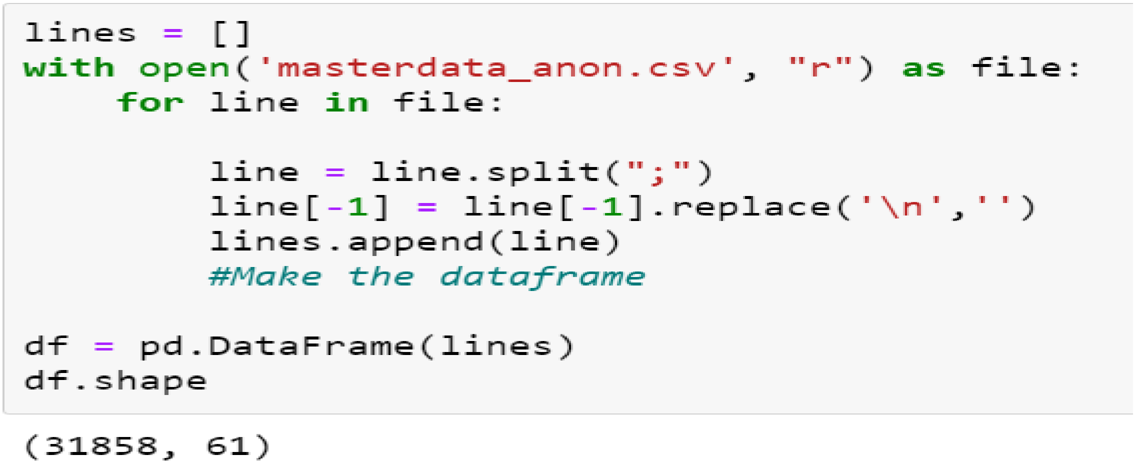
c_pasAlder_r refers to age group. 20-30, 30-40, 40-50, 50-60, 60-70, 70-80 and above 80 are age group divided equally. For example, in the age group [20-30[, 20 is included and 30 is excluded, it means till 29 years.

3.2.2 Data preparation

Data preparation also refers to data pre-processing, which deals with what the data is, check data quality and make it better for modeling, check for the data types, combining and consolidating data, and transform data to use for analysis purposes. In conclusion, it gives better data analysis experience. Data accessing, sorting, exploratory data analysis all come under data preparation. Without data preparation or little use of this phase have misleading results because it provides misleading results. Dirty dataset always needs to be formatted and clean and by following a standard set of rules only it should be ready for the analysis(Modelling). Analyst spends too much time planning and doing data preparation because of dirty data, errors, and imbalanced data. This is expensive and time-consuming. Missing values, changing data types, checking relationships using libraries are tested and checked in this phase.

3.2.2.1 Data Loading

This is the most important part to start with data preparation. Without loading the data, it is impossible to prepare. Loaded data is prepared for analysis. Data loading is a primary step in data analysis. The read_csv() method converts a CSV file to a Pandas object which is understandable by python. The dataset I have received is a CSV file and this needs to be converted to Pandas objects.



```
lines = []
with open('masterdata_anon.csv', "r") as file:
    for line in file:

        line = line.split(";")
        line[-1] = line[-1].replace('\n','')
        lines.append(line)
        #Make the dataframe

df = pd.DataFrame(lines)
df.shape
```

```
(31858, 61)
```

Listing1: Loading CSV file masterdata_anon into pandas DataFrame

In this listing, a CSV file is loaded into DataFrame. It becomes Pandas' objects after loading it. This process is a part of the ETL(Extract Transform Load) Process. This dataset has 31858 rows and 61 columns.

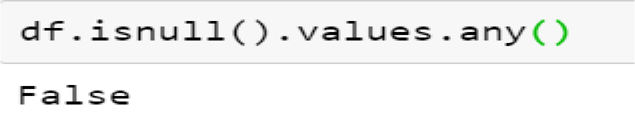
3.2.2.2 Data Cleaning

This is the process of removing or fixing data, corrupted, inaccurate, incomplete data within the dataset. This type of imbalanced and dirty data gives undesired results. It is also called Data scrubbing. Good data cleaning assists to make better decisions. There are several steps for data cleaning.

First of all, removing duplicate data or irrelevant data, Duplication is considered a serious issue and often makes the analysis hard and makes the wrong output. De-duplication is the area of interest not to be overlooked during data science projects.

Handling missing values plays a vital role and with missing values classifier gives an error message. Dropping the missing values is the easiest way to work faster and it is the most common method to tackle in this type of situation.

Checking data types is also a good practice to explore and analyze data. The data variables that need to be analyzed and use in a model need to be numeric. The data provided to me was of Object data type. I had to change the data types to numeric which made the work for analysis and evaluation of models quick and hassle-free("Data cleaning: The benefits and steps to creating and using clean data", 2021).



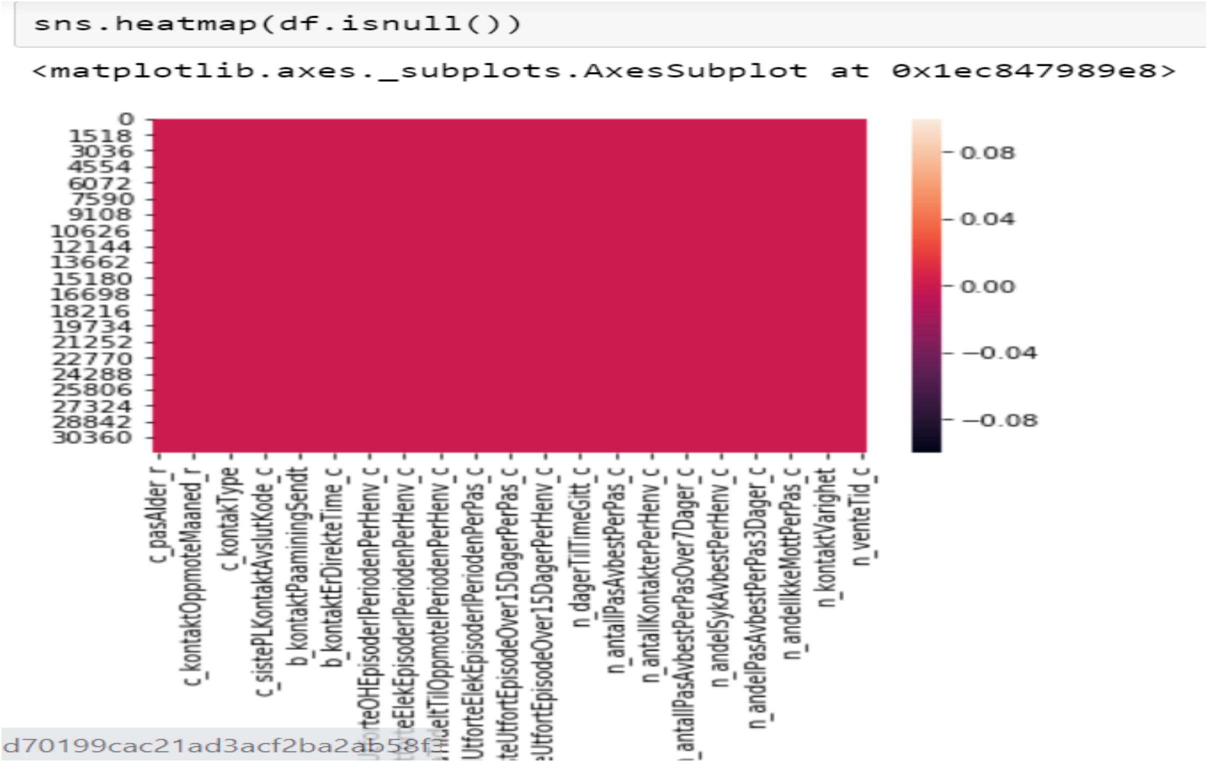
```
df.isnull().values.any()
```

```
False
```

Listing2: Checking for null values

This command returns False, which means there are no null values.

By pictorial representation as well, I can show whether the dataset has null values or not.

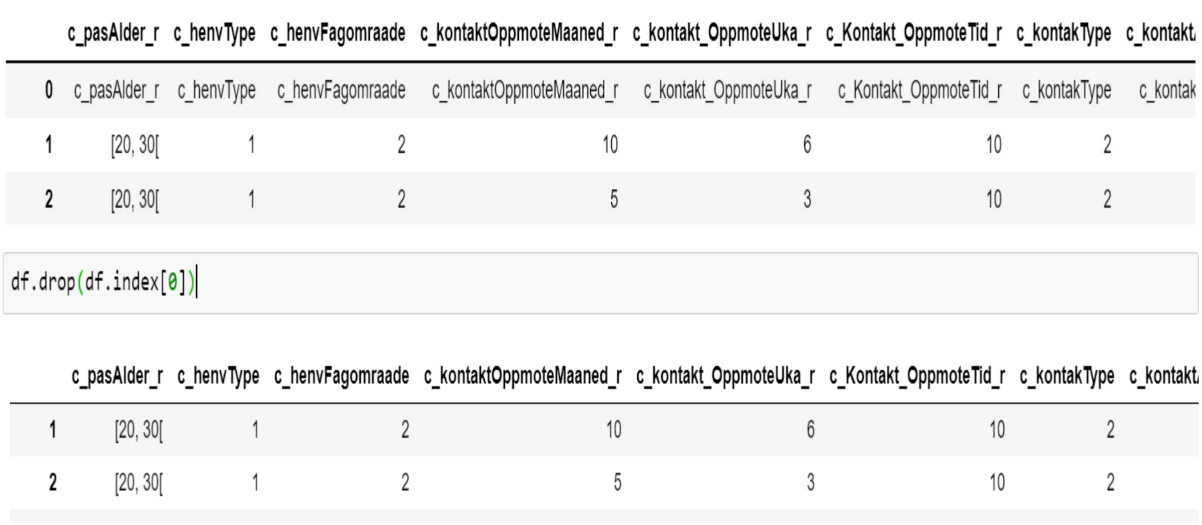


```
sns.heatmap(df.isnull())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ec847989e8>
```

0
1518
3036
4554
6072
7590
9108
10626
12144
13662
15180
16698
18216
19734
21252
22770
24288
25806
27324
28842
30360
c_pasAlder_r
c_kontaktOppmoteMaaned_r
c_kontakType
c_sistePLKontaktAvslutKode_c
b_kontaktPaaminingSendt
b_kontaktErDirekteTime_c
rteOHEpisoderIPeriodenPerHenv_c
eElekEpisoderIPeriodenPerHenv_c
eltTilOppmoteIPeriodenPerHenv_c
UtforteElekEpisoderIPeriodenPerPas_c
teUtfortEpisodeOver15DagerPerPas_c
eUtfortEpisodeOver15DagerPerHenv_c
n_dagerTilTimeGitt_c
n_antallPasAvbestPerPas_c
n_antallKontakterPerHenv_c
antallPasAvbestPerPasOver7Dager_c
n_andelSykAvbestPerHenv_c
n_andelPasAvbestPerPas3Dager_c
n_andelIkkeMottPerPas_c
n_kontaktVarighet
n_venteTid_c
0.08
0.04
0.00
−0.04
−0.08
d70199cac21ad3acf2ba2ab58f3

Figure 9: Heat Map to show any null values



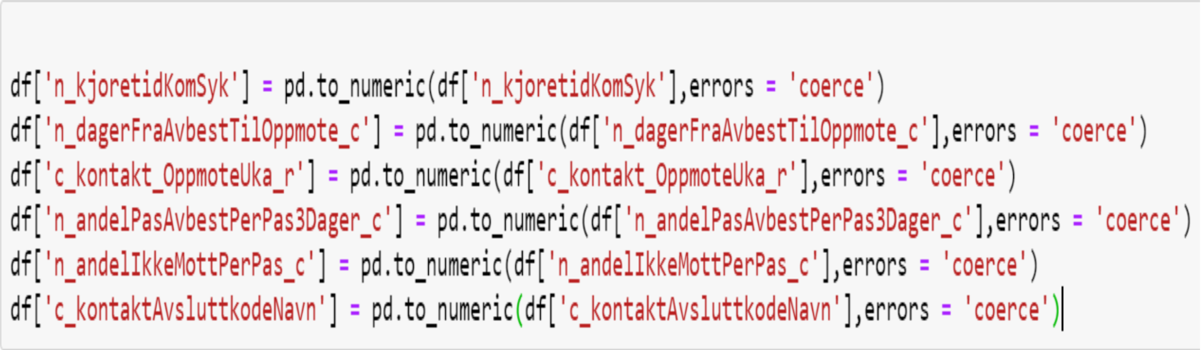
	c_pasAlder_r	c_henvType	c_henvFagomraade	c_kontaktOppmoteMaaned_r	c_kontakt_OppmoteUka_r	c_Kontakt_OppmoteTid_r	c_kontakType	c_kontakt
0	c_pasAlder_r	c_henvType	c_henvFagomraade	c_kontaktOppmoteMaaned_r	c_kontakt_OppmoteUka_r	c_Kontakt_OppmoteTid_r	c_kontakType	c_kontak
1	[20, 30[	1	2	10	6	10	2	
2	[20, 30[	1	2	5	3	10	2	

```
df.drop(df.index[0])
```

	c_pasAlder_r	c_henvType	c_henvFagomraade	c_kontaktOppmoteMaaned_r	c_kontakt_OppmoteUka_r	c_Kontakt_OppmoteTid_r	c_kontakType	c_kontakt
1	[20, 30[	1	2	10	6	10	2	
2	[20, 30[	1	2	5	3	10	2	

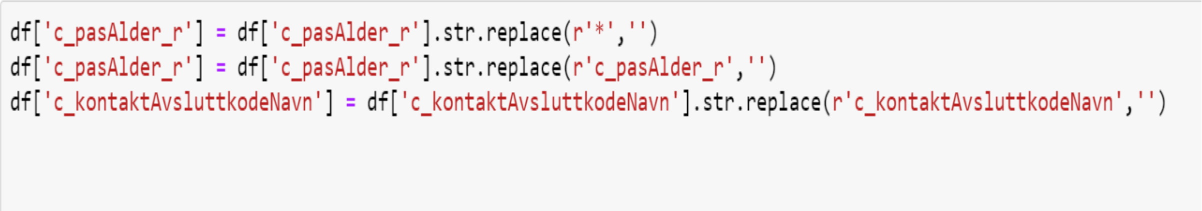
Listing 3: Shifting rows

This command we see will shift the row index upwards. Since that make, the analysis wrong, shifting row was necessary.



```
df['n_kjoretidKomSyk'] = pd.to_numeric(df['n_kjoretidKomSyk'],errors = 'coerce')
df['n_dagerFraAvbestTilOppmote_c'] = pd.to_numeric(df['n_dagerFraAvbestTilOppmote_c'],errors = 'coerce')
df['c_kontakt_OppmoteUka_r'] = pd.to_numeric(df['c_kontakt_OppmoteUka_r'],errors = 'coerce')
df['n_andelPasAvbestPerPas3Dager_c'] = pd.to_numeric(df['n_andelPasAvbestPerPas3Dager_c'],errors = 'coerce')
df['n_andelIkkeMottPerPas_c'] = pd.to_numeric(df['n_andelIkkeMottPerPas_c'],errors = 'coerce')
df['c_kontaktAvsluttkodeNavn'] = pd.to_numeric(df['c_kontaktAvsluttkodeNavn'],errors = 'coerce')
```

Listing 4: Changing data type to numeric data type



```
df['c_pasAlder_r'] = df['c_pasAlder_r'].str.replace(r'*','')
df['c_pasAlder_r'] = df['c_pasAlder_r'].str.replace(r'c_pasAlder_r','')
df['c_kontaktAvsluttkodeNavn'] = df['c_kontaktAvsluttkodeNavn'].str.replace(r'c_kontaktAvsluttkodeNavn','')
```

Listing 5: Removing dirty data using str.replace

.str.replace will replace all occurrences of dirty data that will be specified in the code ("API reference — pandas 1.2.4 documentation", 2021).

In my case for c_pasAlder_r, *, and c_pasAlder_r is included in the rows. Similarly, c_kontaktAvsluttkodeNavn is also included in c_kontaktAvsluttkodeNavn column. Therefore, these non-useful data are removed by using ‘’ in the parameters.

For analysis, all required columns need to be numeric. These data types are object data types. Before modeling, I have converted to numeric, and analysis is performed.

3.2.2.3 The class imbalance problem

Class imbalance refers to the condition in which the dataset within the class has more examples than the other.

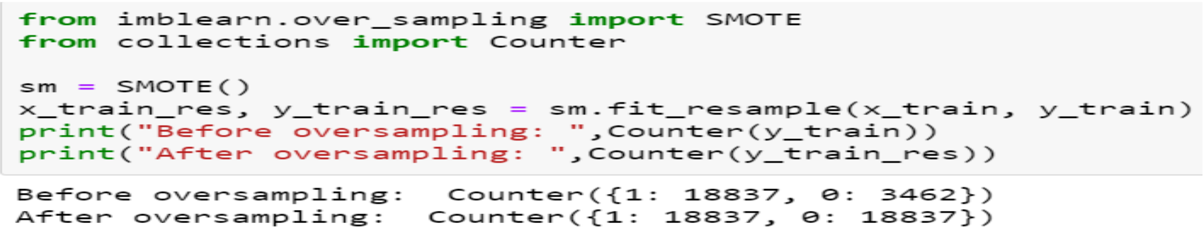
Before working on machine learning models, data need to be balanced. Imbalanced classification is predominantly tough as a predictive modeling task because of the highly skewed class distribution.

This will eventually result in penurious performance with conventional machine learning models and evaluation metrics that assume a balanced class distribution. There are different techniques to handle the unbalanced dataset namely class weights, SMOTETomek, Oversampling, and Undersampling. During my master's thesis, I have selected oversampling techniques.

In Oversampling technique, no information from training data will be lost because this technique will increase the minority class. Another advantage of this method is that its attenuates overfitting caused by oversampling. The majority class is 30 times more than the minority class. The drawback of this technique is due to duplicate data from minority class overfitting rises for some models.

However, Undersampling works by reducing the majority class drastically to match with the minority class. In my case, 30 times less than the majority class. Consequently, I have decided to use SMOTE.

SMOTE stands for Synthetic Minority Over-sampling Technique and widely used oversampling method. Data need to be balanced only for training data, testing data should not be balanced otherwise incorrect classifications can be expected with the wrong confusion matrix (Khamsan & Maskat, 2019). Imblearn is a special python package for balancing a dataset. The various resampling techniques include: Under Sampling, Oversampling, and combination of both samples ("imbalanced-learn API — imbalanced-learn 0.3.0.dev0 documentation", 2021).



```
from imblearn.over_sampling import SMOTE
from collections import Counter

sm = SMOTE()
x_train_res, y_train_res = sm.fit_resample(x_train, y_train)
print("Before oversampling: ",Counter(y_train))
print("After oversampling: ",Counter(y_train_res))
```

```
Before oversampling:  Counter({1: 18837, 0: 3462})
After oversampling:  Counter({1: 18837, 0: 18837})
```

Listing 6: Balancing the training data.

With smote, I oversample count of zero from 3462 samples to 18837 samples. A counter will count the number of Occurrences of each sample.

3.2.2.4 Accuracy Trap

It is now widely accepted that high accuracy is not necessarily an indicator of good classifier performance and therefore lies the accuracy paradox. Despite able to do a perfect job of classification error rate, high accuracy models may fail to capture pivotal information. Due to the highly imbalanced dataset accuracy is useless anyway. Sometimes, a test set might have high accuracy and perform worse than a test with lower accuracy. All the incorrect classifications are treated the same by Accuracy. False negatives and False positives are also treated the same. In medical terms, this is undesirable(Valverde-Albacete & Peláez-Moreno, 2014). For example, a person having diabetes untreated is highly undesirable than some testing that turns out to be unnecessary. In this sense, accuracy tends out to be not that useful.

3.2.2.5 Algorithms and Software

The programming language for this thesis is Python with the software Jupyter notebook. Python is easy to learn, powerful programming language. The popularity is huge because of libraries for visualization such as Matplotlib, Seaborn, and Plotly. Plotly is also a popular and powerful data visualization tool because it plots complex visualizations using the concept of data, layout, and interactive figures ("ROC and PR Curves", 2021). I have used powerful packages like Pandas, TensorFlow, Keras, NumPy, etc. Jupyter notebooks have gained so much popularity among the data science community because of their interactive, easy-to-access through the web browser. The code can be explained in the same notebook which makes easy for students and data science enthusiast. It is an interactive web-based browser and can show audio, video, text, images. It also has a facility to download notebooks in the form of pdf, ipynb files, etc. Code can be easy written in text format by markdown language (Granado & Garcia, 2021).

4 Results and Discussion

This section is divided into two sections. They are as follows: Exploratory Data analysis explaining basic statistics, summary data, and finding interesting insights. Classification Models evaluation in terms of Confusion matrix, ROC Curve, and Precision-Recall curve.

4.1 Exploratory Data Analysis

Before diving deep into modeling and evaluation, it is momentous to look at the data from different scenarios to find disentangling patterns. This can be performed by exploring the dataset, showing basic statistics, plotting insightful graphs from python visualization libraries which supports the decision-making process during the evaluation phase.

First, it is crucial to understand the target variable and have a good idea about all the classes present in that target variable. There are three classes in this target variable namely Ordinært avsluttet(Show), Pasientønsket avbest(Cancelled by patient), and Ikke møtt/ingen beskjed(No-show).

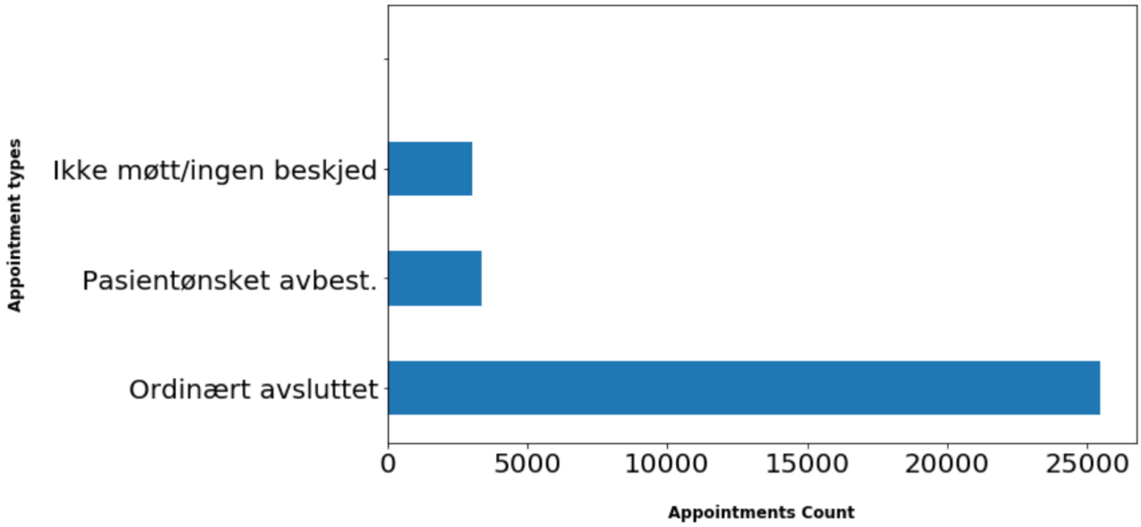
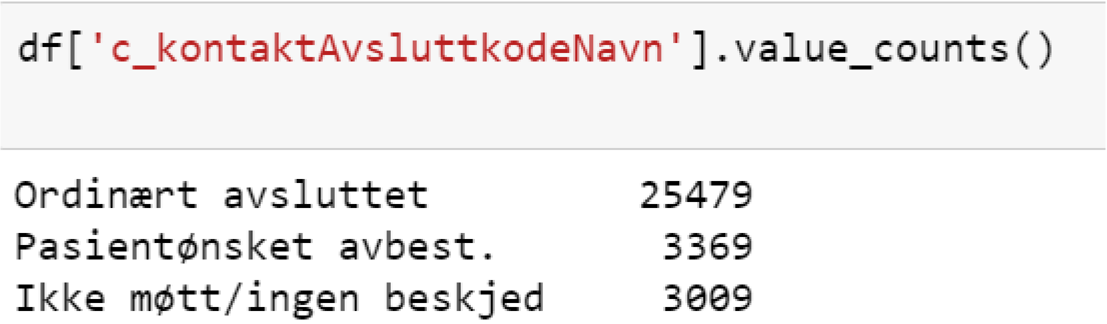


Figure 10: Appointment types and its distribution in the dataset.



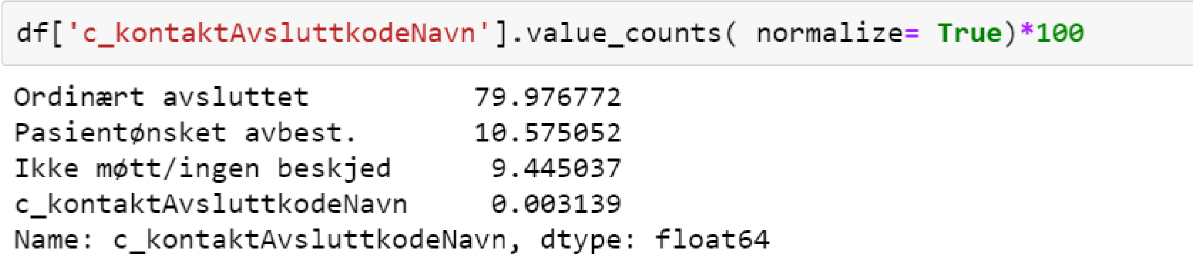
```
df['c_kontaktAvsluttkodeNavn'].value_counts()
```

```
Ordinært avsluttet          25479
Pasientønsket avbest.        3369
Ikke møtt/ingen beskjed      3009
```

Figure 11: Number of samples from each appointment types

A close look at the chart and data distribution clearly shows that number of appointment attendees is way higher than no-shows. In other words, Ordinært avsluttet tops the list with 25479 people show at the scheduled appointment. Contradictory, 3369 cancelled the appointment(Pasientønsket avbest), and 3009 missed the scheduled appointment(Ikke møtt/Ingen beskjed).

It is good practice to show the information from the dataset in percentage form.



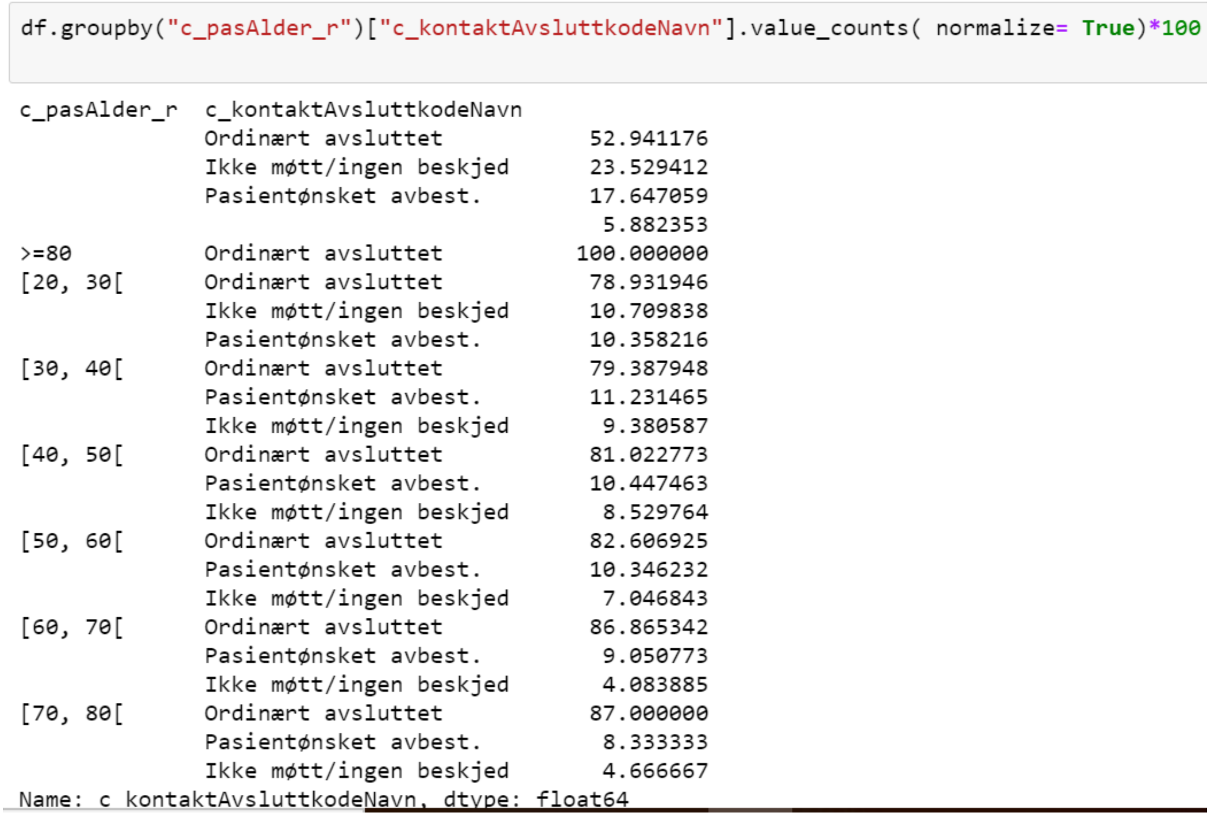
```
df['c_kontaktAvsluttkodeNavn'].value_counts( normalize= True)*100
```

```
Ordinært avsluttet          79.976772
Pasientønsket avbest.       10.575052
Ikke møtt/ingen beskjed      9.445037
c_kontaktAvsluttkodeNavn     0.003139
Name: c_kontaktAvsluttkodeNavn, dtype: float64
```

Figure 12: Appointment types in Percentage.

9.44% miss the scheduled appointment. When normalize is set to True, returns relative frequency by dividing all the values by the sum of values.

Age group is also an important variable to look at it. It is crucial to know the status of their appointment concerning age group. This statistical information helps to target which age group needs to be targeted so that they come to the scheduled appointment.



```
df.groupby("c_pasAlder_r")["c_kontaktAvsluttkodeNavn"].value_counts( normalize= True)*100
```

```
c_pasAlder_r  c_kontaktAvsluttkodeNavn
              Ordinært avsluttet           52.941176
              Ikke møtt/ingen beskjed      23.529412
              Pasientønsket avbest.        17.647059
                                            5.882353
>=80          Ordinært avsluttet          100.000000
[20, 30[      Ordinært avsluttet           78.931946
              Ikke møtt/ingen beskjed      10.709838
              Pasientønsket avbest.        10.358216
[30, 40[      Ordinært avsluttet           79.387948
              Pasientønsket avbest.        11.231465
              Ikke møtt/ingen beskjed       9.380587
[40, 50[      Ordinært avsluttet           81.022773
              Pasientønsket avbest.        10.447463
              Ikke møtt/ingen beskjed       8.529764
[50, 60[      Ordinært avsluttet           82.606925
              Pasientønsket avbest.        10.346232
              Ikke møtt/ingen beskjed       7.046843
[60, 70[      Ordinært avsluttet           86.865342
              Pasientønsket avbest.         9.050773
              Ikke møtt/ingen beskjed       4.083885
[70, 80[      Ordinært avsluttet           87.000000
              Pasientønsket avbest.         8.333333
              Ikke møtt/ingen beskjed       4.666667
Name: c_kontaktAvsluttkodeNavn, dtype: float64
```

Figure 13: Appointment types with percentage share for age-groups

I have grouped 'c_kontaktAvsluttkodeNavn' and 'c_pasAlder_r' by using groupby method. This method will group only these columns and provide the required results. It is an excellent way to filter and get all appointment types across age groups. 10.7% of patients of age-group [20-30[missed the appointments followed by 9.38% and 8.52% of age-group [30-40[, and [40-50[respectively. Conversely, 4.08% and 4.66% of patients of age group [60-70[, and [70-80[missed the appointments, respectively.

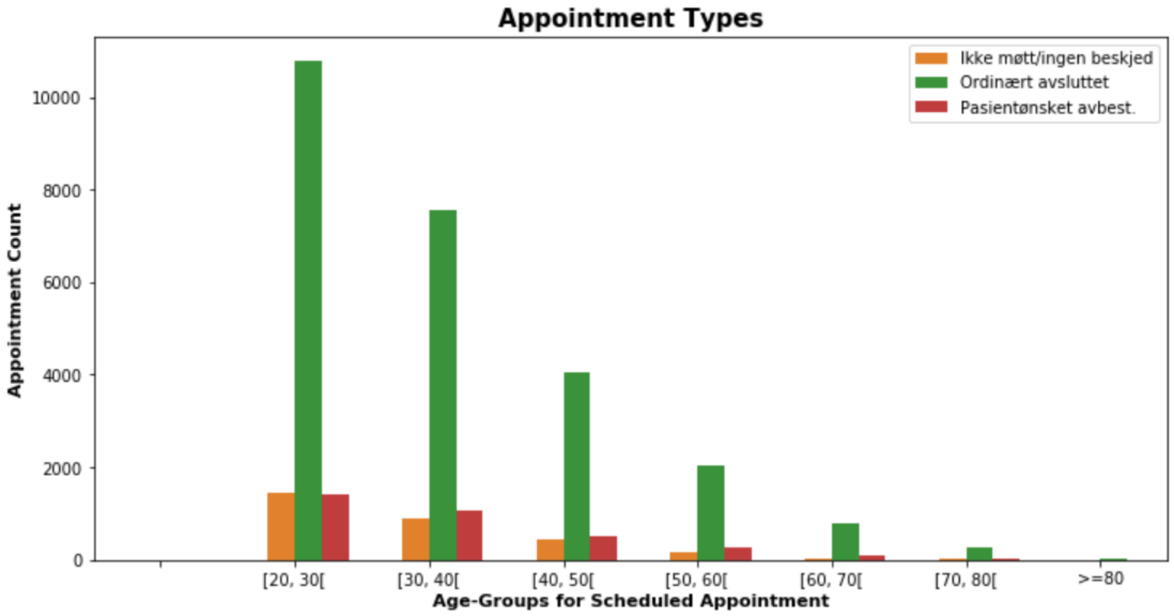


Figure 14: Comparison of appointment types.

Figure 14 elucidates all appointment types for age groups. This bar chart has some interesting trends to understand the data. There is a significant drop of Ordinært avsluttet(show). The reason is that th e dataset has more appointments from age groups from [20-30[. A total of 13651 appointments are scheduled for younger patients. Patients cancelling the appointments are also higher in the age grou p [20-30and missing the appointments are also higher in this category. Moreover, all types of appoin tments bottomed when the patient's aged increases.

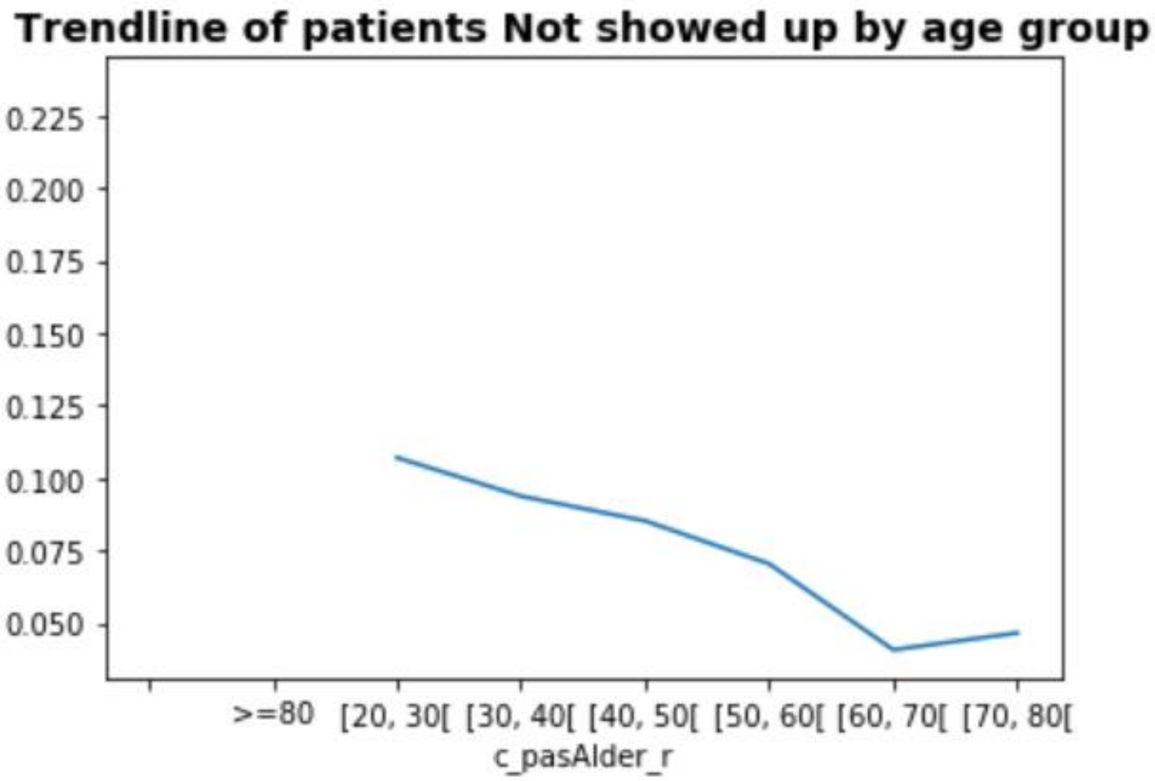
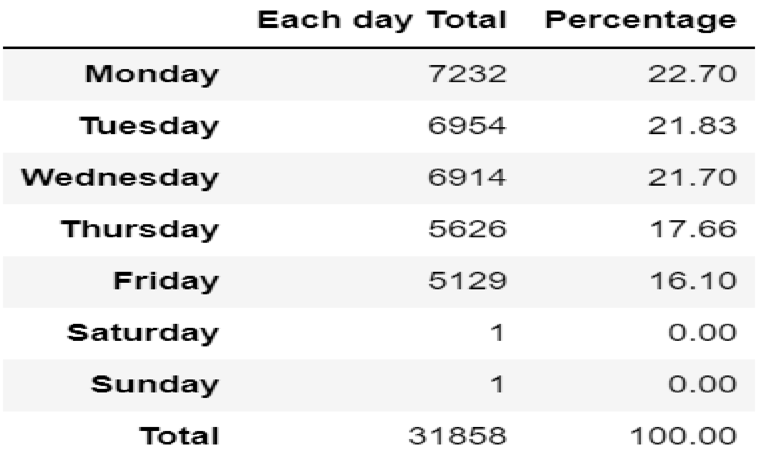


Figure 15: Trendline of no-show for age groups.

Figure 15 illuminates that, Age group [60-70[missed the least appointments with 4.08%. This can be verified from this line graph and the calculation done before for all groups in percentages. Also, the interesting trend here is, there is a slight increase in a no-show for [70-80[age group with 4.667%. Overall, there is a significant reduction of miss appointments when age gets higher.

While exploring the data, the day of the week and month are also important factors to consider. Finding insightful results is the key while performing data exploration. The results for which days and which month patients missed the appointments are useful information for business aims and objectives at the end.



	Each day Total	Percentage
Monday	7232	22.70
Tuesday	6954	21.83
Wednesday	6914	21.70
Thursday	5626	17.66
Friday	5129	16.10
Saturday	1	0.00
Sunday	1	0.00
Total	31858	100.00

Figure 16: Breakdown of appointments in each day

Figure 16 gives detailed information about each day scheduled appointment. Monday contributes 22.70% followed by Tuesday 21.83%, Wednesday 21.70% in descending order. On weekends, too few or almost no appointments are conducted.

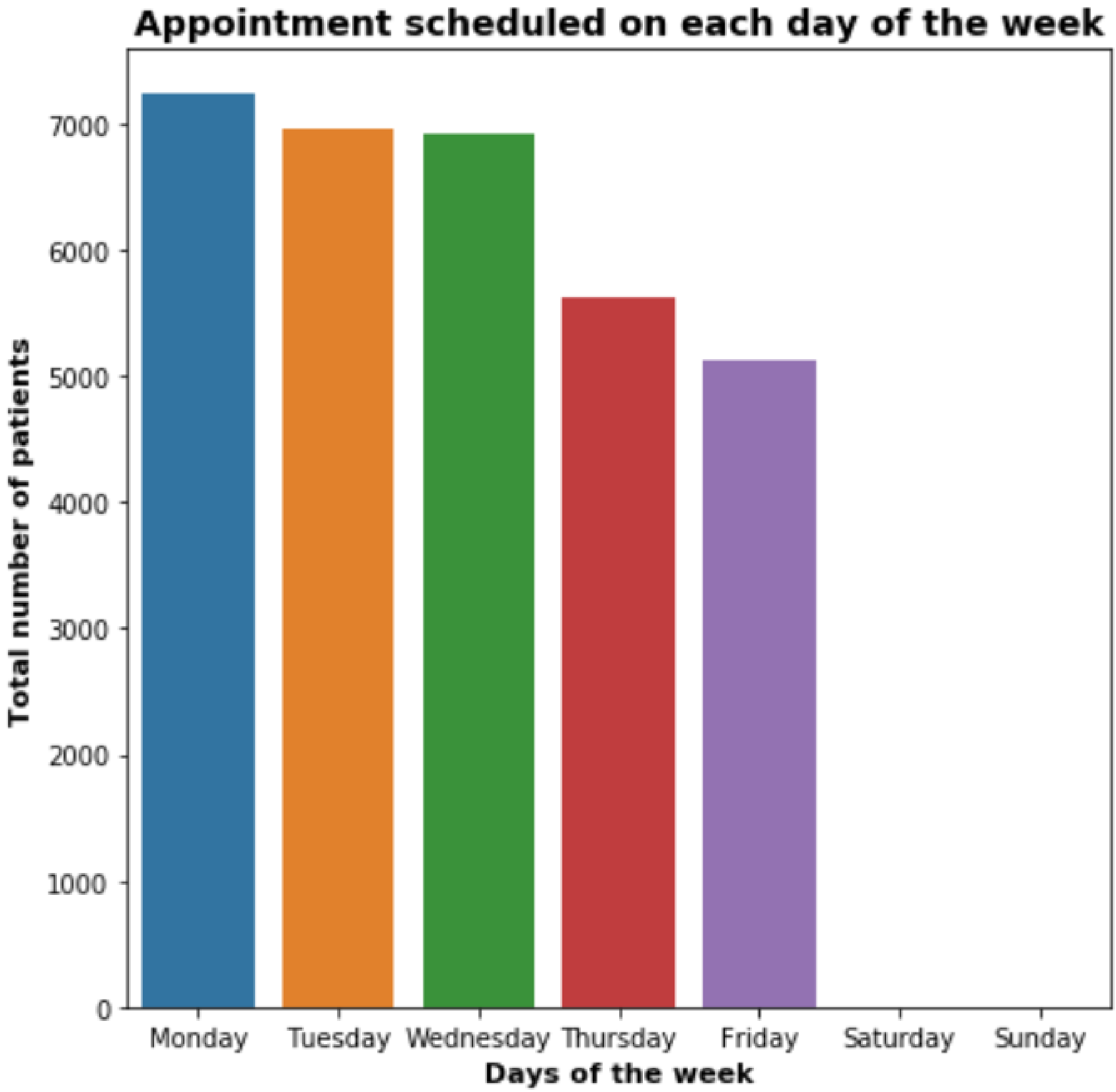


Figure 17: Bar chart to show scheduled appointments for each day.

Figure 16 and Figure 17 represent each week's scheduled appointments with the count-wise and percentage-wise separately in a bar chart.

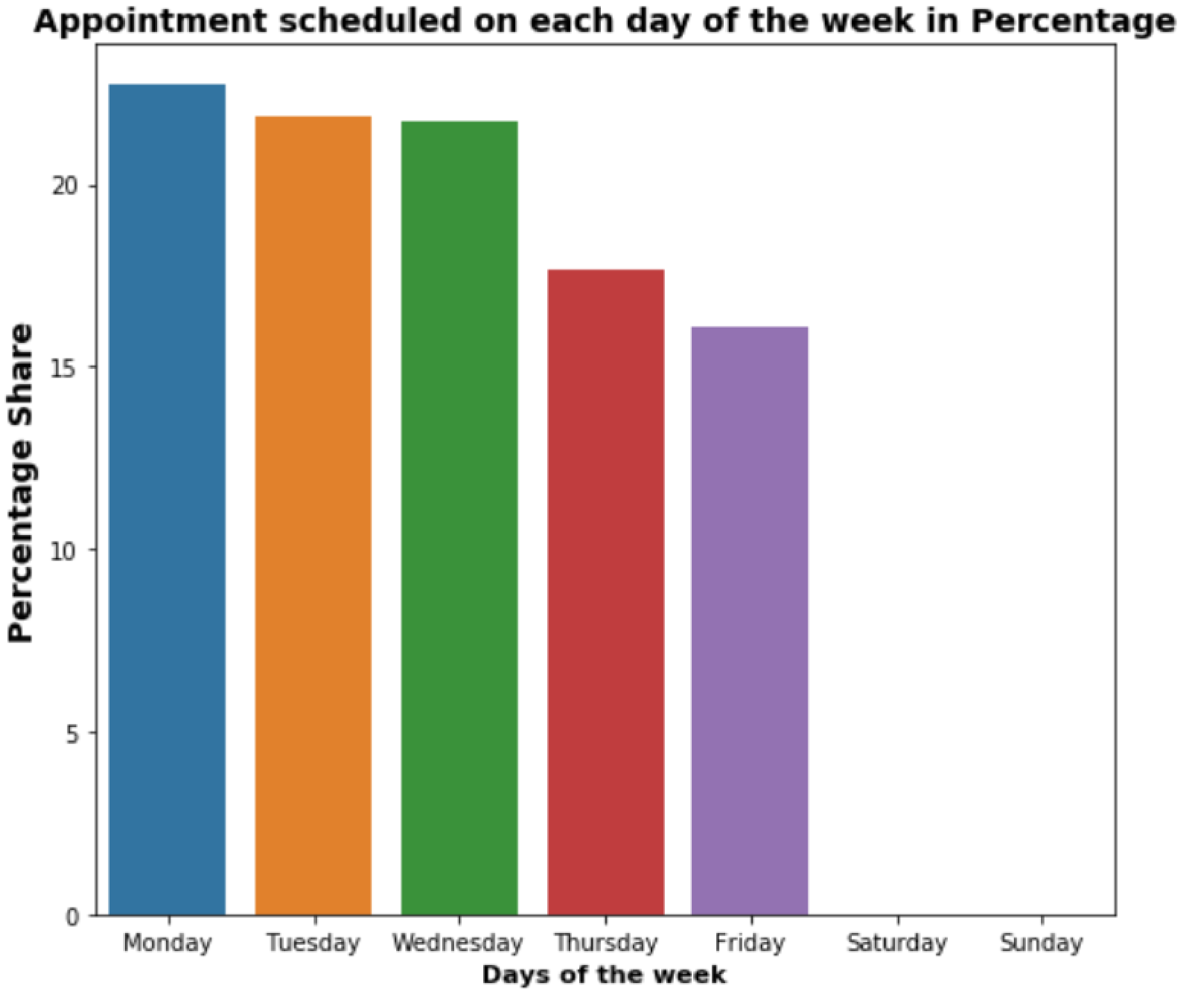


Figure 18: The percentage share of Schedule Appointment

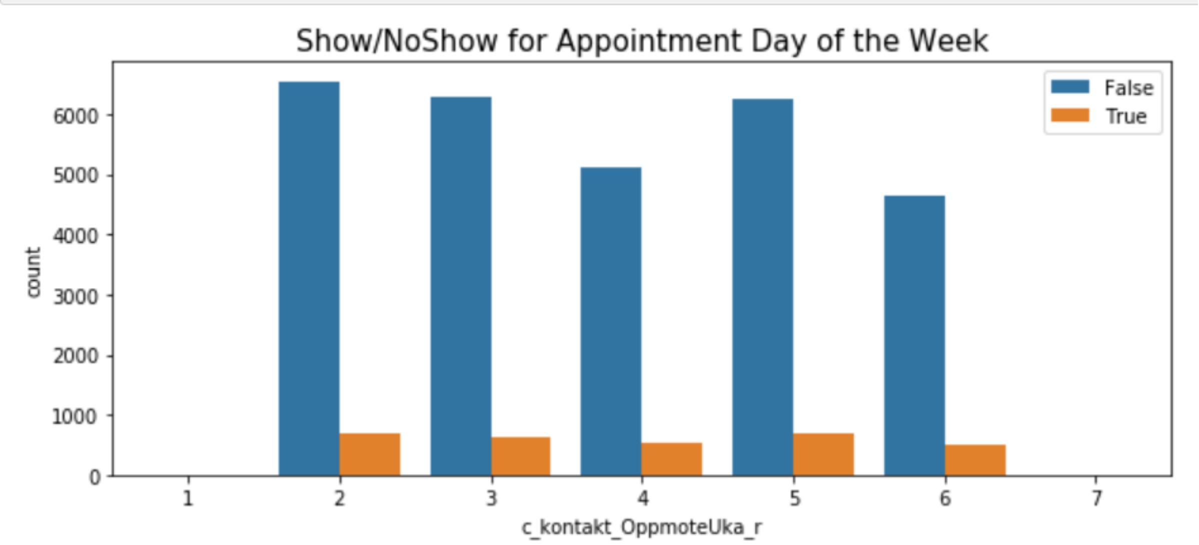
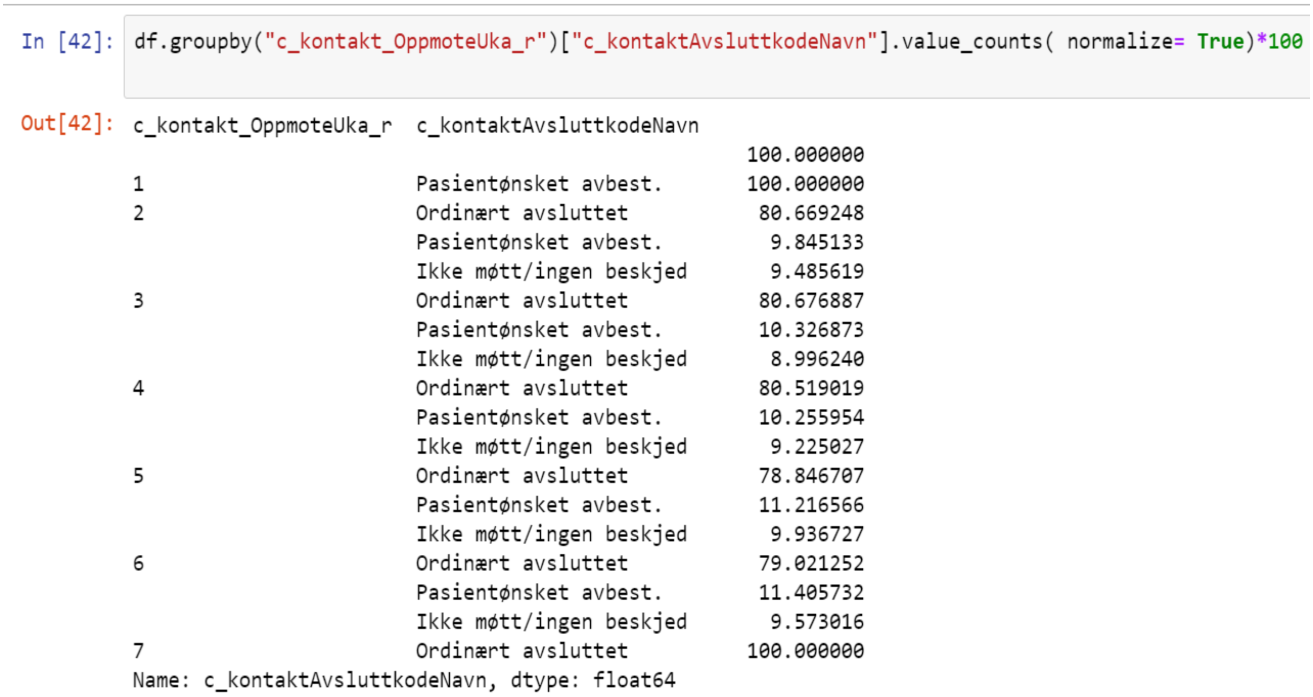


Figure 19: Show and No-show for all days of a week

However, Figure 19, provides no-show results as well. It looks like no-show contributes similarly for all days. Counting and percentage share give more knowledge about the data. Day 1 and Day 7 are Sunday and Saturday, respectively.



```
In [42]: df.groupby("c_kontakt_OppmoteUka_r")["c_kontaktAvsluttkodeNavn"].value_counts( normalize= True)*100

Out[42]: c_kontakt_OppmoteUka_r  c_kontaktAvsluttkodeNavn
                                                           100.000000
         1                       Pasientønsket avbest.      100.000000
         2                       Ordinært avsluttet          80.669248
                                 Pasientønsket avbest.        9.845133
                                 Ikke møtt/ingen beskjed      9.485619
         3                       Ordinært avsluttet          80.676887
                                 Pasientønsket avbest.       10.326873
                                 Ikke møtt/ingen beskjed      8.996240
         4                       Ordinært avsluttet          80.519019
                                 Pasientønsket avbest.       10.255954
                                 Ikke møtt/ingen beskjed      9.225027
         5                       Ordinært avsluttet          78.846707
                                 Pasientønsket avbest.       11.216566
                                 Ikke møtt/ingen beskjed      9.936727
         6                       Ordinært avsluttet          79.021252
                                 Pasientønsket avbest.       11.405732
                                 Ikke møtt/ingen beskjed      9.573016
         7                       Ordinært avsluttet         100.000000
         Name: c_kontaktAvsluttkodeNavn, dtype: float64
```

Figure 20: Distribution of appointments on weekdays

From Figure 20, the group method decorates the bar chart more precisely and informative-wise. Thursday has a percentage share of 9.93% for no-shows ensued by Friday 9.57%.

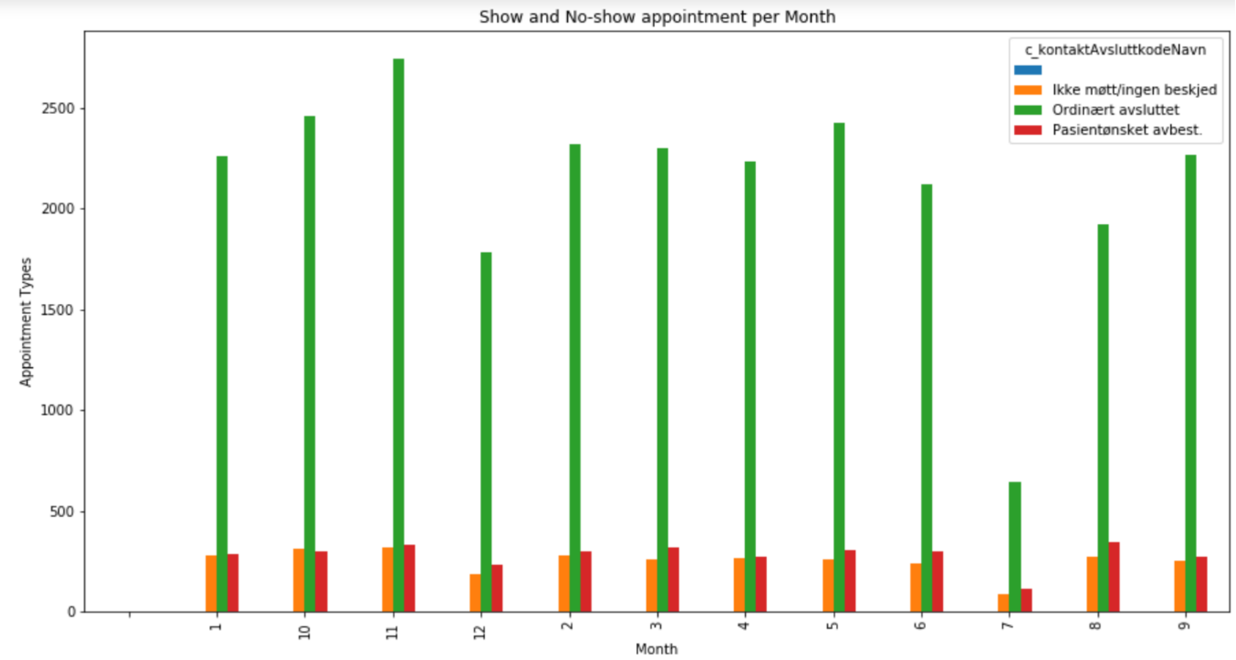
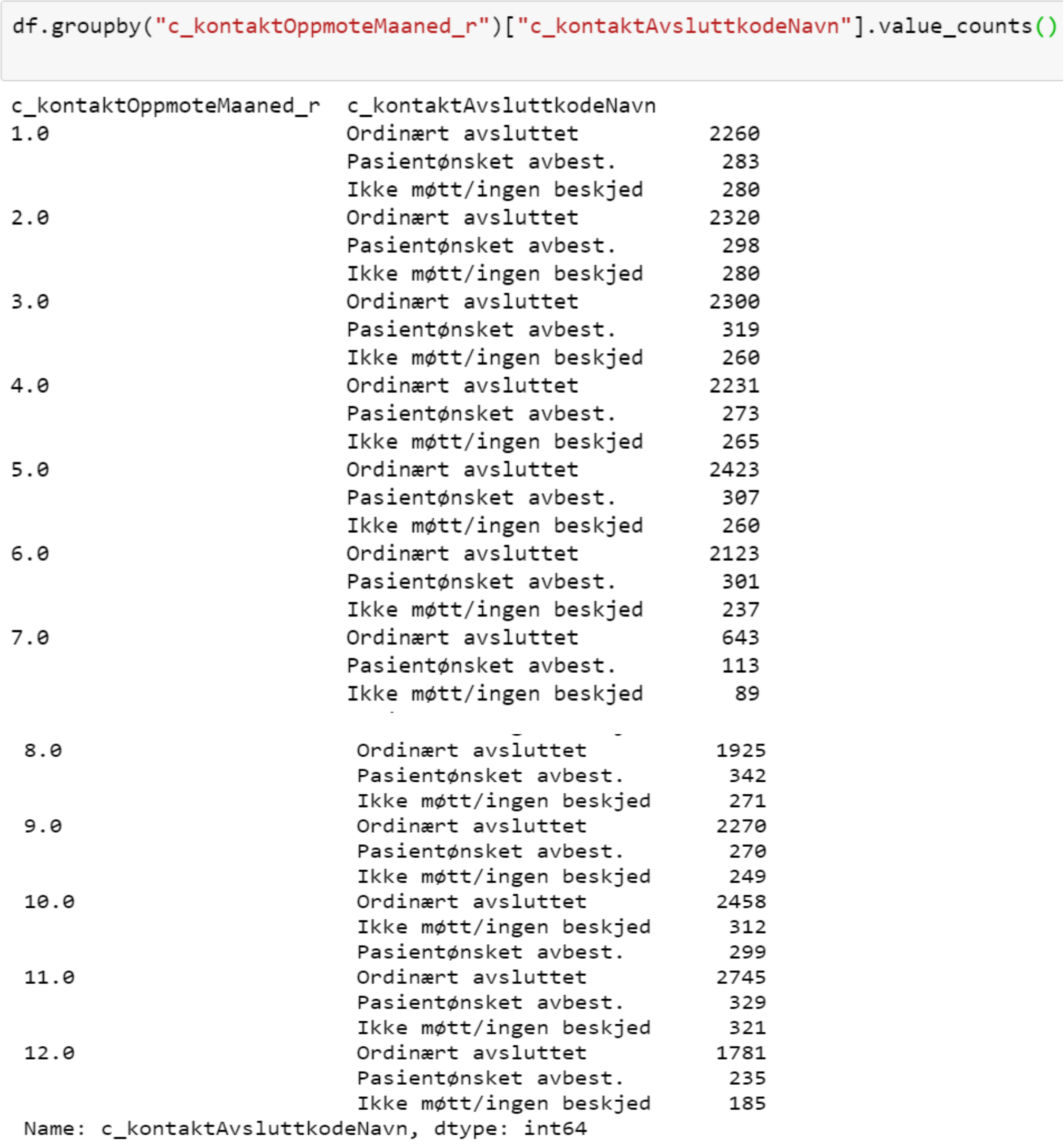


Figure 21: Appointment types per Month

According to Figure 20 and Figure 21, Month 7(July) have the least missed appointment with just 89 missed appointments and followed by 185 missed appointments in Month 12(December). However, November has the highest missed appointments with the count of 321, followed by October with the count of 312**.**



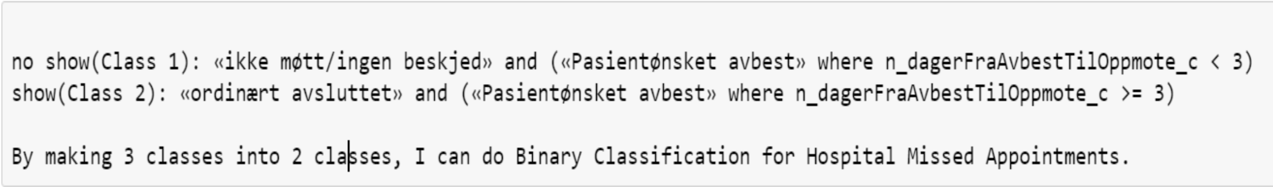
```
df.groupby("c_kontaktOppmoteMaaned_r")["c_kontaktAvsluttkodeNavn"].value_counts()
```

```
c_kontaktOppmoteMaaned_r  c_kontaktAvsluttkodeNavn
1.0                       Ordinært avsluttet          2260
                          Pasientønsket avbest.        283
                          Ikke møtt/ingen beskjed      280
2.0                       Ordinært avsluttet          2320
                          Pasientønsket avbest.        298
                          Ikke møtt/ingen beskjed      280
3.0                       Ordinært avsluttet          2300
                          Pasientønsket avbest.        319
                          Ikke møtt/ingen beskjed      260
4.0                       Ordinært avsluttet          2231
                          Pasientønsket avbest.        273
                          Ikke møtt/ingen beskjed      265
5.0                       Ordinært avsluttet          2423
                          Pasientønsket avbest.        307
                          Ikke møtt/ingen beskjed      260
6.0                       Ordinært avsluttet          2123
                          Pasientønsket avbest.        301
                          Ikke møtt/ingen beskjed      237
7.0                       Ordinært avsluttet           643
                          Pasientønsket avbest.        113
                          Ikke møtt/ingen beskjed       89
8.0                       Ordinært avsluttet          1925
                          Pasientønsket avbest.        342
                          Ikke møtt/ingen beskjed      271
9.0                       Ordinært avsluttet          2270
                          Pasientønsket avbest.        270
                          Ikke møtt/ingen beskjed      249
10.0                      Ordinært avsluttet          2458
                          Ikke møtt/ingen beskjed      312
                          Pasientønsket avbest.        299
11.0                      Ordinært avsluttet          2745
                          Pasientønsket avbest.        329
                          Ikke møtt/ingen beskjed      321
12.0                      Ordinært avsluttet          1781
                          Pasientønsket avbest.        235
                          Ikke møtt/ingen beskjed      185
Name: c_kontaktAvsluttkodeNavn, dtype: int64
```

Figure 22: Distribution of Appointments per Month

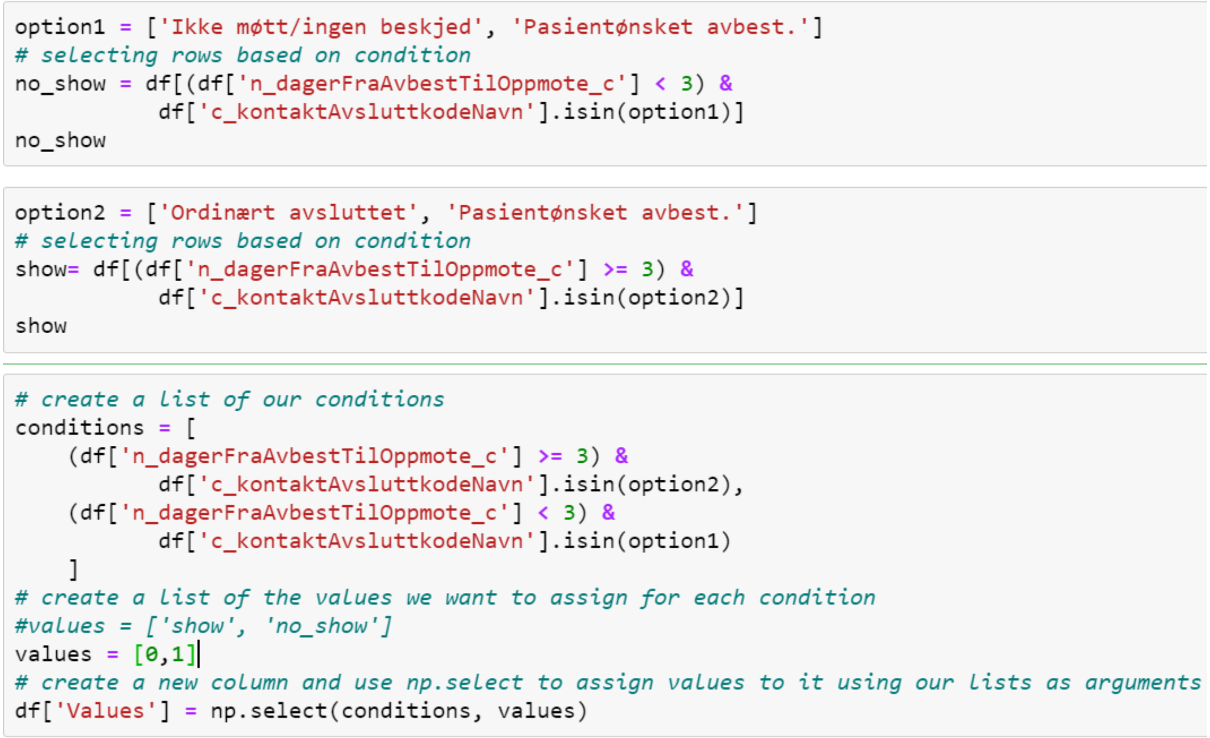
Figure 22, above, shows appointment types for all types of appointments.

Before diving into the evaluation of classification models, I need to clear that, target variable has 3 class instances, which means a multi-class classification problem. But I have transformed it into a binary classification problem which makes the complex problem a simpler one.



```
no show(Class 1): «ikke møtt/ingen beskjed» and («Pasientønsket avbest» where n_dagerFraAvbestTilOppmote_c < 3)
show(Class 2): «ordinært avsluttet» and («Pasientønsket avbest» where n_dagerFraAvbestTilOppmote_c >= 3)

By making 3 classes into 2 classes, I can do Binary Classification for Hospital Missed Appointments.
```

Listing 7: Converting 3 Class classification system to 2 class Classification System



```
option1 = ['Ikke møtt/ingen beskjed', 'Pasientønsket avbest.']
# selecting rows based on condition
no_show = df[(df['n_dagerFraAvbestTilOppmote_c'] < 3) &
           df['c_kontaktAvsluttkodeNavn'].isin(option1)]
no_show
```

```
option2 = ['Ordinært avsluttet', 'Pasientønsket avbest.']
# selecting rows based on condition
show= df[(df['n_dagerFraAvbestTilOppmote_c'] >= 3) &
           df['c_kontaktAvsluttkodeNavn'].isin(option2)]
show
```

```
# create a list of our conditions
conditions = [
    (df['n_dagerFraAvbestTilOppmote_c'] >= 3) &
           df['c_kontaktAvsluttkodeNavn'].isin(option2),
    (df['n_dagerFraAvbestTilOppmote_c'] < 3) &
           df['c_kontaktAvsluttkodeNavn'].isin(option1)
    ]
# create a list of the values we want to assign for each condition
#values = ['show', 'no_show']
values = [0,1]
# create a new column and use np.select to assign values to it using our lists as arguments
df['Values'] = np.select(conditions, values)
```

Listing 8: Way of converting to 2 class classification system.

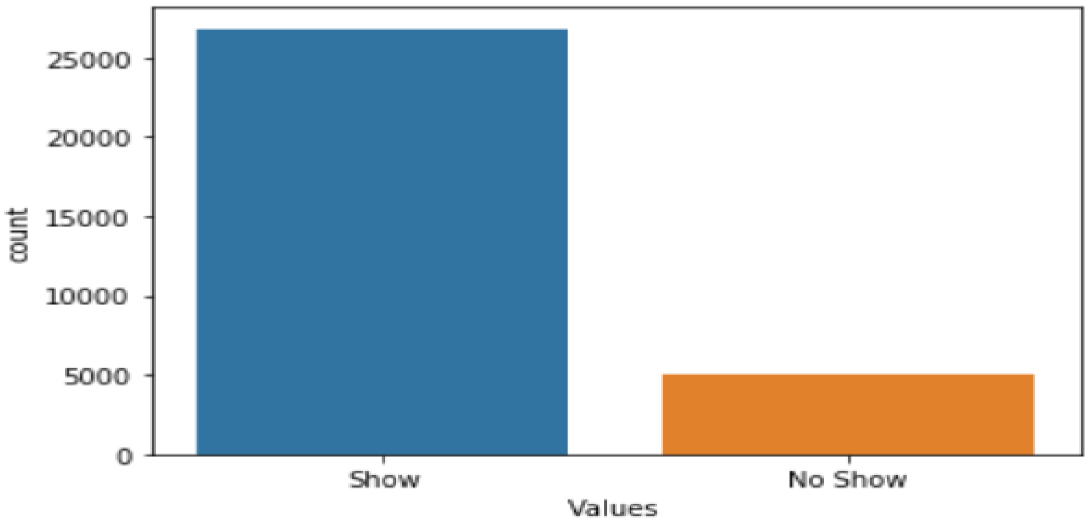
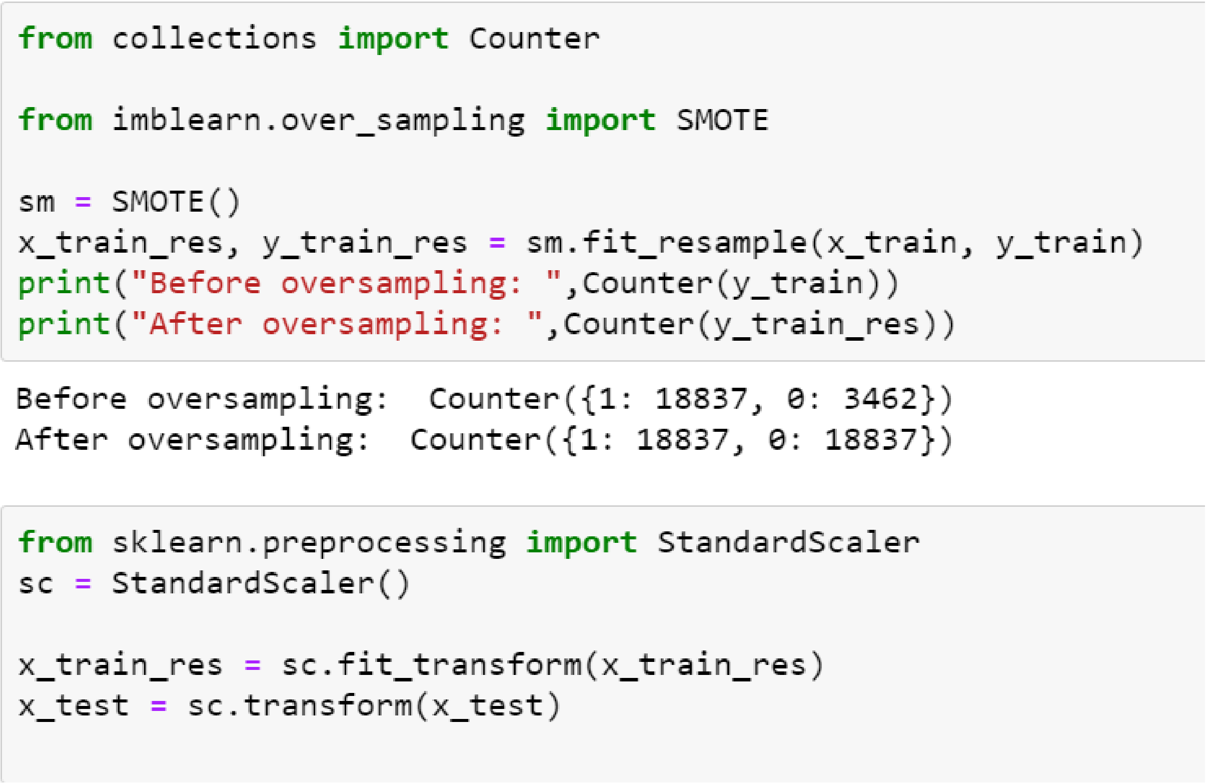


Figure 23: Binary Classification with show and no-show results

This is an imbalanced dataset. It was also cleared that from the beginning the data we have for the show was about 80% and no show was around 10%. Data need to be balanced on the training dataset and tested on the test data set. Before working on the models, it is an excellent way to scale and normalize the data.



```
from collections import Counter

from imblearn.over_sampling import SMOTE

sm = SMOTE()
x_train_res, y_train_res = sm.fit_resample(x_train, y_train)
print("Before oversampling: ",Counter(y_train))
print("After oversampling: ",Counter(y_train_res))
```

```
Before oversampling:  Counter({1: 18837, 0: 3462})
After oversampling:  Counter({1: 18837, 0: 18837})
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()

x_train_res = sc.fit_transform(x_train_res)
x_test = sc.transform(x_test)
```

Listing 9: Oversampling results and scaling the data.

The dataset is split into 70% training data and 30% testing data. By using SMOTE as a technique to handle imbalanced data and upsampling the minority class, data gets balanced. The most crucial concept is that data need to be balanced only in training data.

The next sub-chapter will be the results and evaluation of machine learning models in terms of classification will be discussed thoroughly.

4.2 Classification Models Evaluation

Evaluation of classification models is an integral part of CRISP-DM methodology. There are numerous evaluation metrics developed to understand the binary classification system. First, the accuracy score of both training and testing data, development of confusion matrix, and its associated results concerned with it such as Recall, Precision, F1-score, ROC Curve analysis, AUC ROC curve, and PR curve will be discussed in detail. Lastly, some uncommon but beneficial statistical measures will be discussed to evaluate the models.

Model Accuracy is the correct classification a model predicts divided by the total number of the prediction made. It is one of the ways to see the model performance but certainly, there are other better evaluation metrics as well. It is agile to see the model performance for both training and testing data before we go for other metrics evaluation.

Accuracy scores for Training and testing Data for Classification Models

Classification Models	Training Score	Testing Score
Logistic Regression	62.93%	67.84%
K Nearest Neighbor	71.1%	63.68%
Decision Tree	71.13%	68.96%
Random Forest	64.91%	62.01%
Gradient Boosting	64.35%	66.07%
GaussianNB	60.49%	74.35%

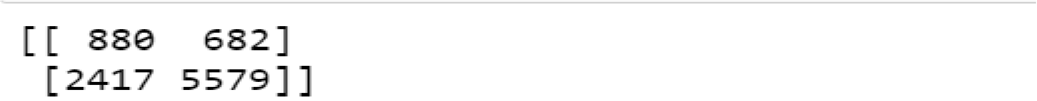
Figure 24: Accuracy Score for Training and Testing data for classification System

A confusion matrix elucidates a synopsis of predictive calculation in a classification problem. Correct and incorrect predictions are shown in a table of 2*2 matrix with their values and broken down by each class 0 and 1. Most of the classification evaluations rely on a confusion matrix to assess the performance of models and try to juxtapose them with the business objectives of the company.

The decision tree has the highest training score of 71.13% followed by K Nearest Neighbour with 71.1%. On the other hand, GaussianNB has the highest testing score of 74.35%. Subsequently, Decision Tree has the second-highest testing accuracy score of 68.96%.

Nonetheless, it is a goof to rely on accuracy score on training and testing data because of an imbalanced dataset. Although we have the balanced data, it is oversampled already.

The Confusion Matrix of all the models are presented below:



```
[[ 880  682]
 [2417 5579]]
```

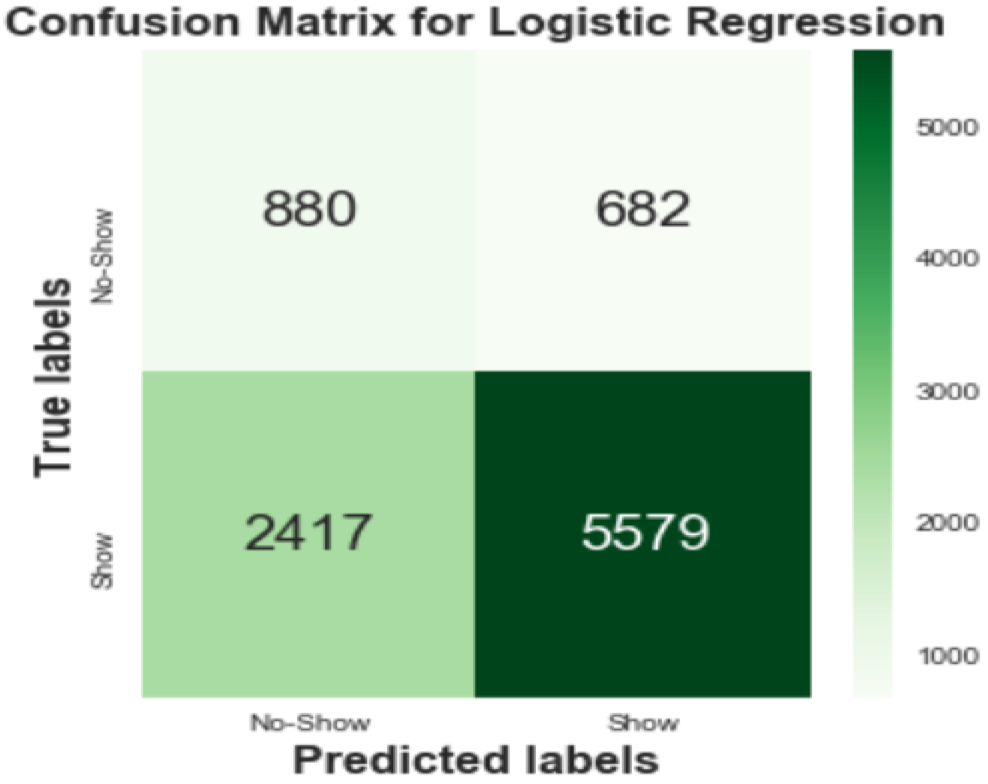
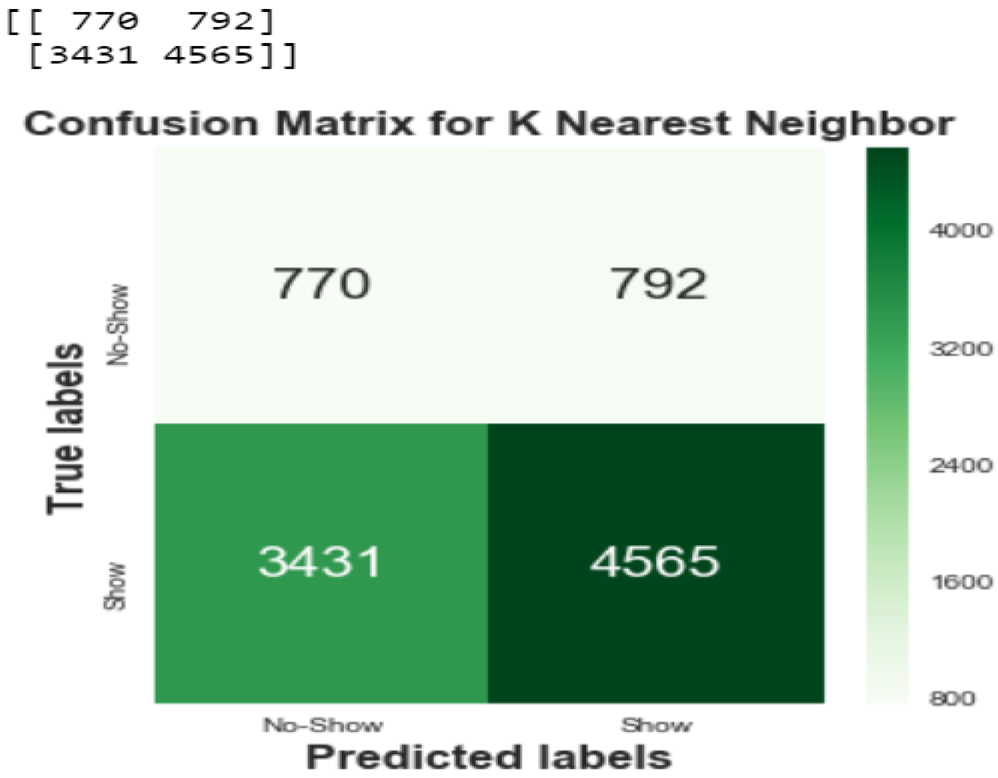


Figure 25: Confusion Matrix for Logistic Regression

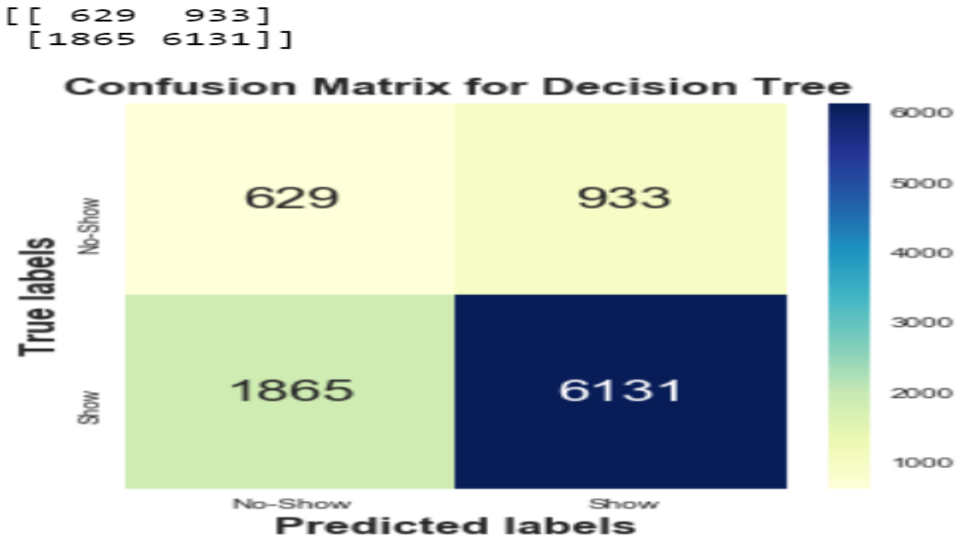
Figure 25 portrays the confusion matrix for Logistic Regression. 2.2.1 sub-chapter has detailed information about definitions of all four quadrants of a confusion matrix as well. 5579 samples have been predicted correctly as show and in actual it is a show, and it is termed as True Positive. 880 samples have been precited no-show and in actual it is no- show and it is called True Negative. 682 samples are actual no-shows but predicted show termed as False Positive. 2417 samples are actual show but predicted no-show is called as False Negative.



```
[[ 770  792]
 [3431 4565]]
```

Figure 26: Confusion Matrix for K nearest neighbor

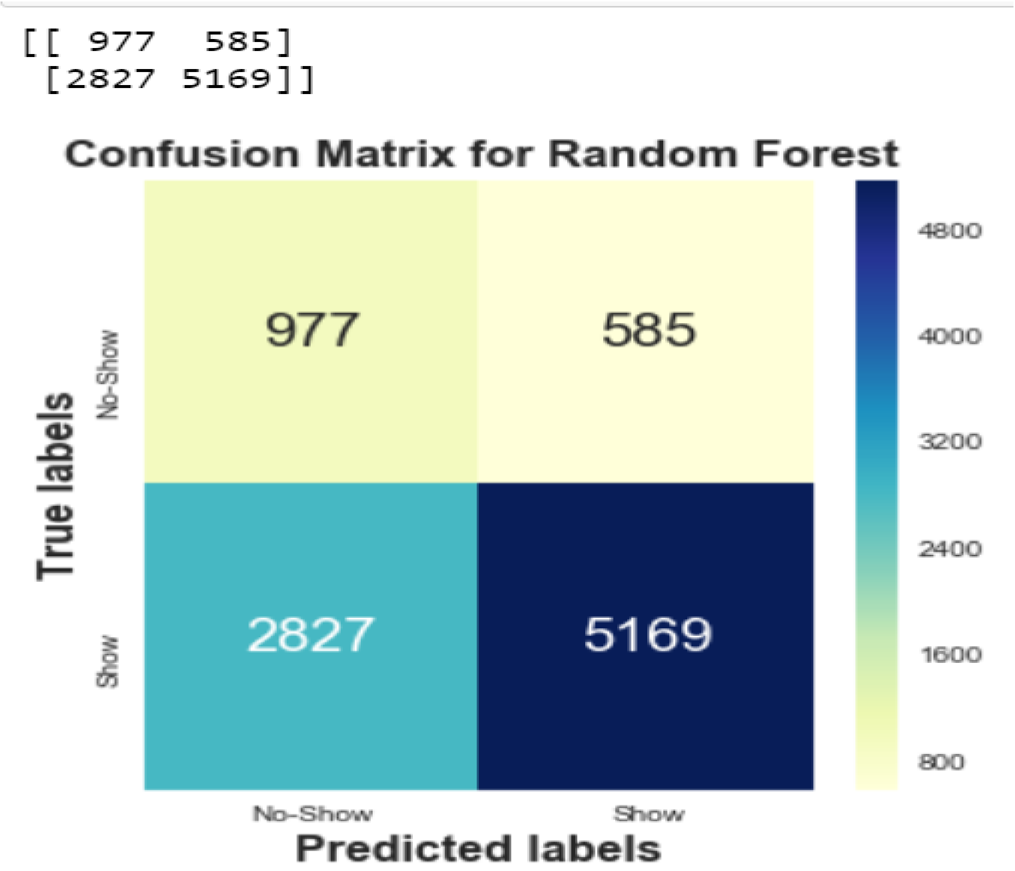
Figure 26 portrays the confusion matrix for K Nearest Neighbor. 2.2.1 sub-chapter has detailed information about definitions of all four quadrants of a confusion matrix as well. 4565 samples have been predicted correctly as show and in actual it is a show, and it is termed as True Positive. 770 samples have been precited no-show and in actual it is no- show and it is called True Negative. 792 samples are actual no-shows but predicted show termed as False Positive. 3431 samples are actual show but predicted no-show is called as False Negative.



```
[[ 629  933]
 [1865 6131]]
```

Figure 27: Confusion Matrix for Decision Tree

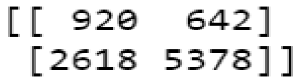
Figure 27 explicate the confusion matrix for the Decision Tree classifier. 2.2.1 sub-chapter has detailed information about definitions of all four quadrants of a confusion matrix as well. 6131 samples have been predicted correctly as show and in actual it is a show, and it is termed as True Positive. 629 samples have been precited no-show and in actual it is no- show and it is called True Negative. 933 samples are actual no-shows but predicted show termed as False Positive. 1865 samples are actual show but predicted no-show is called as False Negative.



```
[[ 977  585]
 [2827 5169]]
```

Figure 28: Confusion Matrix for Random Forest

Figure 28 depicts the confusion matrix for Random Forest. 2.2.1 sub-chapter has detailed information about definitions of all four quadrants of a confusion matrix as well. 5169 samples have been predicted correctly as show and in actual it is a show, and it is termed as True Positive. 977 samples have been predicted no-show and in actual it is no- show and it is called True Negative. 585 samples are actual no-shows but predicted show termed as False Positive. 2827 samples are actual show but predicted no-show is called as False Negative.



```
[[ 920  642]
 [2618 5378]]
```

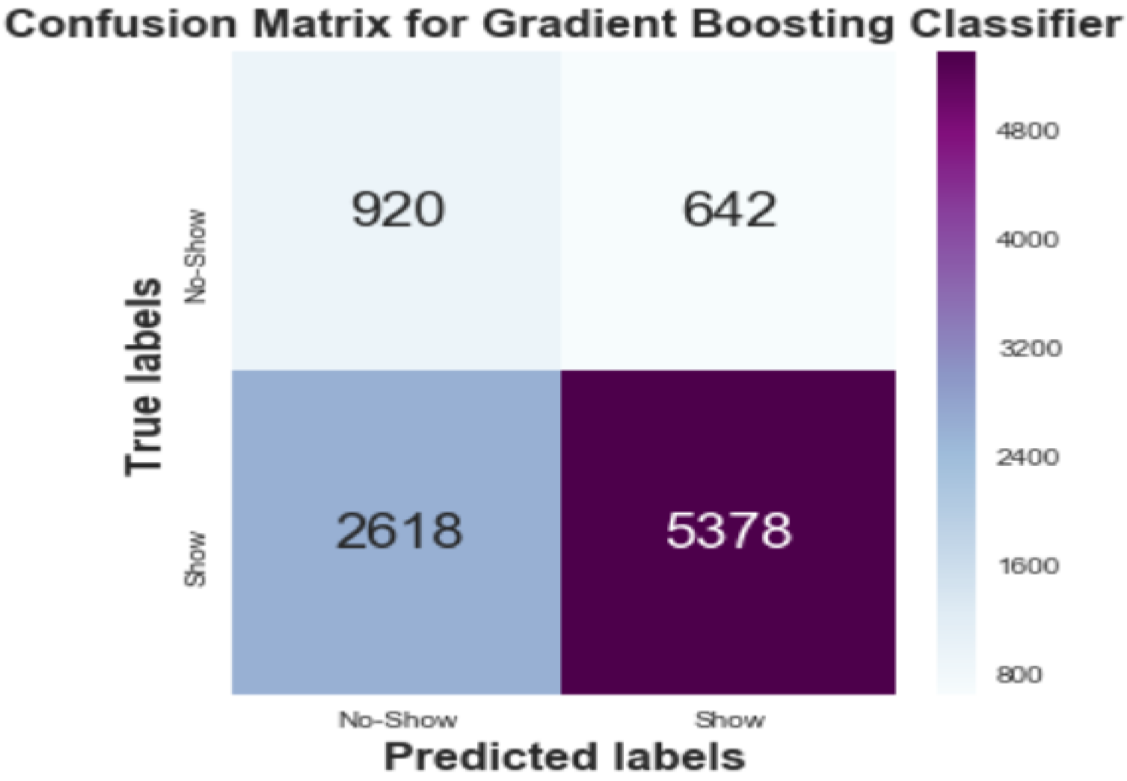
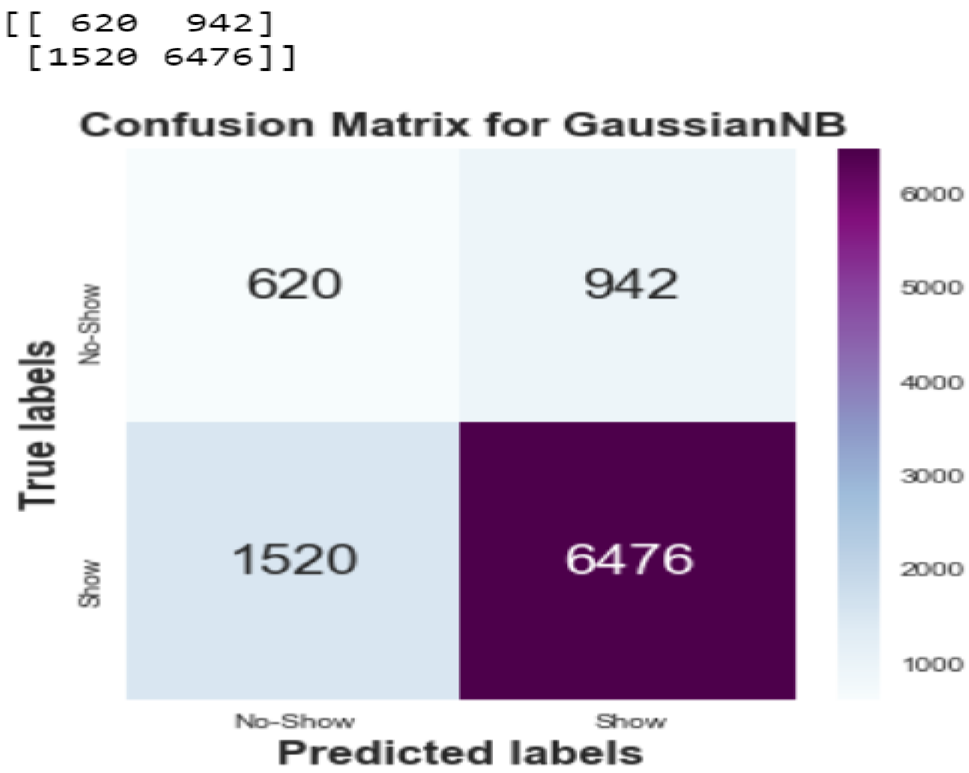


Figure 29: Confusion Matrix for Gradient Boosting Classifier

Figure 29 represents the confusion matrix for Gradient Boosting Classifier. 2.2.1 sub-chapter has detailed information about definitions of all four quadrants of a confusion matrix as well. 5378 samples have been predicted correctly as show and in actual it is a show, and it is termed as True Positive. 920 samples have been precited no-show and in actual it is no- show and it is called True Negative. 642 samples are actual no-shows but predicted show termed as False Positive. 2618 samples are actual show but predicted no-show is called as False Negative.



```
[[ 620  942]
 [1520 6476]]
```

Figure 30: Confusion Matrix for GaussianNB Classifier

Figure 30 portrays the confusion matrix for GaussianNB. 2.2.1 have detailed information about definitions of all four quadrants of a confusion matrix as well. 6476 samples have been predicted correctly as show and in actual it is a show, and it is termed as True Positive. 620 samples have been precited no-show and in actual it is no- show and it is called True Negative. 942 samples are actual no-shows but predicted show termed as False Positive. 1520 samples are actual show but predicted no-show is called as False Negative.

4.3 Adjusting Classification Threshold

The classification models return probability outputs that are instantly converted into classes by using a threshold probability. The default value for the threshold is 0.5, which means that a probability above 0.5 means positive class, and a probability below 0.5 indicates negative class. However, each problem must find its optimal threshold. It is free to choose a threshold according to business aims and objectives. A threshold is a trade-off between True Positive Rate and False Positive Rate. Changes in threshold give different AUCROC scores. TPR and FPR are inversely proportional to each other meaning if one increases other decreases and vice-versa (Saito & Rehmsmeier, 2015). The adjustment of the threshold depends on some conditions or results we obtained from precision and recall. The conditions are as follows:

- Precision ≈ Recall: When precision is almost equal or equal to recall, it means the number of False Positives is equal to the number of False Negatives. Alternatively, the number of no-shows that were incorrectly classified as the show is equal to the number of shows that were incorrectly classified as no-shows.

- High precision and low Recall: When we have high precision with low recall, the selected model will not detect many no-shows, but it detects then the model is reliable. This means resources will be used wasting money and time whereas, there will be less overbooking as it provides no-show information. This condition brings customer satisfaction results too.

- High Recall and Low Precision: when these results appear; it means most no-shows are detected with some shows as no-shows. The system will make more over-booking of the patients whereas resources such as money, time are utilized as well. This will subsequently make long waiting lists with falling customer satisfaction.

Models	Precision	Recall	F1-Score	MCC
Logistic regression	89.19	69.84	0.783	0.208
KNN	86.21	69.66	0.77	0.099
Decision Tree	86.52	74.67	0.80	0.124
Random Forest	89.56	63.14	0.79	0.20
Gradient Boosting	89.39	66.33	0.762	0.199
GaussianNB	87.28	80.87	0.84	0.182

Figure 31: Classification Models Evaluation

A closer look at Figure 31 elucidates more detailed ways to evaluate the models. If we compare the precision, Random Forest tops the list with 89.56 followed by Gradient Boosting with 89.39. KNN holds the least precision among all the models I have included for evaluation.

If we see the recall column in this table, GaussianNB is ranked first with 80.87 followed by Decision Tree with 74.67. Random Forest has the lowest recall among selected models with just 63.14.

It should be noted that Precision and Recall are expressed in terms of Percentage in the table.

F1-score is the harmonic mean of precision and recall. A low F1 score means low precision and low recall. The F1-score I obtained from the models is satisfactory. GaussianNB has the highest F1-score of 0.84. The model with the least F1-score is Gradient Boosting with a score of 0.762.

The last column of the table is MCC which stands for Matthews Correlation Coefficient. This is a widely used single evaluation metric over the accuracy, F1-Score, and other scoring metrics. It has a range from -1 to +1. With a coefficient score of 0.208, logistic regression has the highest MCC followed by Random Forest with an MCC score of 0.20. KNN has a minimum score of 0.099.

A clear conclusion from the Figure 31 is that the KNN model is the least performing model when evaluation all the classes. Several metrics have been tested and scores are low for this classification model.

Models	AUC-ROC Score	PR Score	Error Rate
Logistic regression	0.663	0.87	0.32
KNN	0.553	0.85	0.35
Decision Tree	0.589	0.86	0.307
Random Forest	0.665	0.88	0.38
Gradient Boosting	0.663	0.88	0.34
GaussianNB	0.662	0.87	0.25

Figure 32: Result of AUC-ROC curve, PR score, and Error Rate

Figure 32 is the list of AUC-ROC curves, PR curves, and error rate comparisons among all the classification models.

Random Forest has the highest AUC-ROC score of 0.665 followed by Logistic Regression and Gradient Boosting with the same score of 0.663. KNN is the feeblest classification model with an AUC-ROC score of 0.553.

When it comes to the imbalanced dataset, the Precision-Recall curve becomes an appropriate evaluation method to evaluate all the models. This curve will not use any True Negatives and make the analysis suitable if True negatives are large.

“If the threshold limit was previously set too high, the new results may all be true positives, which will increase precision. If the previous threshold was about right or too low, further lowering the threshold will introduce false positives, decreasing precision” ("Precision-Recall — sci-kit-learn 0.24.2 documentation", 2021).

The documentation for sklearn.metrics.average_precision_score states, “AP summarizes a precision-recall curve as the weighted mean of precision achieved at each threshold, with the increase in recall from the previous threshold used as the weight” ("Precision-Recall — sci-kit-learn 0.24.2 documentation", 2021). This means Average Precision is a kind of weighted-average precision across all thresholds.

Random Forest and Gradient Boosting have a PR score of 0.88 which is an excellent score without True negatives. KNN has a PR score of 0.85.

When it comes to incorrect classification error rate comes into play. The error rate has a score from 0 to 1. Rate near to 0 is best and rate near 1 is worst. It is calculated as the total number of two incorrect predictions divided by total samples. The two incorrect predictions are FP and FN.

GaussianNB has the least error rate with 0.25 which is best among all the models. It means around 75%, the model is good at predicting the actual correct values and actual negative values. The second-best model in terms of less error rate is Decision Tree with a score of 0.307. Random Forest has the most error rate of 0.38.

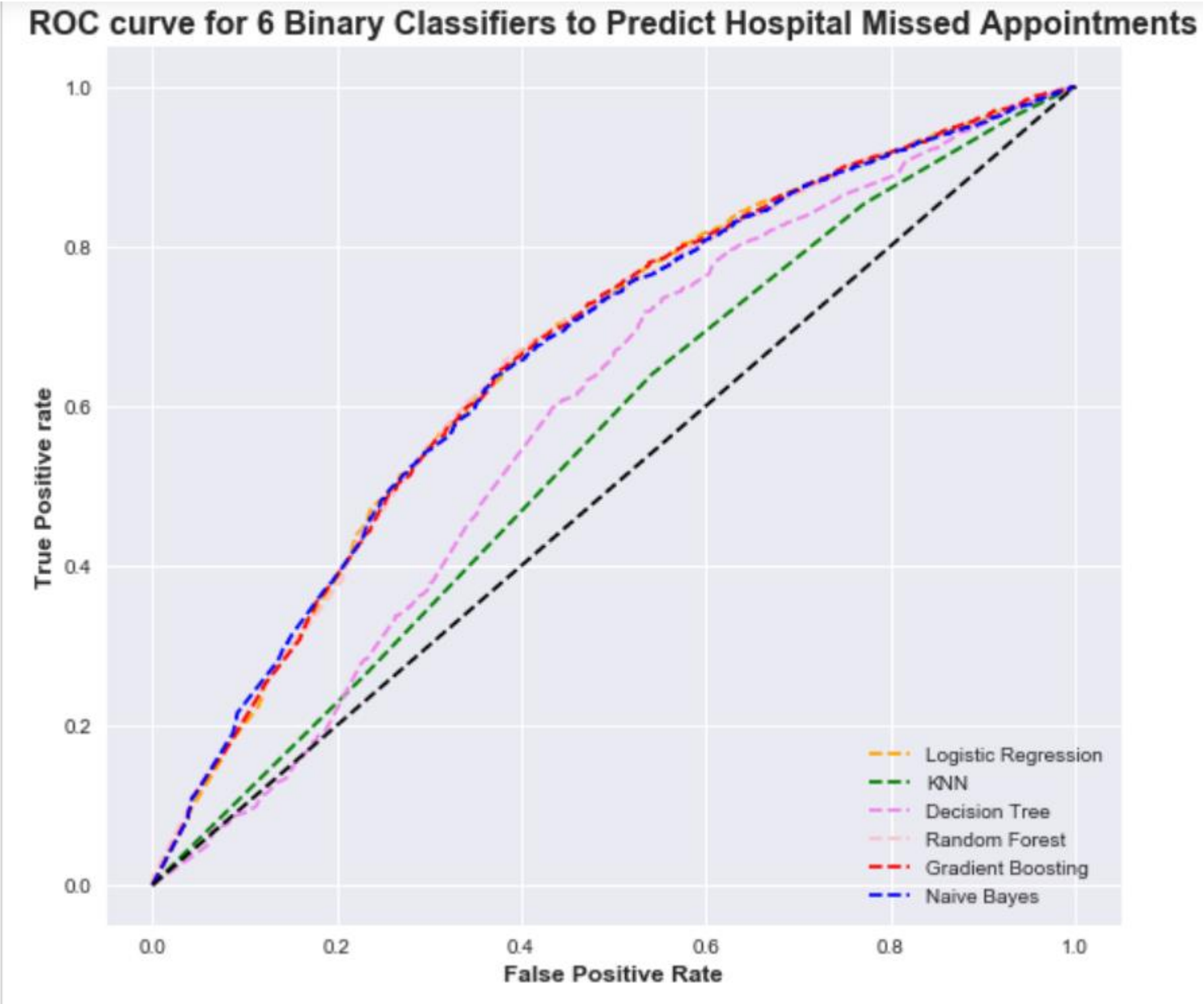


Figure 33: ROC Curve for 6 Binary Classification models

Figure 33 is the plot between True Positive Rate and False Positive Rate. Six different binary classifiers are plotted in this single plot. The score results are already discussed in Figure 32. The best model will have more space on the lower right of the curve which is called Area Under Curve. The higher the AUC the better is the model at predicting true classes and False classes.

When AUC = 1, the model is perfectly identifying correct positive and correct negative classes. However, in a real-world scenario, it is impossible to have this situation. Dataset when gets collected becomes unbalanced and bias which results in having chances of AUC = 1, practically impossible.

It is evident from the plot that the AUC for the Random Forest ROC curve is higher than that for the rest of the ROC curve. Ultimately, I can say that Random Forest did a better job of classifying the positive class in the dataset.

Before, jumping to a conclusion, I need to keep in mind that my dataset is unbalanced and need to study further special metrics called as Precision-Recall curve. I will now focus more on curves formed by using precision and recall only. This curve will rule out True Negatives.

PR curve will use different probability thresholds to summarize the trade-off between precision and recall. For a specific value of recall, precision is plotted and gives the beautiful downward plot. The more space on the lower-left the better the model at evaluating the model.

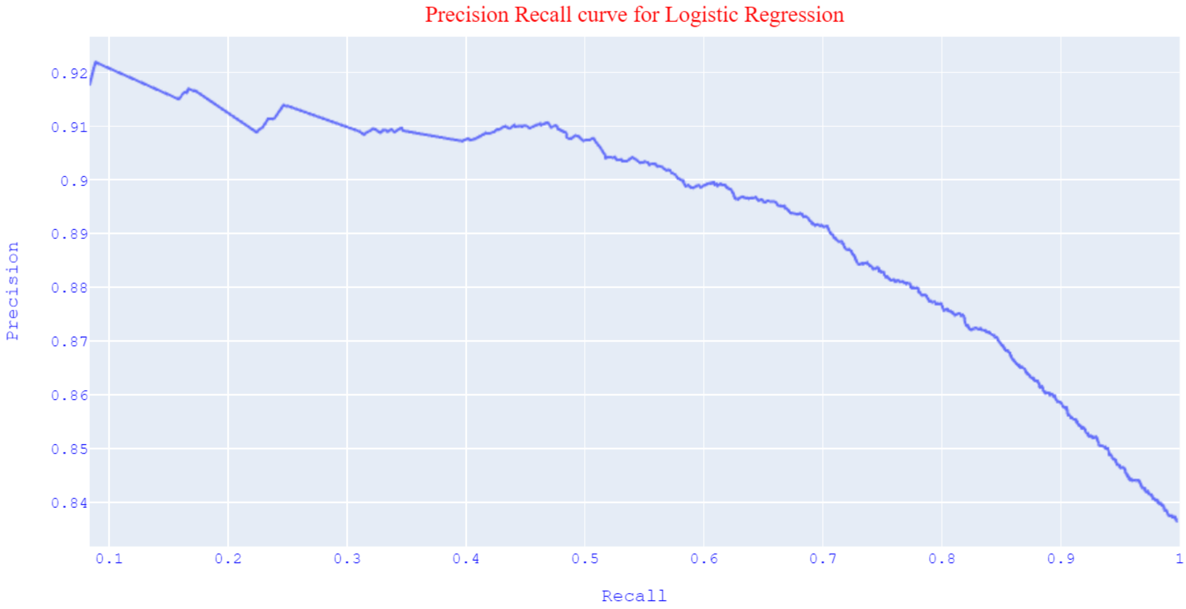


Figure 34: Precision-Recall curve for Logistic Regression

In figure 34, I plotted the PR curve with precision on X-axis and Recall on Y-axis. Precision is plotted against Y-axis. Precision dropped from 1 to 0.92 when the recall was 0. There is slight stability and a slight drop as recall increases. When recall goes towards 1, precision becomes 0.84. This is typically a good PR curve because I have achieved average precision of 0.87. The perfect score for PR is 1. When the recall was 0.3 to 0.4, precision was more stable than any period while plotting the PR curve**.** The area on the lower left also confirms how good the model is. For different thresholds, specific precision and recall can be achieved. With a 0.6 threshold value, 92% precision is obtained but recall is just 0.008%. It means in that threshold the predicted values were 92% correctly predicted as a show and the values were the actual show. When I take the cursor along the line, I get different thresholds with different precision and recall. The desired precision and recall can be obtained with the thresholds we choose. So, choosing the threshold is crucial for business objectives. If the higher recall is the concern, the threshold needs to decrease. The python notebook has the feature to move the cursor to see all thresholds with corresponding precision and recall.

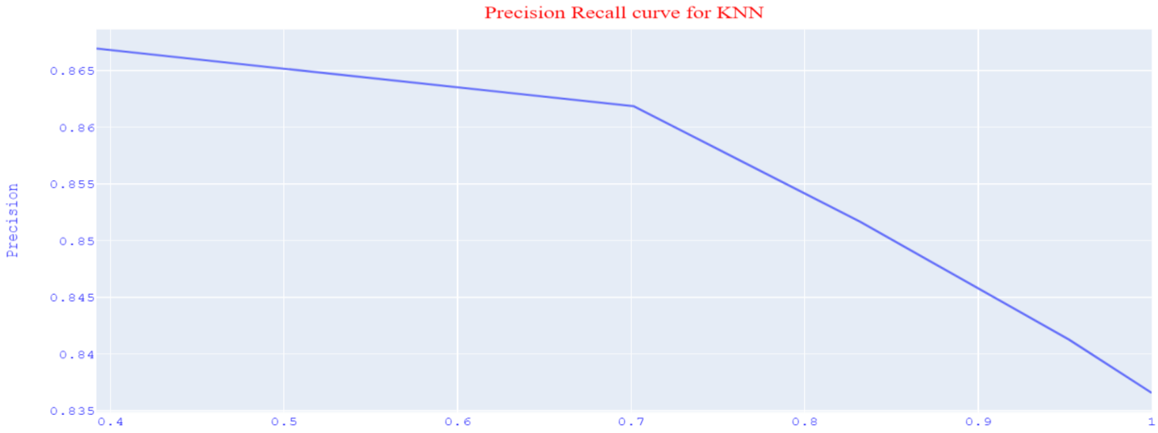


Figure 35: Precision-Recall curve for KNN

Just exploring this PR curve, this model is not good at all or worst among all. The lower left area for this curve is a minimal area meaning for specific recall precision is low most of the time. This infers model is not good at predicting actual positive and actual negative class. When the threshold was set at 1, it gives a precision score of 0.86 and recalls a very low score of 0.39. With a threshold of 0.75 precision score of 0.86 and a recall score of 0.7 were obtained.

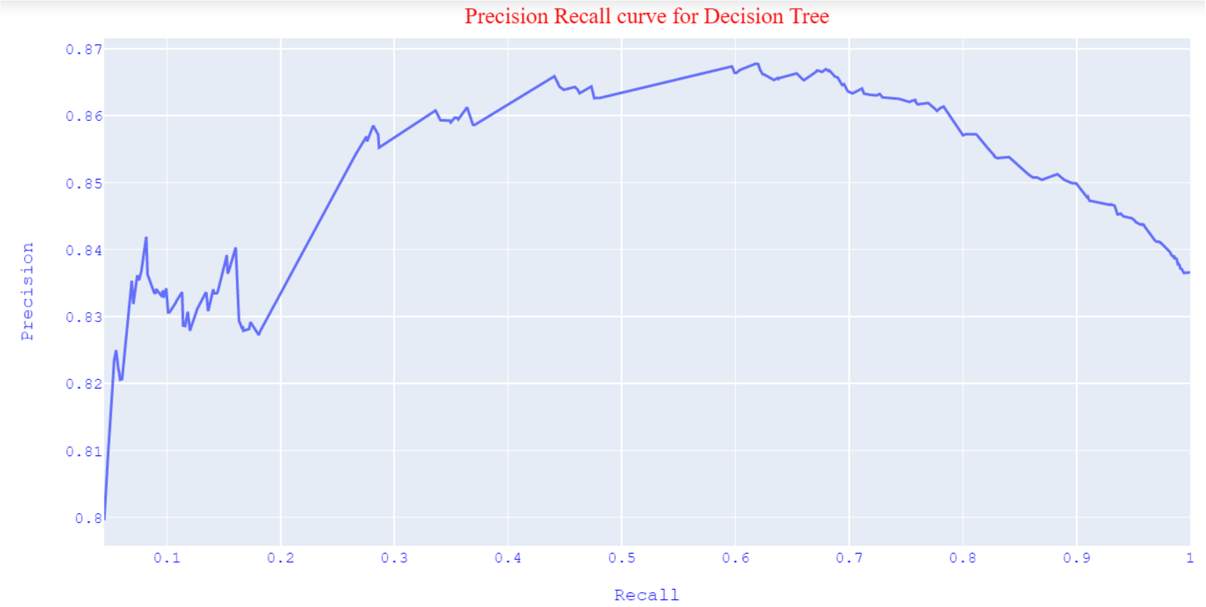


Figure 36: Precision-Recall curve for Decision Tree

Figure 36 clearly shows that for a given recall at 0, precision dropped sharply and had a significant jump when recall reached 0.4. The plot looks better when recall lies around 0.4 to 0.6 and decreases slowly to reach precision around 0.84 when the recall was 1. However, this PR curve is better than that of the KNN PR curve. With a 0.6 threshold, 0.86 precision and 0.61 recall can be obtained.

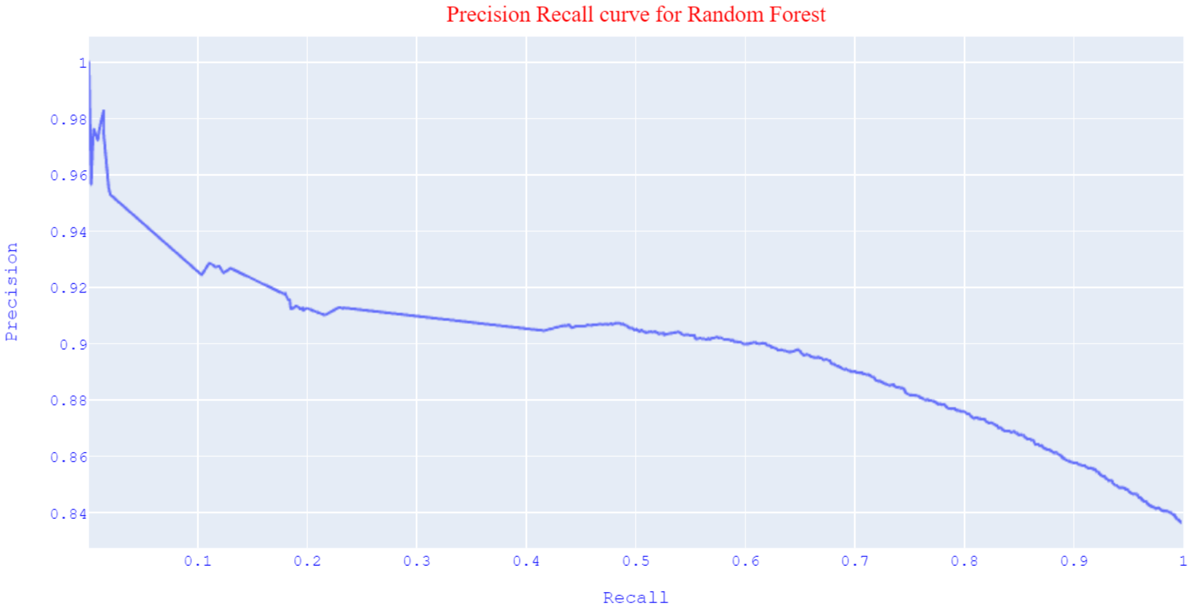


Figure 37: Precision-Recall curve for Random Forest

Figure 37 display the PR curve for Random forest with an average precision score of 0.87. With a 0.8 threshold value, 100% precision is obtained but recall is less than 0.1. It means in that threshold the predicted values were 100% correctly predicted as a show and the values were the actual show. However, the recall obtained was terrible. With a threshold of 0.6, precision was 90% correctly predicted as show and was shown whereas recall was 54% which means out of all show, 54% was predicted show.

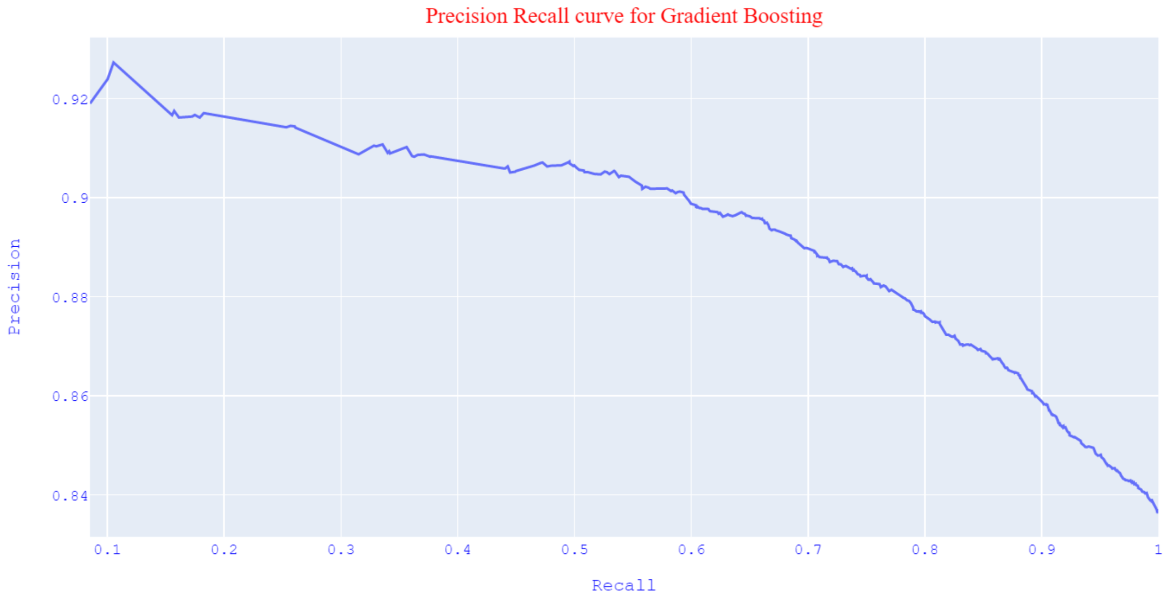


Figure 38: Precision-Recall curve for Gradient Boosting

In figure 38, with the threshold of 0.7, the predicted values were 92% correctly predicted as a show and the values were actual show, whereas 9% of recall means out of all show, 9% was predicted show. With a threshold of 0.42, the predicted values were 87% correctly predicted as a show and the values were the actual show. However, the recall was 79%, implies, out of all show, 79% was predicted show. With a threshold of 0.22, precision was 83%, 83% correctly predicted as a show and the values were actual show, the recall was 100%, out of all show 100% was predicted show.

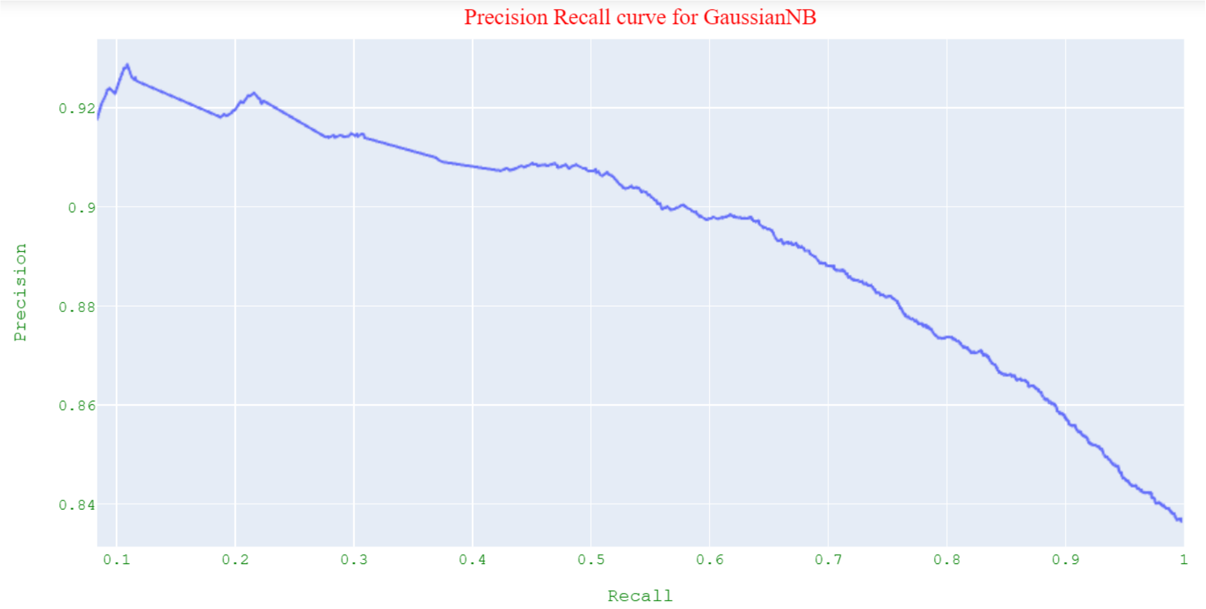


Figure 39: Precision-Recall curve for GaussianNB

In figure 39, with the threshold of 0.81, the predicted values were 92% correctly predicted as a show and the values were actual show, whereas 9.17% of recall means out of all show, 9.17% was predicted show. With a threshold of 0.50, the predicted values were 87% correctly predicted as a show and the values were the actual show. However, the recall was 80%, implies, out of all show, 80% was predicted show. With a threshold of 0.2, precision was 85%, 85% correctly predicted as a show and the values were actual show, the recall was 90%, out of all show 90% was predicted show.

5 Conclusion

This chapter consists of two sections. The first section, 5.1 provides a brief overview of contributions in this master thesis report. Section 5.2 discusses the limitations and future work and my further investigation which will be considered in the article related to this thesis.

5.1 Achievements

In this study, I successfully explored and classified the hospital missed appointment dataset to predict the missed appointments. I have selected and used 6 Binary classification systems: Logistic Regression, K Nearest Neighbour, Decision Tree, Random Forest, Gradient Boosting, and GaussianNB.

About 9.44% missed the appointments and 10.57% cancelled the appointments and young adults miss most of the appointments. Monday was the top-performing day for scheduling appointments. Thursday is the most missed appointment day and November is the most missed appointment month.

The dataset was originally in the Norwegian language with column names, but the variable description was later translated into English for a better understanding of the problem. First, the data wrangling tasks were performed. In other words, exploratory data analysis was the major task for a better understanding of data. Since data was imbalanced, so SMOTE was used for balancing the dataset. This is an oversampling technique widely used for imbalanced datasets. Furthermore, data were scaled using the minmaxscaler technique. This technique is useful with Logistic regression and Decision Trees because if scales are non-uniform it shows misleading results.

Then, the dataset is split into 70% of training data and 30% of testing data.

Gaussian Naïve Bayes was extremely fast while executing the training dataset as compared to the other 5 binary classifiers and it does not require a large training dataset to obtain a good estimate of probability. Also, the calculation for the decision tree was minimal and therefore easy to understand and implement.

Gaussian naïve Bayes have minimum misclassification rate of 74.35%. This metric is judicious if the costs associated with each error are the same. However, it is not favourable to select the best model just with the lowest misclassification rate although the cost associated with it is the same. Having said that, it provides rich knowledge of the importance of the confusion matrix and the total cost or loss associated with the classification rate scenario.

Random Forest is best in terms of AUC score followed by Logistic Regression and Gradient Boosting equally. This classifier(RF) is better among all in terms of distinguishing between shows and no-shows. However, it ignores predicted probability values and goodness-of-fit of the model and most importantly it ignores well predicted missed appointments or no-shows.

Precision was highest for Random Forest with 89.56% correctly predicted as a show and the values were actual show, whereas recall was highest for GaussianNB which was 80.87% which means out of all show, 80.87% was predicted show.

PR curve is preferred over when no-show is more interesting and needed than the no-shows. Given the probability of the problem how meaningful is the show result, then PR precedes over the ROC curve. Gradient Boosting and Random Forest have an equal score of 0.88.

Another determining evaluation metric is Matthews's Correlation coefficient. Logistic regression achieved a 0.208 score and Random Forest achieved a 0.2 score. The score will be higher if all the negative and positive classes of a confusion matrix perform better in terms of the proportion it holds in the dataset. Therefore, it is a more reliable and trustworthy single metric while evaluating the models.

To sum up, it would be well served to develop an understanding of the situation and business objectives under which each evaluation metric should be utilized.

5.2 Limitations and Future work

Like with other studies with the problem of limitations, there are several limitations while doing this thesis. Several important features were missing and if available some are not allowed to use. For instance, Gender, SMS reminders. Some other pivotal features were excluded such as disease information, mental problem status, depression, scholarship, Diabetes, Handicap, missing appointments in specific medical departments, the introduction of a fee for missed appointments, and married or not. Other features such as Weather information, date around March 2020, or past data to compare the patterns of missed appointments were missing. These features might have a significant impact on missing scheduled appointments. Appointment scheduled day, waiting times were not in the dataset. Moreover, patient address, sociocultural background plays a crucial role in predicting show or no-show. All these features when explored properly give huge data knowledge and helps to understand the problem. Furthermore, the reason behind the missed appointments was not recorded in the dataset. This also plays a decisive role or at least has predictive power to study the characteristics of patients who will miss appointments in the future.

Several evaluation metrics could have been added for evaluating the binary classifier such as Cohen’s kappa score, balanced accuracy, informedness, markedness. Although, several evaluation metrics have been included to evaluate the models. It might be fruitful to engage patients in research that will have a meaningful impact on my research. Furthermore, the availability of more data has a tremendous impact on training, testing, and selecting the best model. More data added to training data and test data creates a larger dataset and could contribute to training more robust models.

Several robust models for classifications such as XGBoost, Support Vector machines could have added better comparisons with other models. Unsupervised machine learning models such as hidden Markov models, natural language processing, principal component analysis, etc might contribute rich and better analysis for this master thesis.

6 References

Ahmadi E, Garcia-Arce A, Masel DT, Reich E, Puckey J, Maff R. A metaheuristic-based stacking model for predicting the risk of patient no-show and late cancellation for neurology appointments. IISE Transact Healthcare Syst Engineering. 2019;9(3):272–91

Andrews R, Morgan JD, Addy DP, McNeish AS. Understanding non-attendance in outpatient paediatric clinics. Arch Dis Child 1990; 65(2):192–5

An Implementation and Explanation of the Random Forest in Python. (2021). Retrieved 6 June 2021, from https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76

API reference — pandas 1.2.4 documentation. (2021). Retrieved 8 June 2021, from https://pandas.pydata.org/pandas-docs/stable/reference/index.html

Bech, M. (2005). The economics of non-attendance and the expected effect of charging a fine on non-attendees. Health Policy, 74(2), 181-191. doi:10.1016/j.healthpol.2005.01.001.

Barron WM. Failed appointments. Who misses them, why they are missed, and what can be done. Prim Care. 1980 Dec;7(4):563-74. PMID: 7010402.

Bhattacharya, J., Hyde, T., & Tu, P. (2014). Health Economics. London: Palgrave Macmillan

Bunker, R., & Thabtah, F. (2019). A machine learning framework for sport result prediction. *Applied Computing And Informatics*, *15*(1), 27-33. doi: 10.1016/j.aci.2017.09.005

Chicco, D., Tötsch, N., & Jurman, G. (2021). The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *Biodata Mining*, *14*(1). doi: 10.1186/s13040-021-00244-z

Classification and regression - Spark 3.1.2 Documentation. (2021). Retrieved 6 June 2021, from https://spark.apache.org/docs/latest/ml-classification-regression.html#decision-trees

Classification: ROC Curve and AUC | Machine Learning Crash Course. (2021). Retrieved 11 April 2021, from https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=no

Daggy J, Lawley M, Willis D, Thayer D, Suelzer C, DeLaurentis PC, et al. Using no-show modeling to improve clinic performance. Health Informatics Journal 2010;16:246–59

Dantas, L.F., Fleck, J.L., Cyrino Oliveira, F.L., & Hamacher, S. (2018). No-shows in appointment scheduling – a systematic literature review. Health Policy, 122(4), 412-421. https://doi.org/10.1016/j.healthpol.2018.02.002.

Data cleaning: The benefits and steps to creating and using clean data. (2021). Retrieved 1 June 2021, from https://www.tableau.com/learn/articles/what-is-data-cleaning

DiMatteo, M.R., Lepper, H.S., & Croghan, T.W. (2000). Depression Is a Risk Factor for Noncompliance With Medical Treatment: Meta-analysis of the Effects of Anxiety and Depression on Patient Adherence. Arch Intern Med, 160(14), 2101-2107. doi:10.1001/archinte.160.14.2101

Elvira C, Ochoa A, Gonzalvez JC, Mochón F. Machine-learning-based no show prediction in outpatient visits. International Journal of Interactive Multimedia & Artificial Intelligence. 2018 Mar 1;4(7).

Granado, E., & Garcia, E. (2021). GUIDE TO JUPYTER NOTEBOOKS FOR EDUCATIONAL PURPOSES. Retrieved 8 June 2021, from https://eprints.ucm.es/id/eprint/48305/1/ManualJupyterIngles.pdf

Guy, R., Hocking, J., Wand, H., Stott, S., Ali, H., & Kaldor, J. (2012). How effective are short message service reminders at increasing clinic attendance? A meta-analysis and systematic review. Health Service Research, 47(2), 614-632. Doi:10.1111/j.1475- 6773.2011.01342.x

Hanauer, D., & Huang, Y. (2014). Patient No-Show Predictive Model Development using Multiple Data Sources for an Effective Overbooking Approach. *Applied Clinical Informatics*, *05*(03), 836-860. doi: 10.4338/aci-2014-04-ra-0026

Hasvold, P.E., & Wootton, R. (2011). Use of telephone and SMS reminders to improve attendance at hospital appointments: a systematic review. Journal of Telemedicine and Telecare, 17(7), 358-364. doi:10.1258/jtt.2011.110707

Healthwatch Lincolnshire. (2014). Report on the Impact of Patient ´Did Not Attend´ Appointments at GP Surgeries in Lincolnshire. Retrieved from https://www.healthwatchlincolnshire.co.uk/sites/healthwatchlincolnshire.co.uk/files/Report-on-the-Impact-of-Patient-DNA-Final%20%283%29.pdf

imbalanced-learn API — imbalanced-learn 0.3.0.dev0 documentation. (2021). Retrieved 2 June 2021, from http://glemaitre.github.io/imbalanced-learn/api.html

Karter AJ, Parker MM, Moffet HH, Ahmed AT, Ferrara A, Liu JY, et al. Missed appointments and poor glycemic control: an opportunity to identify high-risk diabetic patients. Medical Care 2004;42:110–5

Khamsan, M., & Maskat, R. (2019). HANDLING HIGHLY IMBALANCED OUTPUT CLASS LABEL. *MALAYSIAN JOURNAL OF COMPUTING*, *4*(2), 304. doi: 10.24191/mjoc.v4i2.7021

Kheirkhah P, Feng Q, Travis LM, Tavakoli-Tabasi S, Sharafkhaneh A. Prevalence, predictors and economic consequences of no-show. BMC Health Services Research 2016;16:1–6

K-Nearest Neighbors (KNN) Algorithm for Machine Learning. (2021). Retrieved 2 June 2021, from https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26

Kurasawa H, Hayashi K, Fujino A, Takasugi K, Haga T, Waki K, Noguchi T, Ohe K. Machine-learning-based prediction of a missed scheduled clinical appointment by patients with diabetes. J Diab Sci Technol. 2016;10(3):730–6.

Logistic Regression — ML Glossary documentation. (2021). Retrieved 6 June 2021, from https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html

Moghadassi, F. Parvizian, and S. Hosseini, “A New Approach Based on Artificial Neural Networks for Prediction of High Pressure Vapor-liquid Equilibrium”, Australian Journal of Basic and Applied Sciences, Vol. 3, No. 3, pp. 1851-1862, 2009

Mohamed, H., Negm, A., Zahran, M., & Saavedra, O. C. (2015). Assessment of artificial neural network for bathymetry estimation using High Resolution Satellite imagery in Shallow Lakes: case study El Burullus Lake. In International water technology conference

Mohammadi I, Wu H, Turkcan A, Toscos T, Doebbeling BN. Data analytics and modeling for appointment no-show in community health centers. J Primary Care Community Health. 2018;9:2150132718811692

Menendez ME, Ring D. Factors associated with non-attendance at a hand surgery appointment. Hand 2015;10:221–6

Naives Bayes Classifiers for Machine Learning. (2021). Retrieved 6 June 2021, from https://medium.com/capital-one-tech/naives-bayes-classifiers-for-machine-learning-2e548bfbd4a1

NHS Digital. (2018). *Hospital Outpatient Activity, 2017-18* (p. 4). England: NHS. Retrieved from https://digital.nhs.uk/data-and-information/publications/statistical/hospital-outpatient-activity/2017-18

Peng Y, Erdem E, Shi J, Masek C, Woodbridge P. Large-scale assessment of missed opportunity risks in a complex hospital setting. Informatics for Health and Social Care 2016;41:112–27

Pesata, V., Pallija, G., & Webb, A.A. (1999). A descriptive study of missed appointments: Families´ Perception of barriers to care. Journal of Pediatric Health Care, 13(4), 178-182. https://doi.org/10.1016/S0891-5245(99)90037-8.

Precision-Recall — scikit-learn 0.24.2 documentation. (2021). Retrieved 20 May 2021, from https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html#sphx-glr-auto-examples-model-selection-plot-precision-recall-py

ROC and PR Curves. (2021). Retrieved 14 June 2021, from https://plotly.com/python/roc-and-pr-curves/

Saito, T., & Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, *10*(3), e0118432. doi: 10.1371/journal.pone.0118432

Stubbs, N.D., Geraci, S.A., Stephenson, P.L., Jones, D.B., & Sanders, S. (2012). Methods to reduce outpatient non-attendance. The American Journal of the Medical Science, 344(3), 211-219. Doi:10.1097/MAJ.0b013e31824997c6.

Torres O, Rothberg MB, Garb J, Ogunneye O, Onyema J, Higgins T. Risk factor model to predict a missed clinic appointment in an urban, academic, and underserved setting. Population Health Management 2015;18:131–6

Triemstra, J., & Lowery, L. (2018). Prevalence, Predictors, and the Financial Impact of Missed Appointments in an Academic Adolescent Clinic. *Cureus*. doi: 10.7759/cureus.3613

Turkcan, A., Nuti, L., DeLaurentis, P., Tian, Z., Daggy, J., Zhang, L., Lawley, M., & Sands, L. (2013). No-show Modeling for Adult Ambulatory Clinics. In B.T Denton (Eds.). Handbook of Healthcare Operations Management: Methods and Applications (pp. 251-288). New York: Springer. Doi:10.1007/978-1-4614-5885-2.

U. Khan, T. K. Bandopadhyaya, and S. Sharma, "Classification of Stocks Using Self Organizing Map", International Journal of Soft Computing Applications, Issue 4, 2009, pp.19-24

Understanding Confusion Matrix. (2021). Retrieved 13 May 2021, from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

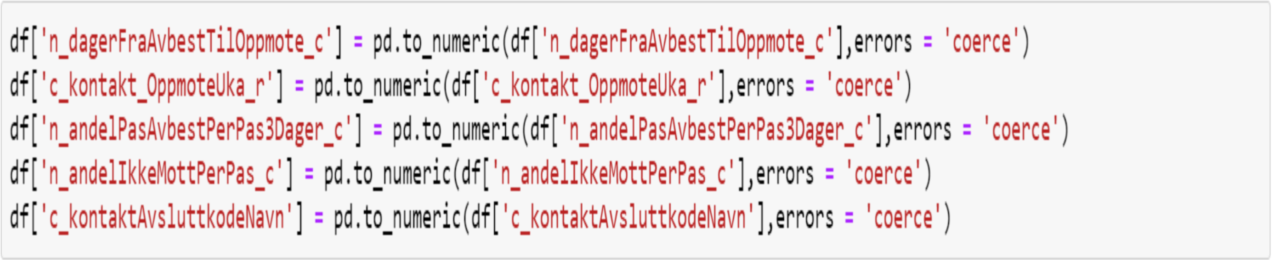
Understanding Gradient Boosting Machines. (2021). Retrieved 6 June 2021, from https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

Valverde-Albacete, F., & Peláez-Moreno, C. (2014). 100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox. *Plos ONE*, *9*(1), e84217. doi: 10.1371/journal.pone.0084217

Wirth, R., & Hipp, J. CRISP-DM: Towards a Standard Process Model for Data Mining. Retrieved from http://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf

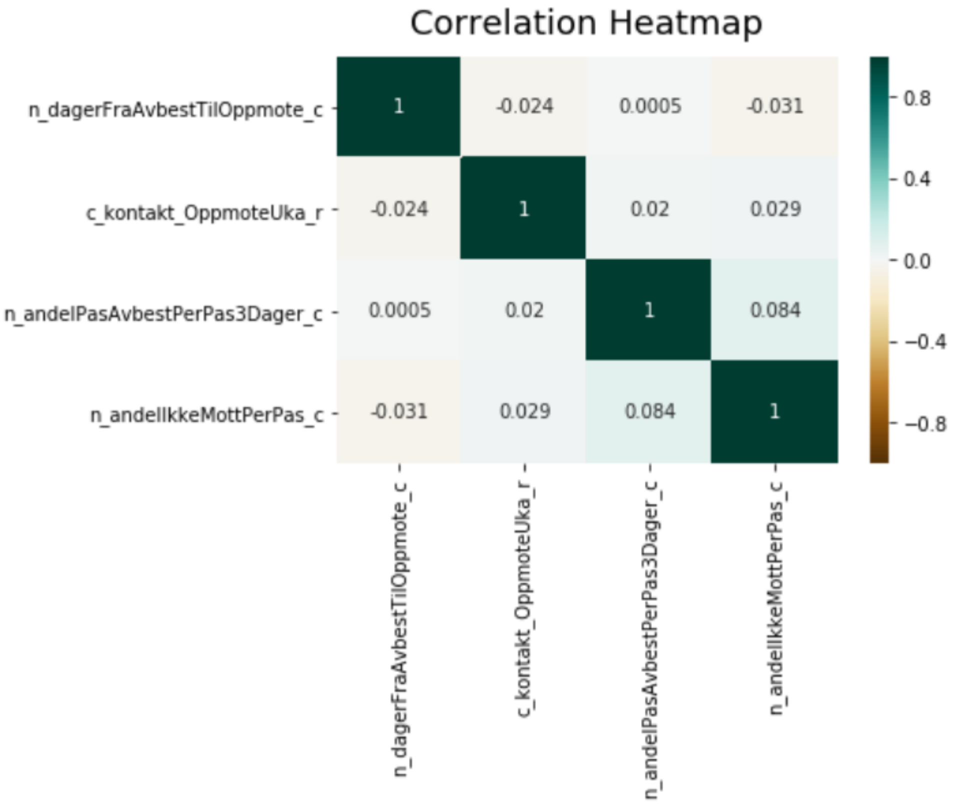
Witten, I. H., Frank, E., and Hall, M. A. (2011). Data mining: Practical machine learning tools and techniques. Burlington, MA: Morgan Kaufmann.

7 Appendices

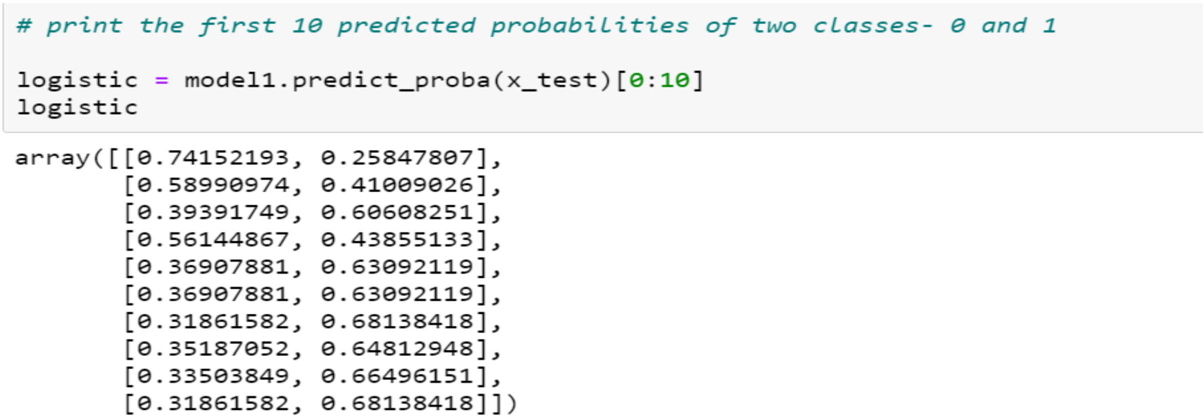


```
df['n_dagerFraAvbestTilOppmote_c'] = pd.to_numeric(df['n_dagerFraAvbestTilOppmote_c'],errors = 'coerce')
df['c_kontakt_OppmoteUka_r'] = pd.to_numeric(df['c_kontakt_OppmoteUka_r'],errors = 'coerce')
df['n_andelPasAvbestPerPas3Dager_c'] = pd.to_numeric(df['n_andelPasAvbestPerPas3Dager_c'],errors = 'coerce')
df['n_andelIkkeMottPerPas_c'] = pd.to_numeric(df['n_andelIkkeMottPerPas_c'],errors = 'coerce')
df['c_kontaktAvsluttkodeNavn'] = pd.to_numeric(df['c_kontaktAvsluttkodeNavn'],errors = 'coerce')
```

A: Converting Object data type to a numeric datatype

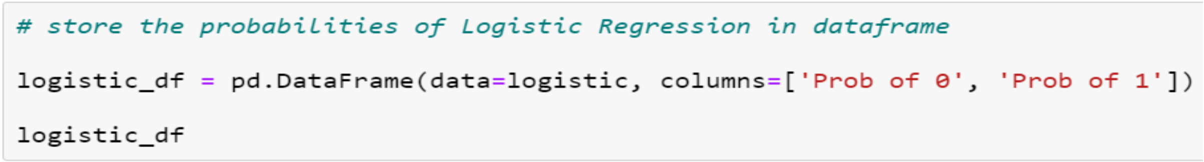


B: Correlation Heatmap between variables:



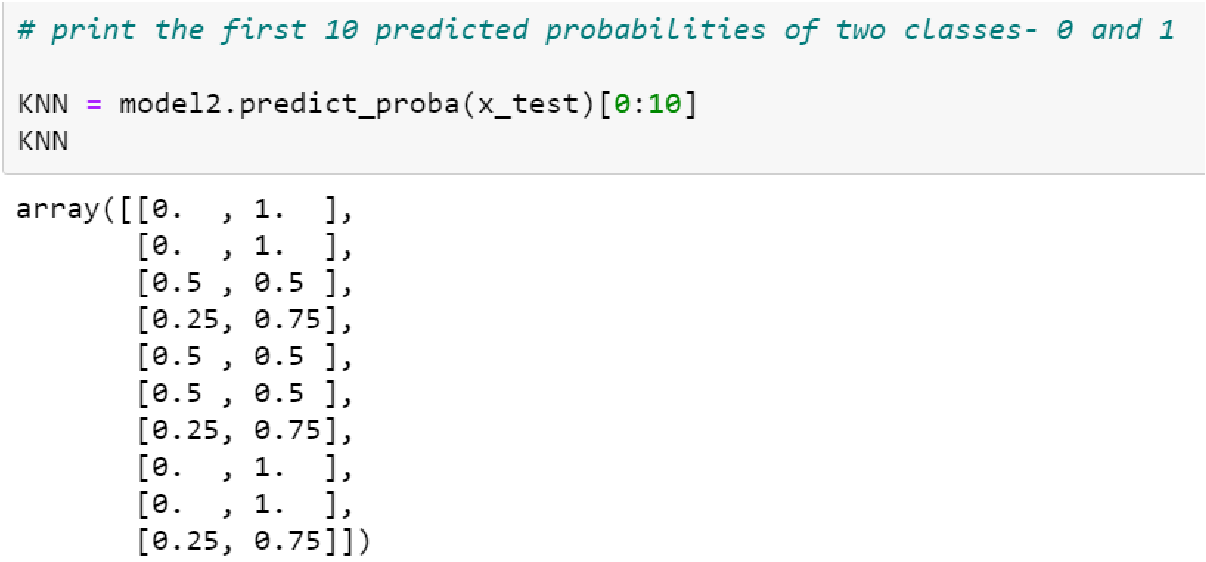
```
# print the first 10 predicted probabilities of two classes- 0 and 1

logistic = model1.predict_proba(x_test)[0:10]
logistic
```

```
array([[0.74152193, 0.25847807],
       [0.58990974, 0.41009026],
       [0.39391749, 0.60608251],
       [0.56144867, 0.43855133],
       [0.36907881, 0.63092119],
       [0.36907881, 0.63092119],
       [0.31861582, 0.68138418],
       [0.35187052, 0.64812948],
       [0.33503849, 0.66496151],
       [0.31861582, 0.68138418]])
```



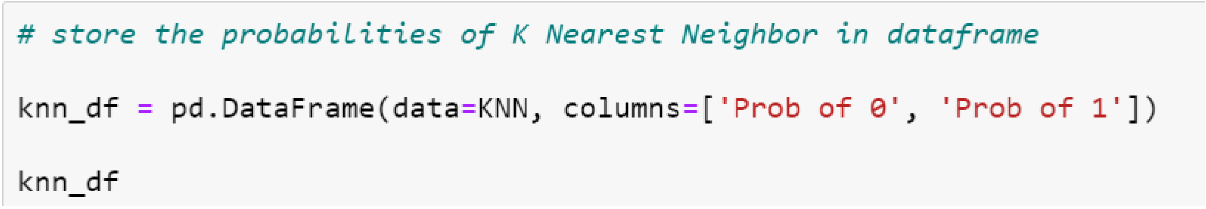
```
# store the probabilities of Logistic Regression in dataframe

logistic_df = pd.DataFrame(data=logistic, columns=['Prob of 0', 'Prob of 1'])

logistic_df
```

C: Predicted Probabilities of Logistic Regression



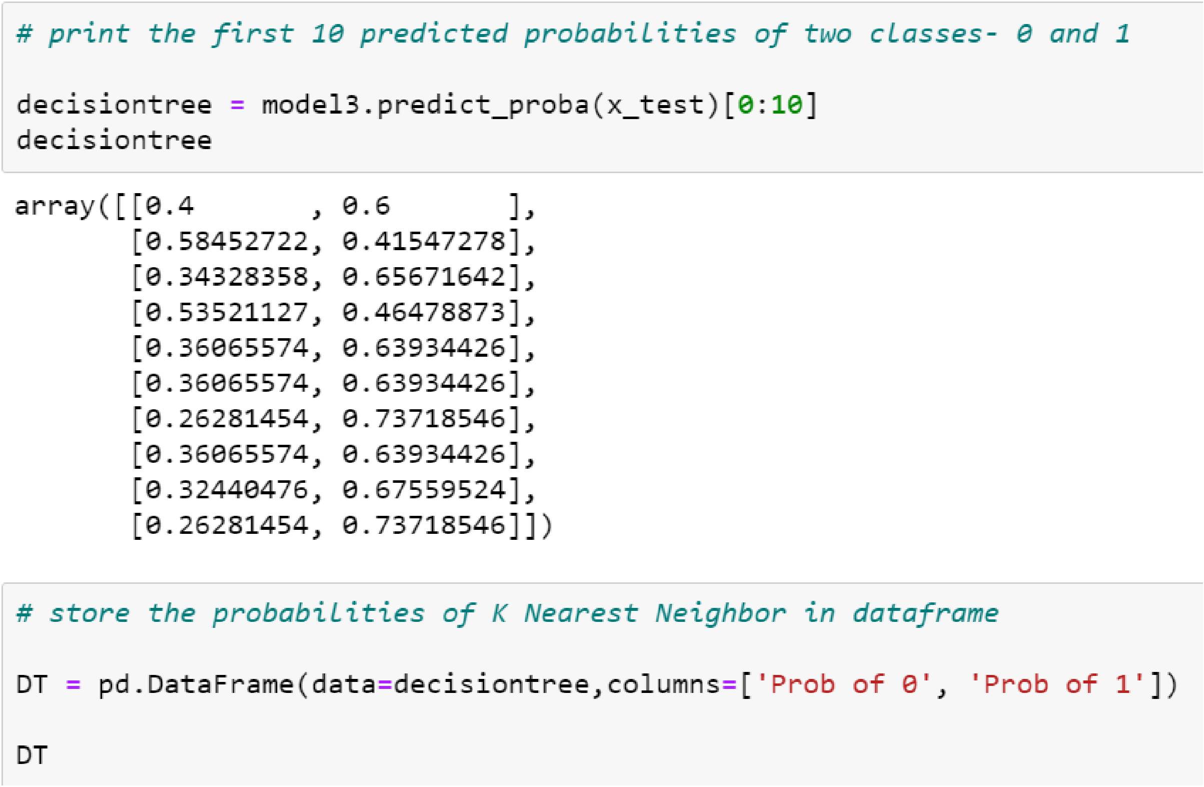
```
# print the first 10 predicted probabilities of two classes- 0 and 1

KNN = model2.predict_proba(x_test)[0:10]
KNN
```

```
array([[0.  , 1.  ],
       [0.  , 1.  ],
       [0.5 , 0.5 ],
       [0.25, 0.75],
       [0.5 , 0.5 ],
       [0.5 , 0.5 ],
       [0.25, 0.75],
       [0.  , 1.  ],
       [0.  , 1.  ],
       [0.25, 0.75]])
```



```
# store the probabilities of K Nearest Neighbor in dataframe

knn_df = pd.DataFrame(data=KNN, columns=['Prob of 0', 'Prob of 1'])

knn_df
```

D: Predicted Probabilities of KNN

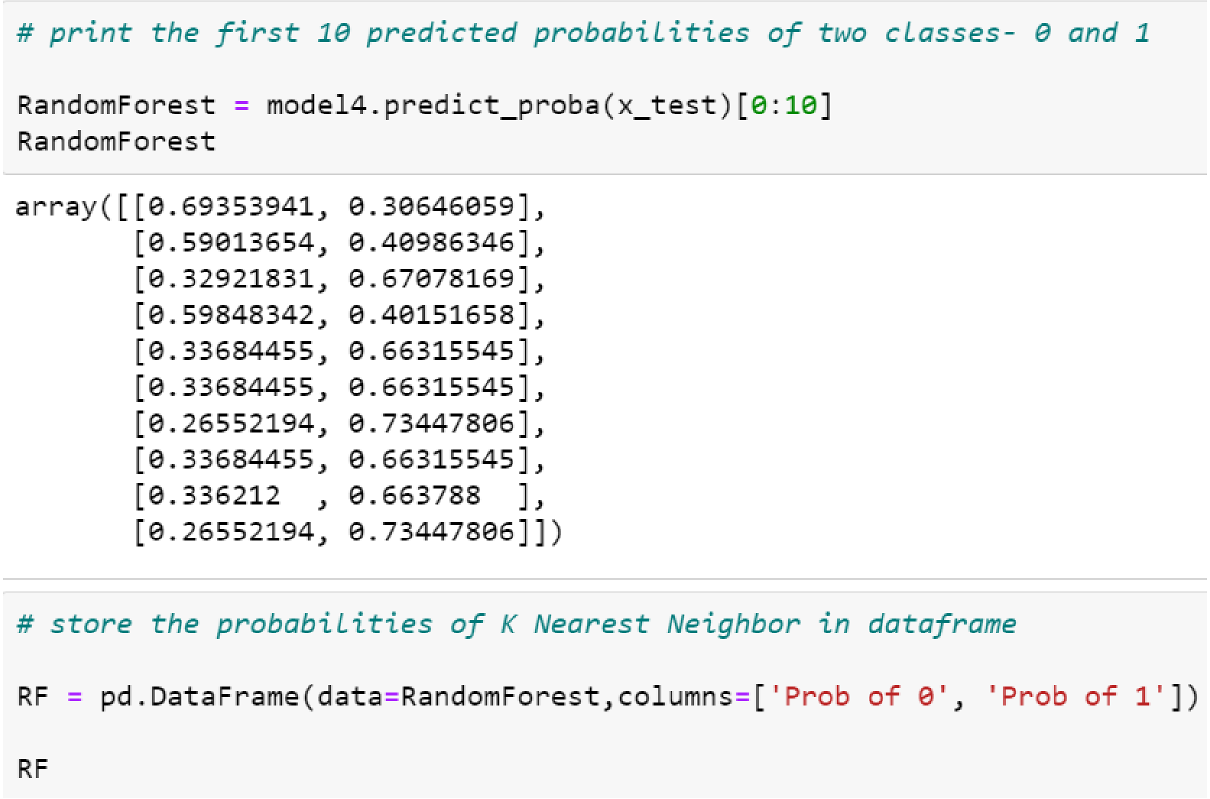


```
# print the first 10 predicted probabilities of two classes- 0 and 1

decisiontree = model3.predict_proba(x_test)[0:10]
decisiontree
```

```
array([[0.4       , 0.6       ],
       [0.58452722, 0.41547278],
       [0.34328358, 0.65671642],
       [0.53521127, 0.46478873],
       [0.36065574, 0.63934426],
       [0.36065574, 0.63934426],
       [0.26281454, 0.73718546],
       [0.36065574, 0.63934426],
       [0.32440476, 0.67559524],
       [0.26281454, 0.73718546]])
```

```
# store the probabilities of K Nearest Neighbor in dataframe

DT = pd.DataFrame(data=decisiontree,columns=['Prob of 0', 'Prob of 1'])

DT
```

E: Predicted Probabilities of DT

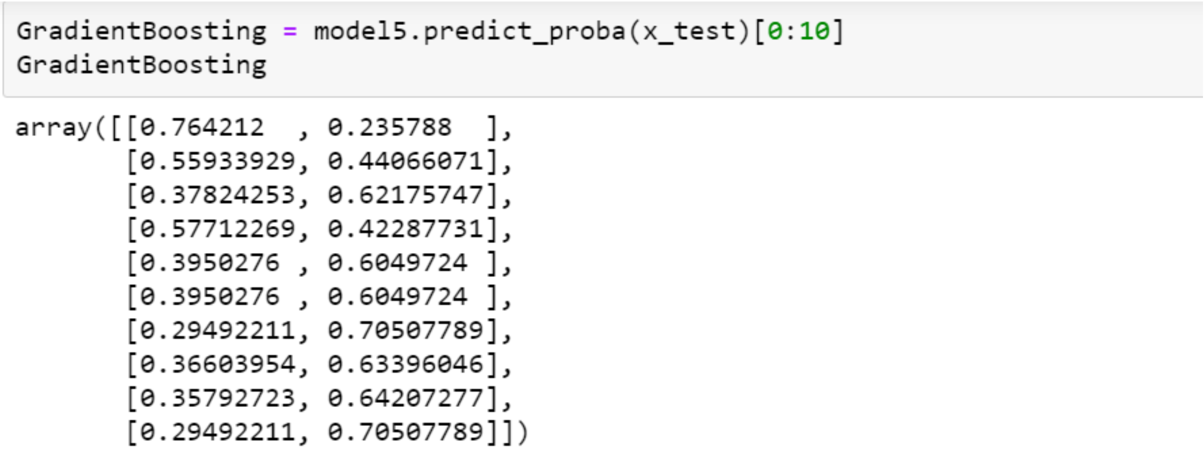


```
# print the first 10 predicted probabilities of two classes- 0 and 1

RandomForest = model4.predict_proba(x_test)[0:10]
RandomForest
```

```
array([[0.69353941, 0.30646059],
       [0.59013654, 0.40986346],
       [0.32921831, 0.67078169],
       [0.59848342, 0.40151658],
       [0.33684455, 0.66315545],
       [0.33684455, 0.66315545],
       [0.26552194, 0.73447806],
       [0.33684455, 0.66315545],
       [0.336212  , 0.663788  ],
       [0.26552194, 0.73447806]])
```

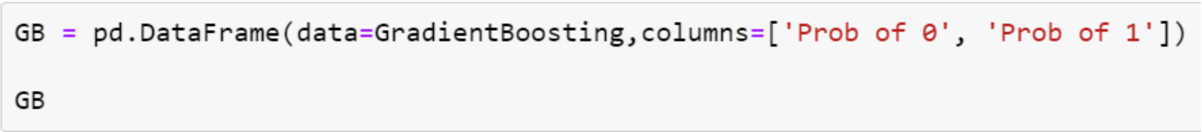
```
# store the probabilities of K Nearest Neighbor in dataframe

RF = pd.DataFrame(data=RandomForest,columns=['Prob of 0', 'Prob of 1'])

RF
```

F: Predicted Probabilities of RF



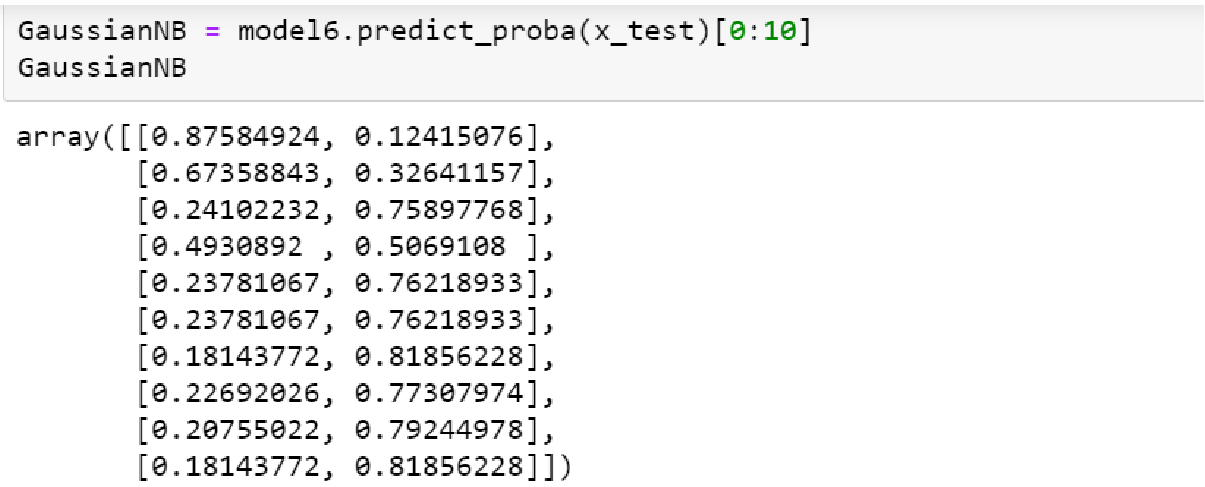
```
GradientBoosting = model5.predict_proba(x_test)[0:10]
GradientBoosting
```

```
array([[0.764212  , 0.235788  ],
       [0.55933929, 0.44066071],
       [0.37824253, 0.62175747],
       [0.57712269, 0.42287731],
       [0.3950276 , 0.6049724 ],
       [0.3950276 , 0.6049724 ],
       [0.29492211, 0.70507789],
       [0.36603954, 0.63396046],
       [0.35792723, 0.64207277],
       [0.29492211, 0.70507789]])
```



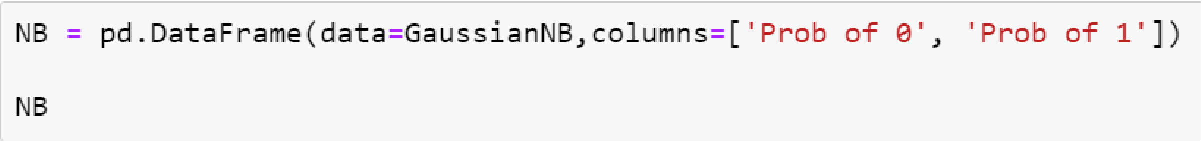
```
GB = pd.DataFrame(data=GradientBoosting,columns=['Prob of 0', 'Prob of 1'])

GB
```

G: Predicted Probabilities of Gradient Boosting



```
GaussianNB = model6.predict_proba(x_test)[0:10]
GaussianNB
```

```
array([[0.87584924, 0.12415076],
       [0.67358843, 0.32641157],
       [0.24102232, 0.75897768],
       [0.4930892 , 0.5069108 ],
       [0.23781067, 0.76218933],
       [0.23781067, 0.76218933],
       [0.18143772, 0.81856228],
       [0.22692026, 0.77307974],
       [0.20755022, 0.79244978],
       [0.18143772, 0.81856228]])
```



```
NB = pd.DataFrame(data=GaussianNB,columns=['Prob of 0', 'Prob of 1'])

NB
```

H: Predicted Probabilities of GaussianNB

All the predicted probabilities of show and no-show are crucial to know which patient will miss the appointment and which type of patient attends the scheduled appointments.