|  |  |  |
| --- | --- | --- |
| Image | |  |
| REPORT **No Show Prediction Model** | | |
|  |  | |

|  |
| --- |
|  |
| Image |
| ABSTRACT The objective of this project is to predict the that a customer will show up for the appointment or not. The project work has been done with python programming language; the data used for model training is Q1 data of year 2020. The accuracy of model is 0.99.  Multiple machine learning models are used in the project, Model with best performance metrics is chosen with focus on interpretability. |

|  |  |  |
| --- | --- | --- |
|  | |  |
|  | | |
| Table of Content  * Project Background (Introduction) * Project Objective * Exploratory Data Analysis      * Research Approach * Inferences from the ML Models * Recommendation and Conclusion |  | |

#### List of Tables and Figures

The training data contain 170258 entries of patient details and the testing data contains 47465 entries. Both the dataset has 36 columns in which the Target Column/Prediction Variable is also included.

#### Variable Type & Description

|  |  |  |
| --- | --- | --- |
| Serial Number | Name of Column | Data Type |
| 1 | FACILITY | object |
| 2 | LOCATION\_NAME | object |
| 3 | MRN | object |
| 4 | FIN | object |
| 5 | GENDER | object |
| 6 | AGE | int64 |
| 7 | NATIONALITY\_IDENTIFIER | object |
| 8 | FIN\_CLASS | object |
| 9 | APPT\_TYPE | object |
| 10 | ENCT\_TYPE | object |
| 11 | TYPE\_NAME | object |
| 12 | APPT\_CLASS | object |
| 13 | BOOKING\_DT\_TM | datetime64[ns] |
| 14 | APPT\_DATE\_TIME | datetime64[ns] |
| 15 | REG\_DT\_TM | datetime64[ns] |
| 16 | APPT\_STATUS | object |
| 17 | BOOKING\_PRSNL | object |
| 18 | RESOURCE\_NAME | object |
| 19 | SEEN\_BY\_PHYSICIAN | object |
| 20 | SEEN\_BY\_GD | object |
| 21 | SEEN\_BY\_PHYSICIAN\_ID | object |
| 22 | SPECIALTY | object |
| 23 | SUB\_SPECIALTY | object |
| 24 | Gender Group | object |
| 25 | Age Group | object |
| 26 | Nat Group | object |
| 27 | Fin Group | object |
| 28 | Appt Class Group | object |
| 29 | Appt Hour | int64 |
| 30 | Appt Day | object |
| 31 | Appt Month | object |
| 32 | Appt Wait (Days) | float64 |
| 33 | Appt Wait (Range) | object |
| 34 | User Group | object |
| 35 | No Show | object |
| 36 | Category | object |

#### Summary Statistics

At glance we have 3 numerical columns available in the data, for which the statistics summary is given below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | AGE | Appt Hour | Appt Wait (Days) |
| count | 170258 | 170258 | 170258 |
| mean | 35.95915023 | 11.54580108 | 21.19704588 |
| std | 22.29471106 | 2.809250609 | 37.57067152 |
| min | 0 | 0 | -105.0125 |
| 25% | 17 | 9 | 0.838888889 |
| 50% | 35 | 11 | 8.059722222 |
| 75% | 52 | 13 | 23.73611111 |
| max | 120 | 23 | 1284.98125 |

#### Project Objective.

The objective of this document is to summarize and propose a model to predict the probability of patient No Show occurrence by way of:

* Understanding of the factors driving No Show;
* Profiling of current No Show population;
* Application of Operational Research Statistics;
* Evaluation of Prediction Success Rate; and
* Prediction Model Roll Out & Reduction Tactics.

#### Data Source.

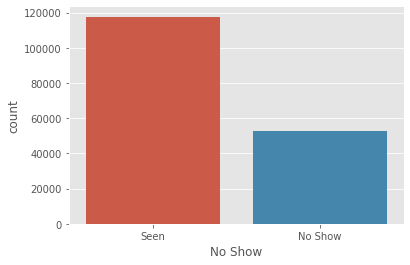
The testing data contain 170258 entries of patient details and the testing data contains 47465 entries. Both the dataset has 36 columns in which the Target Column/Prediction Variable is also included.

#### Statistical Tools and Techniques.

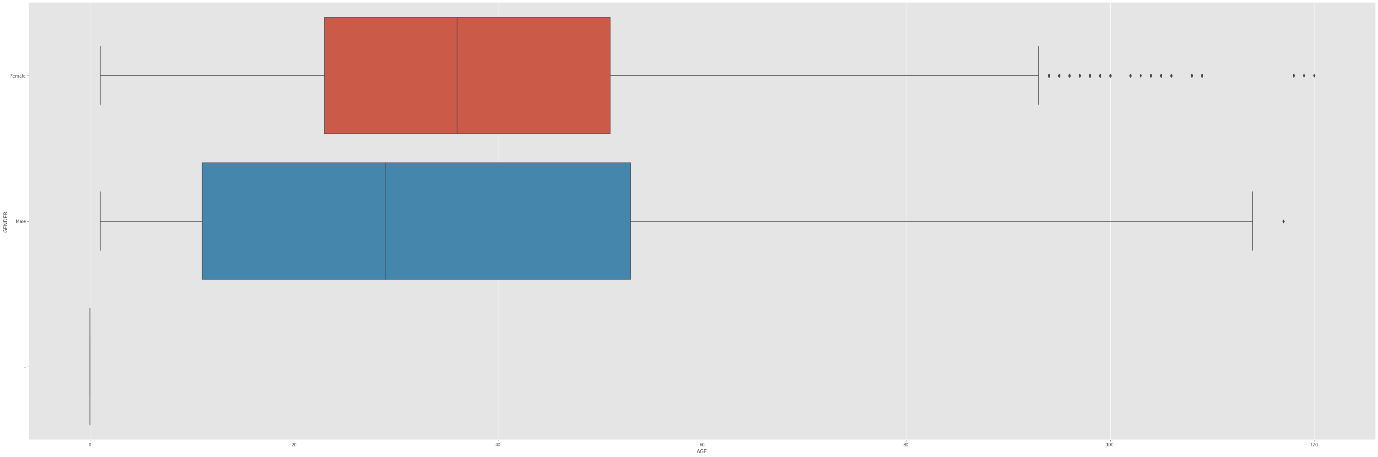
|  |  |
| --- | --- |
| Techniques | Machine Learning using DecisionTreeClassifier, KNeighborsClassifier, AdaBoostClassifier, GradientBoostingClassifier, SVC, LogisticRegression |
| Tools | Python, MS Excel |

#### Exploratory Data Analysis.

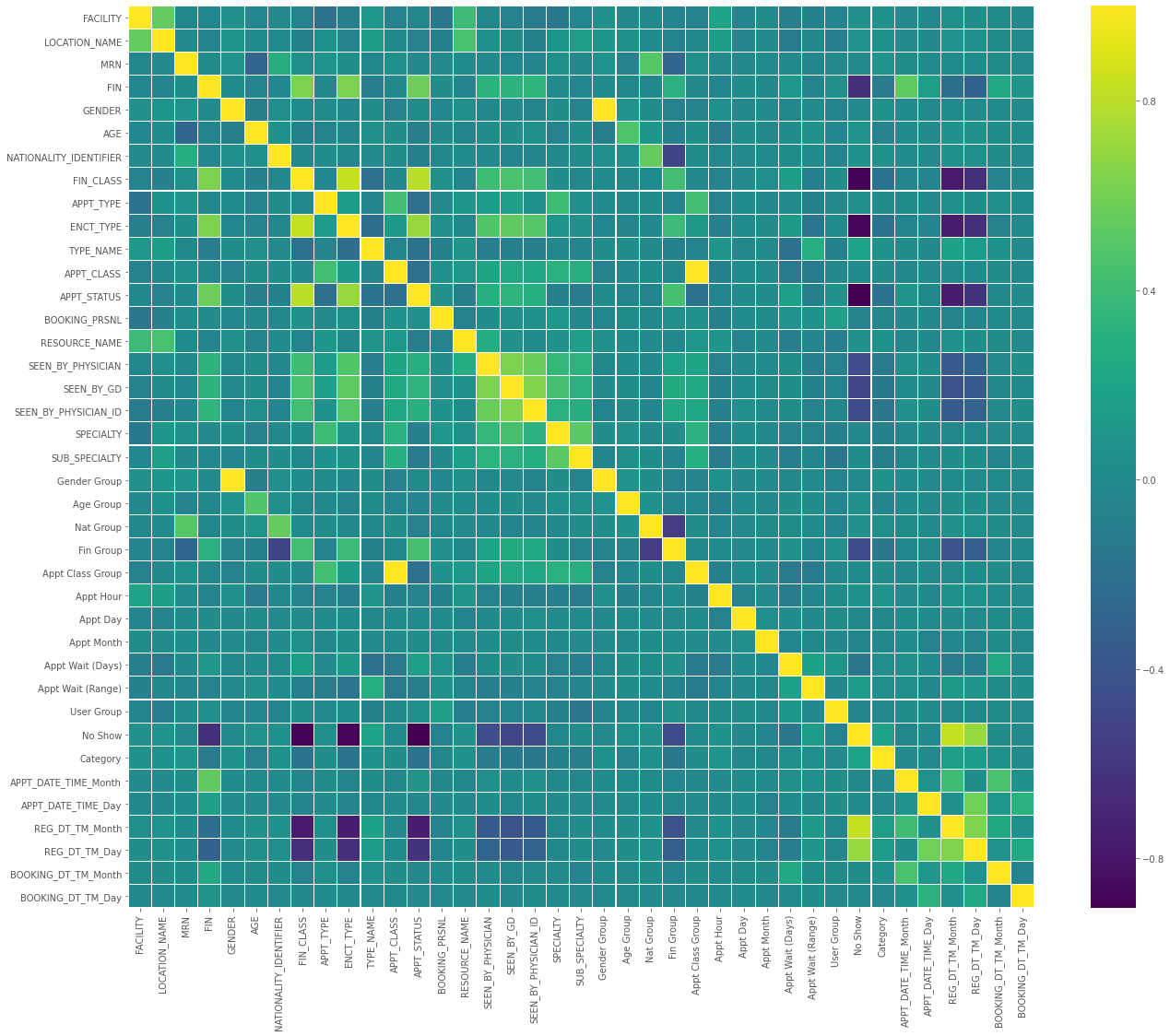
1. Total number of Show and Not Show available in the dataset.



1. Box Plot of Age with respect to Gender.



1. Correlation Plot of different data (After the data has been converted from categorical to numerical data using Encoding techniques).



1. Correlation Matrix.



#### Data preparation

1. Loading the data, the data is in 4 excel files, First the files are loaded separately into four separate data frames and concatenating them in testing and training file.
2. Convert data type of "BOOKING\_DT\_TM", "APPT\_DATE\_TIME", "REG\_DT\_TM" columns from object to datetime. This will create some missing values in the data as the column have “ - ” in place of missing data.
3. Imputing the missing data with unique date i.e. 1 Jan 2021.
4. Adding two new columns of the day, month in the data frame so that we can include the date and time data in our machine learning model.
5. Checking for duplicate values in the data, dropping those duplicated rows if present.
6. Encoding the data, converting the categorical variables into numerical data so that we can use the data in machine learning model.
7. Checking for outliers and removing the outliers if present in the data.
8. Using Yeo – Johnson Method to remove the skewness in the data by transforming the whole dataset.
9. Training and Testing data, after importing the files, three-month (March, April, May) data is used as training data for model training, and one-month data (June) is used for testing.

#### Machine Learning Models: Models Used

As the objective of this project is to predict the patient will show or not for the appointment, it is a classification problem. The dataset contains target variable “No Show”, wherein “0” represents that the patient has not Shown up for appointment and “1” that patient has shown up. Therefore, supervised learning algorithm needs to be applied for this prediction.

Different supervised learning algorithms in classification problems that are applied are:

1. Logistic Regression: Logistic Regression is a classification algorithm that estimates discrete values like yes/no, true/false, 0 or 1 etc. This model is most useful for understanding the influence of several independent variables on a single outcome variable. It works very well on linearly separable classes, making use of odds ratio and sigmoid function.
2. Support Vector Classifier: The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is.
3. Gradient boosting: It is a type of machine learning boosting. It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize the error. How are the targets calculated? The target outcome for each case in the data depends on how much changing that case's prediction impacts the overall prediction error:

* If a small change in the prediction for a case causes a large drop in error, then next target outcome of the case is a high value. Predictions from the new model that are close to its targets will reduce the error.
* If a small change in the prediction for a case causes no change in error, then next target outcome of the case is zero. Changing this prediction does not decrease the error.

The name gradient boosting arises because target outcomes for each case are set based on the gradient of the error with respect to the prediction. Each new model takes a step in the direction that minimizes prediction error, in the space of possible predictions for each training case.

1. AdaBoost: AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers.[1] In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.
2. KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).
3. DecisionTree: Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

#### Machine Learning Models: Best Model

The best model chosen is Logistic Regression, The Accuracy Score of the model is above 95% on testing data.

The performance metrics of the model are given as:

|  |  |
| --- | --- |
| **Performance Metrics** | **Logistic Regression** |
| **Accuracy score** | **0.999977035** |
| **f1 score** | **0.999983382** |
| **Log Loss** | **0.000793193** |
| **Cross Validation** | **0.999977035** |