**Report**

# Summary

Predictive analytics is undertaken on the data provided after undertaking a series of data understanding, pre-processing and model selection. Data understanding is undertaken under data preparation to get a clear picture with regard to statistical data summaries, missing entries in the data, duplicate row data. The imperfections noted are handled through the pre-processing of the data and put in consideration in the assessment of variables viable for predictive task. The main predictive task to be undertaken is classification, with the dependent variable a multi-label. Three classification models, the k-NN, Naïve Bayes, and the Decision Trees are used to train the data, fine-tuned to attain the best parameters to attain models better suited at prediction tasks on the train data. The two best models based on their performances are used to predict the test data provided. All steps are backed by concrete explanations with regard to the choice.

# Methodology

The data preparation section handles the identifying and removal of irrelevant attributes, detecting and handling of missing entries, duplicates in the instances and attributes, the selection of suitable data types for attributes and data transformation whenever required.

## Data Preparation

### Data Description

The data stored within a SQLite file is imported into the workspace and the first few rows explored.

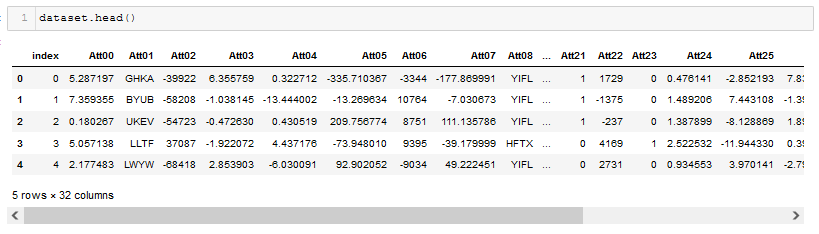


Figure 1 First 5 Observation

The dataset comprises of 1200 observations with 32 attributes. The Table 1 Data Types and Summary Statistics provides the summary statistics of numeric data columns, for which we can note the diverse ranges on the data variables and the count of non-null values within all columns.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Column | Non-Null Count | Dtypes |  | Column | count | mean | std | min | 25% | 50% | 75% | max |
| index | int64 | 1200 |  | index | 1200 | 599.5 | 346.5545 | 0 | 299.75 | 599.5 | 899.25 | 1199 |
| Att00 | float64 | 1191 |  | Att00 | 1191 | 1.606569 | 3.434696 | -12.2721 | -0.6613 | 1.663014 | 3.895994 | 13.34124 |
| Att01 | object | 1200 |  | Att02 | 1200 | -12176 | 47228.95 | -161331 | -41743 | -11304.5 | 16961.75 | 160525 |
| Att02 | int64 | 1200 |  | Att03 | 1200 | 0.639387 | 2.574271 | -5.88209 | -1.08479 | 0.558202 | 2.379273 | 10.58359 |
| Att03 | float64 | 1200 |  | Att04 | 1200 | 1.772553 | 15.93892 | -61.4574 | -8.7191 | 2.463898 | 12.94352 | 51.77887 |
| Att04 | float64 | 1200 |  | Att05 | 1200 | -79.4391 | 173.088 | -666.679 | -200.981 | -87.3746 | 32.74447 | 488.7879 |
| Att05 | float64 | 1200 |  | Att06 | 1200 | -122.075 | 5354.166 | -20126 | -3518.75 | -243.5 | 3385 | 18463 |
| Att06 | int64 | 1200 |  | Att07 | 1200 | -42.0894 | 91.70749 | -353.228 | -106.486 | -46.2938 | 17.34906 | 258.9753 |
| Att07 | float64 | 1200 |  | Att09 | 619 | 0.057382 | 3.631853 | -10.7583 | -2.54084 | 0.1042 | 2.564323 | 10.54095 |
| Att08 | object | 1200 |  | Att10 | 1200 | 62.06311 | 132.2171 | -473.29 | -25.468 | 63.5342 | 150.0783 | 514.5211 |
| Att09 | float64 | 619 |  | Att11 | 1200 | 6.314273 | 56.7784 | -218.927 | -31.0596 | 8.777016 | 46.10802 | 184.4492 |
| Att10 | float64 | 1200 |  | Att12 | 1200 | -0.43639 | 4.885329 | -14.5136 | -3.9248 | -0.21678 | 2.975106 | 15.94589 |
| Att11 | float64 | 1200 |  | Att13 | 1200 | 7.430563 | 4.295404 | 0 | 3.715282 | 7.430563 | 11.14584 | 14.86113 |
| Att12 | float64 | 1200 |  | Att14 | 1200 | 0.336955 | 37.22657 | -132.865 | -25.5117 | -0.41236 | 24.99766 | 128.7378 |
| Att13 | float64 | 1200 |  | Att15 | 1200 | 1.991142 | 4.865747 | -11.5839 | -1.34631 | 2.152275 | 5.374269 | 16.28085 |
| Att14 | float64 | 1200 |  | Att16 | 1200 | -33.9297 | 195.9738 | -882.311 | -160.577 | -41.9222 | 89.12509 | 816.5565 |
| Att15 | float64 | 1200 |  | Att17 | 1200 | 0.475639 | 20.24737 | -62.2503 | -12.8273 | -0.07558 | 14.28316 | 71.15446 |
| Att16 | float64 | 1200 |  | Att18 | 1200 | 3.129096 | 12.59823 | -28.7864 | -5.30883 | 2.731787 | 11.64393 | 51.79505 |
| Att17 | float64 | 1200 |  | Att19 | 1200 | 0.725818 | 10.85017 | -34.1983 | -6.84314 | 0.794882 | 8.116634 | 40.06815 |
| Att18 | float64 | 1200 |  | Att20 | 1200 | 4.614506 | 2.667519 | 0 | 2.307253 | 4.614506 | 6.921758 | 9.229011 |
| Att19 | float64 | 1200 |  | Att21 | 1200 | 0.488333 | 0.500072 | 0 | 0 | 0 | 1 | 1 |
| Att20 | float64 | 1200 |  | Att22 | 1200 | -368.534 | 2584.727 | -6940 | -2158.75 | -529.5 | 1051 | 9200 |
| Att21 | int64 | 1200 |  | Att23 | 1200 | 0.515833 | 0.499958 | 0 | 0 | 1 | 1 | 1 |
| Att22 | int64 | 1200 |  | Att24 | 1200 | 1.518331 | 0.877706 | 0 | 0.759166 | 1.518331 | 2.277497 | 3.036663 |
| Att23 | int64 | 1200 |  | Att25 | 1200 | 2.382607 | 9.675472 | -28.1125 | -3.59741 | 2.555237 | 8.763264 | 32.26674 |
| Att24 | float64 | 1200 |  | Att26 | 1200 | 0.13122 | 3.202912 | -10.7619 | -2.04081 | 0.114347 | 2.234868 | 10.06069 |
| Att25 | float64 | 1200 |  | Att27 | 1200 | 21.82187 | 68.80971 | -196.775 | -22.7545 | 22.7677 | 68.7521 | 272.6789 |
| Att26 | float64 | 1200 |  | Att28 | 1200 | -0.54826 | 3.844977 | -10.3225 | -3.21133 | -0.78844 | 1.563436 | 13.6837 |
| Att27 | float64 | 1200 |  | class | 1000 | 1.297 | 0.781922 | 0 | 1 | 1 | 2 | 2 |
| Att28 | float64 | 1200 |  |  |  |  |  |  |  |  |  |  |
| Att29 | object | 1200 |  |  |  |  |  |  |  |  |  |  |
| class | float64 | 1000 |  |  |  |  |  |  |  |  |  |  |

Table 1 Data Types and Summary Statistics

Missing data in the class column is due to the said observations being part of the test data used in the predictive tasks. The Att00 has 9 missing data points and Att09 has 581 missing data points. The missing data points will be handled in the pre-processing phase. A check on the duplicated observations within the data shows that there are no duplicated observations in the data.

Table 2 Categorical data Count shows the most to least frequent categories within the categorical attributes.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  | **Category** | **Att01** |  | **Category** | **Att08** |  | **Category** | **Att29** |  |
|  | LWYW | 318 |  | YIFL | 799 |  | OQDJ | 429 |  |
|  | OSUG | 282 |  | HFTX | 400 |  | YLWZ | 391 |  |
|  | BYUB | 248 |  | VEVT | 1 |  | TOYT | 205 |  |
|  | UKEV | 132 |  |  |  |  | HUUV | 136 |  |
|  | LLTF | 120 |  |  |  |  | FLJD | 25 |  |
|  | GHKA | 53 |  | **Category** | **class** |  | OELG | 13 |  |
|  | SCIJ | 36 |  | 2 | 498 |  | PJIY | 1 |  |
|  | UJJW | 5 |  | 1 | 301 |  |  |  |  |
|  | ACKH | 5 |  | 0 | 201 |  |  |  |  |
|  | TRRP | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Table 2 Categorical data Count

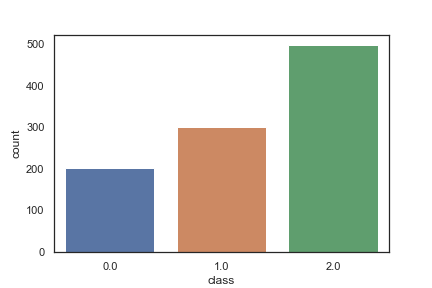


Figure 2 Class Attribute Distribution

The category 2 is the most frequent class in the attribute class.

### Data Correlation

Correlation analysis is conducted on the data to ascertain the linear relation between the attributes and is depicted in the Figure 3 Correlation Plot where there was a fairly a strong correlation between the variables:

1. Att00 and Att10
2. Att03 and Att18
3. Att04 and Att11
4. Att07 and Att05
5. Att16 and Att25
6. Att22 and Att28

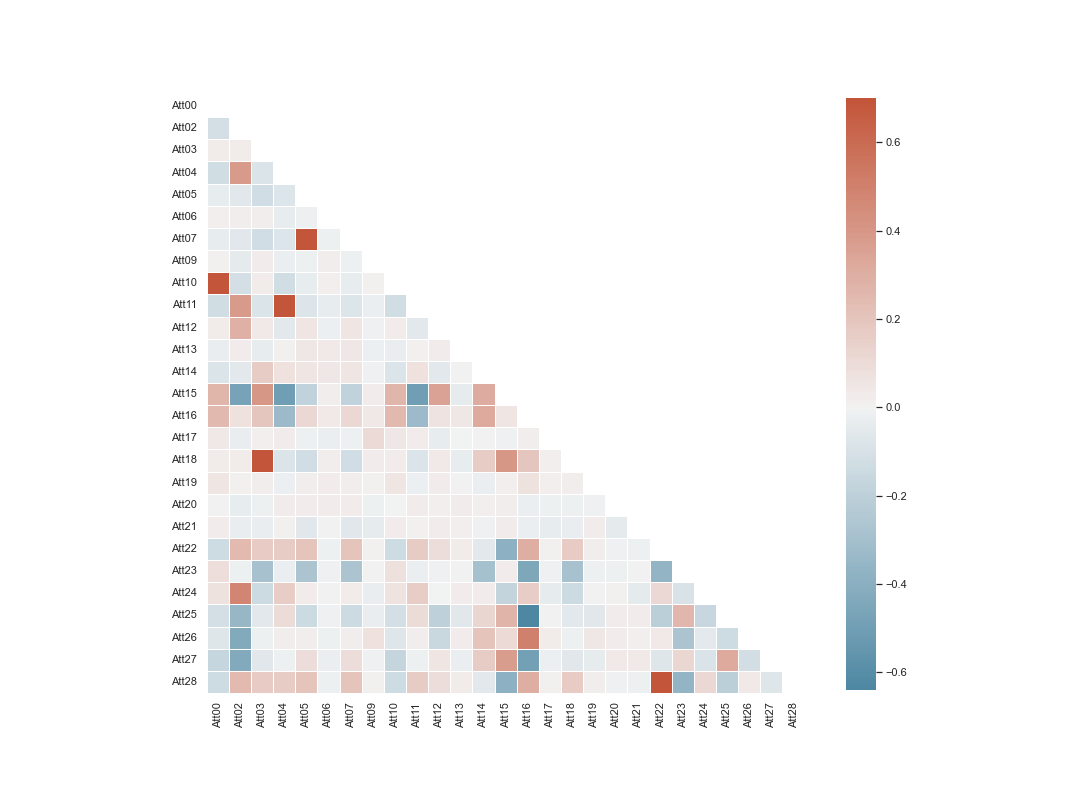


Figure 3 Correlation Plot

### Missing Data and Data Transformation

Data transformation in preparation for the classification tasks involved creating of dummy variables for the categorical variables Att01, Att08 and Att29. The creating of dummy variables was based on the fact that categorical data requires transformation to numeric for ease of handling by the classification models. The columns whose missing values percentage are way above 40% are excluded from the data, for which drops feature Att09. Similarly, missing data points in Att00 are filled with mean of the column due to the negligible missing 9 data points as opposed to Att09’s 581 missing data points.

Transformation of the data through scaling was undertaken for the k-NN and Decision Tree models, while the Naïve Bayes model was carried out with the non-transformed data. Naïve Bayes classifier transformation of data was not carried out due to the need for both positive and negative values for the model to be able to perform better on the training data.

The data was scaled, given the varying ranges and skewness observed in the data description section, and also based on the assumption that the attributes of the data may be in units of varying scales. The normalization of each of the numeric attribute with the exception of the binary attribute was undertaken with the help of the min-max scaler whose formula is as shown below:

Where the min, max = feature\_range.

### Train-Test Split

The data was split into train and test data, with the train data covering the first 1000 instance and the test data covering the last 200 instances. The split was conducted manually through indexing.

## Data Classification

Classification was conducted after the selection of the train and test data sets. The models were fine-tuned to increase model accuracy. Classifiers used in the model are the k-NN, Naive Bayes, and Decision Trees. Cross validation was incorporated into the hyper parameter tuning and modelling to improve on the training accuracy of the models.

### KNN

The k-nearest neighbors (KNN) classifier model takes a positive integer K, for which it is to identify K points closest to a test observation after which it estimates the conditional probability of a given class within the list of classes based on the frequency of the class and takes advantage of Bayes rule to classify the observation to a class that attains the highest conditional probability.

Where are K points closest to and j is a class within the dependent variables.

#### Hyperparameter Tuning

The Exhaustive Grid Search technique for hyperparameter optimization was used. The exhaustive grid search takes in various hyperparameters that are desired and tries every single possible combination within the hyperparameters provided while also undertaking cross-validations on the data. The output of the exhaustive grid search is the best hyperparameter values to input into the model. However, in the interest of time, the most basic input are included in the hyperparameters of KNN.

Parameter used herein for the KNN are:

1. n\_neighbors: The best k based on the computed values.
2. weights: The parameter checks whether the addition of weights to the instances improves model accuracy or not, with the 'uniform' assigning no weights, while 'distance' weighs points based on the inverse of the distance between them which implies that the nearer points will have more weight compared to those points farther.
3. metric: is the metric used in calculating the similarity.

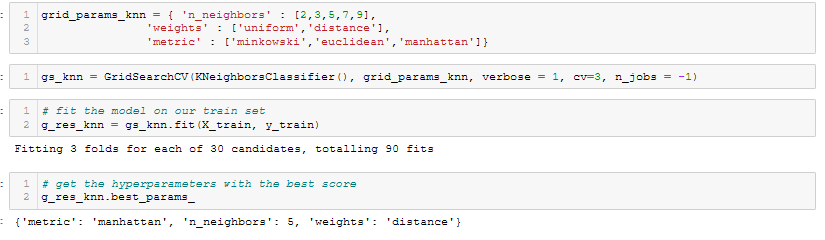


Figure 4 KNN Hyper-parameters

The hyper-parameters that provide the best models were 5 n\_neighbors, manhattan metric and distance weights. The model provided an accuracy of 100% on the training data set.

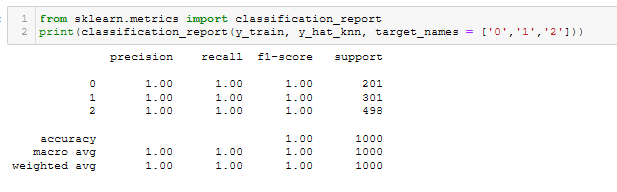


Figure 5 KNN Accuracy

### Naïve Bayes

The naive Bayes classifier works by finding instances of similar attribute class, learns of the classes in the dependent categorical variable for each of the instances and builds upon the most frequent to predict class for new test data. The classifier gives the probability of belonging to any of the available classes after which cutoff probability is used in the assigning of observations to new test data instances. There isn't a hyper-parameter to tune with the Naive Bayes as such, no grid search over was undertaken. The model provided an accuracy of 70.4% on the training data set.

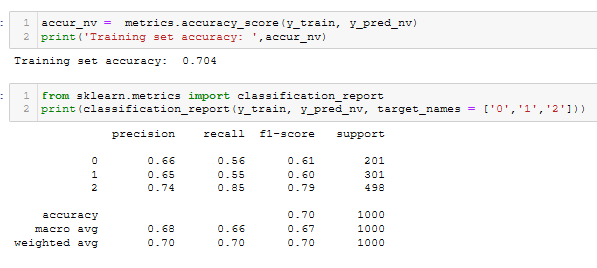


Figure 6 Naive Bayes Accuracy

### Decision Tree

Decision trees predict that each instance of test data falls into the most commonly occurring class of the training instances in the region to which it belongs. Either of the Gini index or the entropy, both of which are sensitive to the purity of the node, simply referred to as the classification error rate can be used to offer evaluation of a splits within the decision trees. For the purposes of prediction accuracy, the classification rate is used in the pruning of the classification trees.

The best hyper parameters for the Decision Tree model were at max\_depth of 9, max\_leaf of 22 and random state of 42.

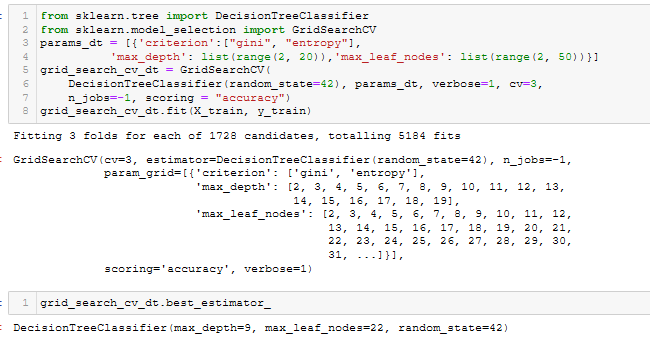


Figure 7 Decision Trees Hyper Parameters

The model provided an accuracy of 70.9% on the training data set.

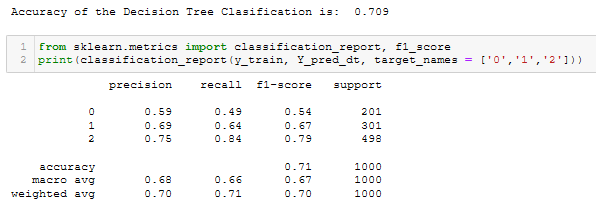


Figure 8 Decision Trees Accuracy

# Conclusion

## Model Comparison and Prediction

Model comparisons were made based on the accuracy, for which the best 2 models were selected for the purpose of predicting class labels for the 200 instances in the test data. The knn training set accuracy was at 100%, Decision Tree Classification at 70.9% and Naïve Bayes training set accuracy at 70.4%.

## Prediction

The models selected for the prediction task were the knn and Decision Tree Classification. The predictions were derived from the models attained in the classification section applied on the test data created in the data preparation stage. Predictions were made on the previously partitioned test data of the dataset based on models used to fit the train data with the tuned parameters.

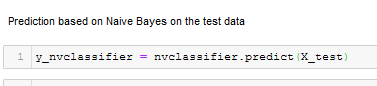


Figure 9Naive Bayes Prediction

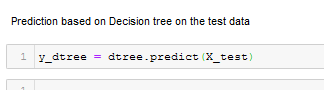


Figure 10 Decision Tree Prediction Test Data

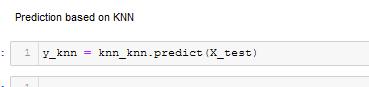
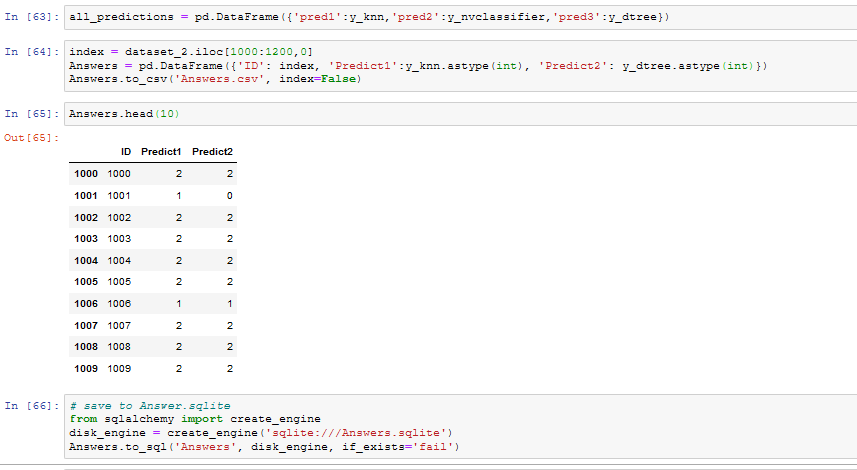


Figure 11KNN Prediction Test Data

The predicted class labels from the best 2 models, the KNN and the Decision Tree that were selected solely based on the accuracy score from fitting on the training data. The predicted variables were converted into type integer, concatenated into a data frame. The results saved in the Answers.sqlite file with the three columns, ID, Predict1 of the KNN model and Predict2 of the Decision Tree model.



Further assessment can be made to include more parameters in the tuning phase as well as extend comparisons of model effectiveness beyond only accuracy measures.

# References

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An introduction to statistical learning: With applications in R*. Springer.

*Sklearn.preprocessing.MinMaxScaler*. scikit. (n.d.). Retrieved October 16, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html.