STATISTICS AND MACHINE LEARNING

INDIVIDUAL PROJECT

MSc in Big Data Analytics 2021-22

VELLINGIRI KOWSALYA Swasthik
3-31-2022

Contents

Task 1: Mechanism of the Algorithms

1.1 Machine Learning Algorithms

The machine learning algorithms selected are,

- Logistic Regression
- Decision Trees
- K Nearest Neighbors
- Gradient Boosting
- Support Vector Machines

1.2 Mechanisms of the machine learning methods

1.2.1 Logistic Regression

Mechanism:

Logistic Regression is a Supervised machine learning algorithm that is used to predict when the dependent variable contains categorical values. The logistic regression is similar to the linear regression, but it uses the sigmoid function and logit function to classify. While linear regression is best at predicting continuous dependent variables, it does not work really well in predicting the probabilities whether the output belongs to a particular category. To be more specific, if the target variable has only 0 and 1 as values, the linear regression might predict a negative probability for some observations and may predict probability higher than 1 for some because of its linear nature.

Objective Function:

The logistic regression is based on the sigmoid function. The formula for sigmoid function is

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

We know that the objective function of linear regression is,

$$p(X) = \beta_0 + \beta_1 X.$$

By applying the sigmoid function to the objective function of the linear regression, we get the

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

function for the logistic regression.

This function can also be written as after manipulation,

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}.$$

The left-hand side of the equation is called the odds and can take values between 0 and infinity meaning lowest and highest probabilities of prediction.

By taking log on both sides in the equation, we get,

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

Now the left-hand side of the equation is called the log-odds or logit.

Fitting the Model:

There are many ways to fit the model and estimate the parameters. The most common and suitable framework for logistic regression is Maximum Likelihood Estimation. The least squares method can also be used but it is not usually preferred.

The maximum Likelihood in simple terms can be explained as the process of seeking estimates for B0 and B1 by plugging them to the function i.e. P(X), such that they are close to either 0 or 1 (for classification of yes/no). This basic definition can be written in a mathematical formula which is as follows.

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

The values of estimates B0 and B1 are chosen such that they maximize the function. After estimating the coefficients, we can make predictions.

Advantages of Logistic Regression

1. Implementing, interpreting, and training logistic regression is easier.
2. It is able to achieve good accuracy for not so complex data sets and is very efficient when the data is linearly separable.
3. It can also interpret coefficients.
4. Logistic regression is not easily overfitted but it is possible in complex datasets.
5. Multiple classes can be analyzed with ease (multinomial regression)

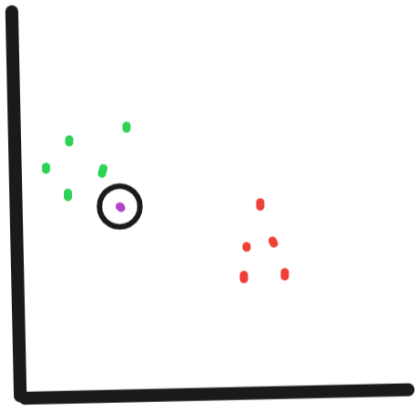
Disadvantages of Logistic Regression

1. If the number of features is higher than observations, it may lead to overfitting.
2. Building complex relationships or Models is not possible with logistic regression.
3. Difficult to capture relationships.
4. Solving non linear problems in logistic regression is not preferred.
5. Outliers in the dataset will affect the predictions made by this model.

1.2.2 K-Nearest Neighbors

K – nearest neighbors is a supervised machine learning algorithm which is used for predicting both classification and regression problems. It is most used in classification settings. As the name suggests, it considers the K number of nearest neighbors to predict the outcomes.

For example, the image below displays two different categories of data points which is green and red. We have a total of 10 data points and the violet point mentioned inside the circle is the new data point that needs to be predicted. By looking at the image, we can predict that it will belong to the green category. This is the principle behind the K-Nearest Neighbors.



K-NN works by calculating the distances and finding the minimum distances from the data point which needs to be predicted. K is the number of data points that needs to be considered in finding the category that is closer to the new data point. And hence we can conclude that if we have the distances and a preferred value for K, then we will be able to predict the category.

Distance Calculation

In order to calculate the distances, there are different ways to do it. Some of the most common distance measurements are,

- Manhattan Distance
- Euclidean Distance
- Minkowski Distance

Manhattan Distance

Manhattan distance is a method for measuring distance between two points in N-dimensional space. It is calculated as the sum of the absolute differences between the two vectors

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Euclidean Distance

Euclidean distance is calculated as the square root of the sum of the squared differences between the two vectors.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Minkowski Distance

Minkowski Distance calculates the distance between two real-valued vectors. It adds a new parameter called the order or 'p' to the existing calculations to allow different distance measures to be calculated.

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^c\right)^{\frac{1}{c}}$$

Finding Optimal K-value

Similar to choosing the right distance calculation method, it is also important to choose the K valye for the model. However, there is no statistical method available to find the right value. The only way is find the optimal K value is to fit the model for different K valye and evaluate the performance of all the models. Then by comparing the results, the model with the lowest error can be chosen.

Advantages of KNN

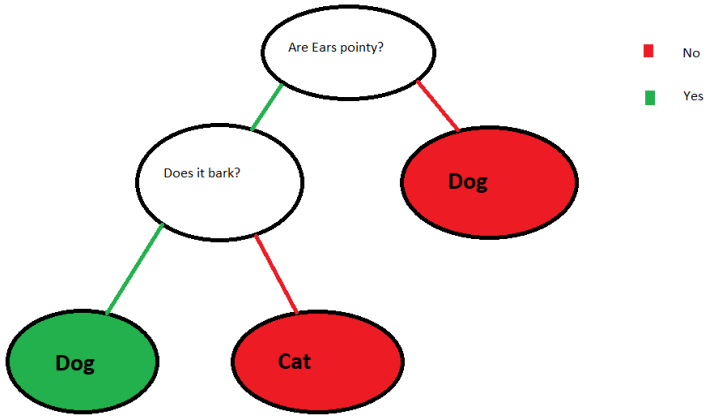
- Implementing, interpreting, and training KNN is easier.
- Many distance metrics to choose from.
- Possible to learn non linear decision boundaries with the help of K value.
- Not many parameters and hence finding K value is the only one.

Disadvantages of KNN

- KNN is not great for large complex datasets.
- Outliers might be a problem as it change the boundaries and hence can impact the predictions.
- It is necessary to do standardization and normalization for KNN.

1.2.3 Decision Trees

Decision Trees is a supervised learning algorithm that is used in both regression and classification problems. As the name suggests, the decision tree algorithm creates a training dataset model to predict based on decisions in a tree like structure. The image posted below displays how the decision tree works. The first node at the top is called the root node. And the consecutive sub nodes indicate decision rules after the rule in the root node. The predictions are the last part of each node. If there are not sub-nodes, then the branch ends there and predictions are made.



Picture Ref: https://towardsdatascience.com/3-decision-tree-based-algorithms-for-machine-learning-75528a0f03d1

Building a Decision Tree

In order to grow a classification Decision tree, we use recursive binary splitting. **Recursive binary splitting** is the process of diving the input space where different split points of the input values are tries and tested using a cost function. **Classification error rate** is used as a criterion for classification decision trees whereas RSS is used for regression decision trees.

Classification error rate in simple terms can defined as the values that don't belong to the common group in particular region. The classification error rate can be defined using the following formula,

$$E = 1 - \max_k(\hat{p}_{mk}).$$

In the equation, Pk represents the values that are present in the common class.

There are also other measures which are preferred as the classification error is not sensitive enough for building decision trees.

The Gini Index is a function that gives the information of how mixed the training data in each node are. It is also known as the measure of how “pure” the leaf nodes are. The formula for Gini Index is as follows.

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}),$$

G = 0 or lower G value indicates that a node has all classes of the same type (i.e. if the values of Pk are close to zero or one). Gini impurity lies between 0 and 0.5

There is also another alternative to Gini Index which is entropy. Entropy is another measure that is used to indicate whether a decision tree has better splits.. The formula for entropy is given by,

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}.$$

The range of Entropy is between 0 and 1 and it works similar to the Gini Index. The smaller the entropy the better is the purity of the tree. Both Gini Index and Entropy are similar based on the working of these functions.

However, since the values of Entropy first increases to 1 and then starts decreasing and whereas the values of Gini increases only upto 0.5 and the decreases, the computational power required for Gini Index is lesser and it makes the Gini Index more efficient than entropy. However, both the Gini Index and Entropy works better than the classification error rate when it comes to identifying purity of the tree.

Advantages of Decision Trees

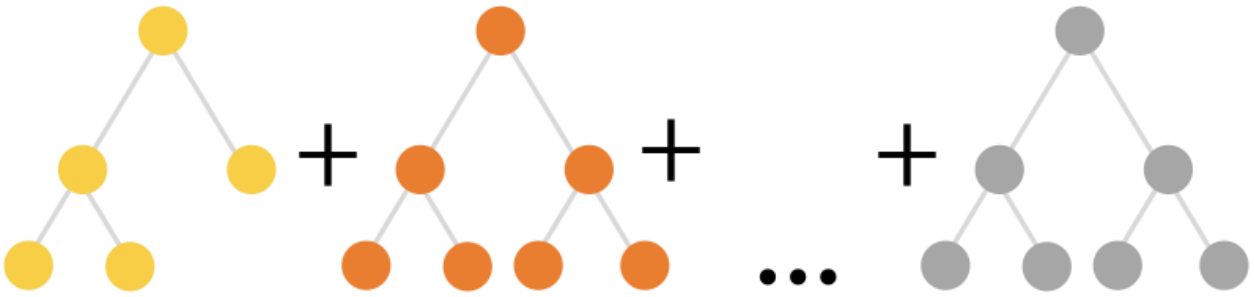
- Simple to code, Visualize and interpret.
- Decision trees can work on complex datasets with good accuracy.
- They can preform feature selection on both categorical and numerical data.
- Non linear relationships does not affect the model and the predictions
- Ability to handle categorical variables without encoding them

Disadvantages of Decision Trees

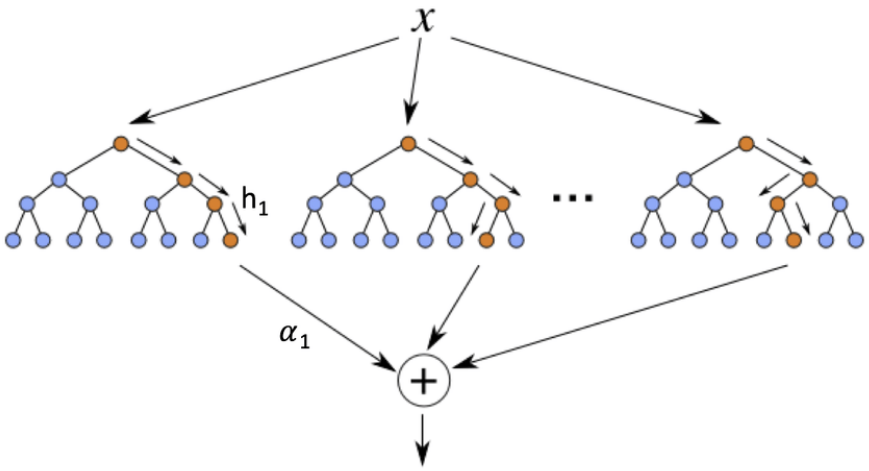
- Though trees can result in good accuracy, there are other models that can outperform.
- They are unstable i.e. changing the data in small may result in a different tree or prediction.

1.2.4 Gradient Boosting Algorithm

The decision trees above have a major problem which is overfitting. In order to overcome over fitting in decision trees, a new framework called ensemble learning was created. The idea behind ensemble learning is that instead of making predictions based on a single model, the predictions are made using a number of different models. This results in less bias and less variance. Ensemble learning has two popular methods called Bagging and Boosting. Boosting is the method where a bunch of models i.e. the trees are grown sequentially. And each tree learns from the decisions or mistakes of the previous trees. Due to the models being connected in series, gradient boosting algorithm is a bit slow to learn but can have high accuracy.



Ref: https://catboost.ai/news/catboost-enables-fast-gradient-boosting-on-decision-trees-using-gpus



Ref: https://www.researchgate.net/figure/Schematic-diagram-of-a-boosted-ensemble-of-decision-trees_fig2_325632132

In order to detect the residuals, loss function is used. For Gradient Boosting in classification tasks, logarithmic loss or log loss is used. This loss function will help us in understanding if a model is making good predictions with the trained data.

Working of Gradient Boosting Algorithm

1. The first step to make predictions is to find the log(odds) of the target variable.
2. Then the log(odds) is converted to a probability score by using logistic functions. This probability score is then used to make predictions.
   The formula for converting into a probability is as follows.

   *Formula = e * log(odds) / (1 + e * log(odds))*

3. The residuals is calculated for every observation in the training set.
4. A new decision tree is grown and predicts the previously found residuals.
5. To find the output of the tree, there is a need to convert the values using the below formula.

$$\frac{\sum Residual}{\sum[PreviousProb * (1 - PreviousProb)]}$$

Making Predictions

1. Find the log(odds) for every observation
2. Log(odds) Is converted to a probability.
3. Then the below formula is used for making predictions.

   *Predictions = base_log_odds + (learning_rate * predicted residual value)*

4. The process is repeated until a specific threshold is reached.

Advantages of Gradient Boosting Classification

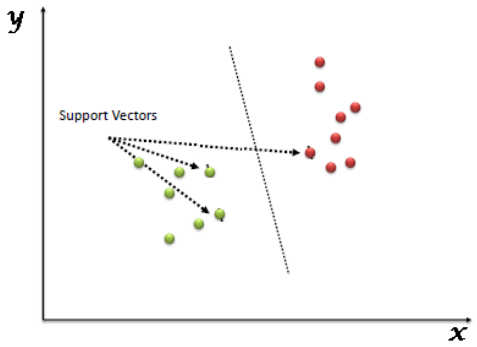
- Very flexible. Possible to optimize the loss functions and hyperparameter tuning.
- Not necessary to encode categorical data. Works without pre-processing
- Provides predictions with good accuracy.

Disadvantages of Gradient Boosting Classification

- Less interpretative.
- Overfitting is possible since the model try to keep on improving to reduce errors.

1.2.5 Support Vector Machines

Support Vector machine is another simple supervised machine learning algorithm used in both regression and classification settings and it is most widely used in classification. This algorithm works based on the principle of finding a hyperplane in an N-dimensional space that is capable to classifying the data points separately. If the data points are plotted in an N-dimensional space, then the hyperplane is used to differentiate the categories. The image below provides a basic understanding on how the support vector machine works. The hyperplane separates two classes, one being the orange data points and the other group is the green data points.

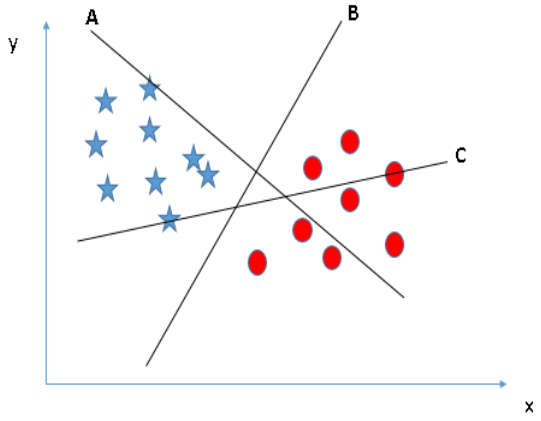


Ref: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

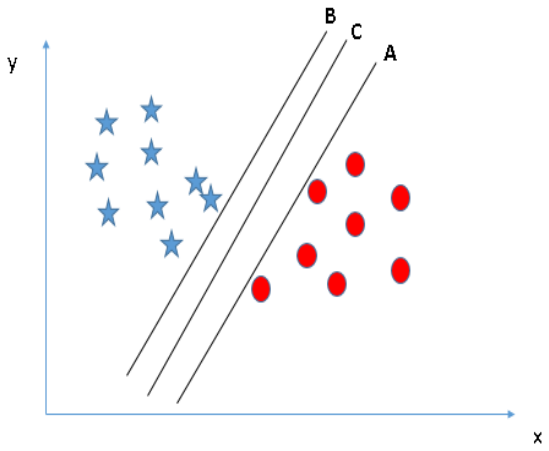
Hyperplanes are decision boundaries that helps in splitting the data points into different categories i.e. to classify. Mathematically, a two-dimensional hyperplane for three parameters can be written as,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

Now that we know how the support vector machine works, it is important to understand what hyperplane to be selected for the models. There is no perfect fixed hyperplane that can be used and hence we need to choose the one suited for the right data.



In this example image, there are two types of data points in the n-dimensional space and the best suited hyperplane is the B. The hyperplane B is good since it divides both the classes equally.



In this example, all three hyperplanes separate the data points properly. However, in order to choose the one best hyperplane, we may calculate a measure called margin. The purpose of calculating the margin is to find the hyperplane which is far away from the nearest data point of either of the class. Hence in this example, C wins.

Hyperplanes are called Maximal margin hyperplanes when the values of the margins are the largest. In simple terms, the hyperplanes that are far away from either of the data points are called Maximal margin hyperplanes. The Maximal margin hyperplanes are selected for the model based on the assumption that these hyperplanes will also have larger margins in the test data.

Advantages of Support vector Machines

- SVMs are stable. Unlike Decision trees, small change to the data does not affect the prediction outputs.
- SVMs can also handle non linear data.
- They are very efficient working with data of high dimensions
- SVMs provide great accuracies if there is a good margin between classification groups.
- It is memory efficient.

Disadvantages of Support Vector Machines

- Unlike Decision trees, SVM Models are difficult to interpret.
- When there is no clear margin between data points, SVM does not work well.
- SVM models are not great in large data sets.
- If the number of variables are higher than number of observations, then SVM will not work well.
- In SVMs, there is no probabilistic clarification for the classifications made.

Task 2: Benchmarking the experiment

2.1 Dataset Information

The Dataset data set is from a Portuguese banking institution with the information of whether a telemarketing (phone calls) campaign was successful or not. The dataset has 20000 observations and 20 variables. The target variable is the “Subscribe” column.

2.2 Goal of the Experiment

The goal of the experiment is to predict the success of the telemarketing campaign using 5 different machine learning algorithms and compare its results.

2.3 Data Preprocessing Steps

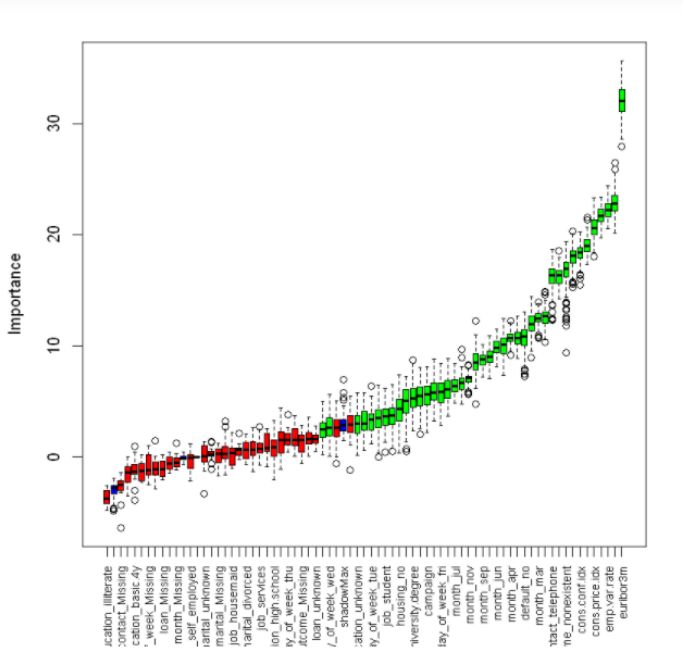
1. Identified the numerical variables and categorical variables in the dataset.
2. Checked the missing values in the dataset.
3. Filled the missing values with 0 for the numerical variables and “missing” for the categorical variables.
4. All the categorical values are encoded.
5. Names of the columns in the data are changed according to the requirements
6. The final base table has 20000 observations and 73 variables

2.4 Feature Selection

Feature selection is one of the important steps while building a machine learning model. It improves the performance and the prediction of the models by removing the variables that are not relevant or not contribute much to the outcome. It helps reducing the dimensions of the dataset without losing the features that impact the predictions very much. The feature selection algorithm used in this experiment is **Boruta**.

The working of Boruta is explained below,

- First, Shadow features i.e. shuffled copies of the features in the dataset are created.
- Then a Random forest classifier is trained on the created dataset.
- Mean Decrease accuracy or the feature importance is calculated to evaluate the importance of the features.
- The real features should have a higher importance than the shadow features. This condition is checked in every iteration and the real features whose Z score is not higher than the maximum Z score of its shadow features are considered not important.
- Once the random forest runs are complete, this process stops. This process can stop before the completion when all the features are either confirmed or rejected.



This plot displays the importance of all the features calculated by the Boruta algorithm. The features in the green color are considered important and hence confirmed. The features in the red color are rejected. The Blue box colored box plots indicate shadow features.

Implementing Boruta on the base table, 42 variables are found to be important. The remaining variables are removed from the dataset.

2.5 Test Train Split

The base table created is then split into train and test sets. The train dataset will have 70% of the observations in the base table and the test dataset will have 30% of the observations in the base table.

The number of observation in train are 14000.
The number of observations in test are 6000.

2.6 Machine Learning Models

2.6.1 Logistic Regression

The experiment using the logistic regression machine learning method is discussed in points below.

- K-fold Cross Validation is applied on the train data set. Number of iterations considered in the cross validation is 100. The aggregated mean of test AUC is 0.775 and the train AUC is 0.780.
- Then a model is built by re-training the model.
- The test dataset and the train dataset is then predicted using the model trained.
- The accuracy for test dataset = 90.16 % and train dataset = 90.46%
- The AUC score for test dataset = 0.6997 and train dataset = 0.8794

2.6.2 Decision Trees

The experiment using the Decision Trees machine learning method is discussed in points below.

- K-fold Cross Validation is applied on the train data set. Number of iterations considered in the cross validation is 30. The aggregated mean of test AUC is 0.692.
- Hyper Parameter Tuning was also done during the cross validation process and the optimal parameters are found to be
    - maxdepth=5; minsplit=25

- Then a model is built by re-training the model with the best hyper parameters.
- The test dataset is then predicted using the decision trees model trained.
- The accuracy for test dataset = 90.150% and train dataset = 90.457%
- The AUC score for test dataset = 0.6995 and train dataset = 0.8796

2.6.3 K- Nearest Neighbors

The experiment using the KNN classifier machine learning method is discussed in points below.

- K-fold Cross Validation is applied on the train data set. Number of iterations considered in the cross validation is 10. The aggregated mean of test AUC is 0.895.
- Hyper Parameter Tuning was also done during the cross-validation process and the optimal parameters are found. The best K value found was
    - K = 100

- Then a model is built by re-training the model with k value as 100.
- The test dataset is then predicted using the KNN model trained.
- The accuracy for test dataset = 90.08% and train dataset = 89.93%
- The AUC score for test dataset = 0.8062 and train dataset = 0.8067

2.6.4 Gradient Boosting Classifier

- K-fold Cross Validation is applied on the train data set. Number of iterations considered in the cross validation is 5. The aggregated mean of test AUC is 0.895.
- Hyper Parameter Tuning was also done during the cross-validation process and the optimal parameters are found. The parameter values are,

```
n.trees=1000; interaction.depth=45; train.fraction=1; shrinkage=0.001
```

- Then a model is built by re-training the model with the best hyper parameters.
- The test dataset is then predicted using the Gradient Boosting model trained.
- The accuracy for test dataset = 90.183% and train dataset = 90.45%
- The AUC score for test dataset = 0.6997 and train dataset = 0.8804

2.6.5 Support Vector Machines

The experiment using the Support Vector Machines method is discussed in points below.

- K-fold Cross Validation is applied on the train data set. Number of iterations considered in the cross validation is 10. The aggregated mean of test AUC is 0.716 and train AUC is 0.914.
- Then a model is built by re-training the model.
- The test dataset is then predicted using the Support vector machines model trained.
- The accuracy for test dataset = 89.933% and train dataset = 89.642%
- The AUC score for test dataset = 0.7843 and train dataset = 0.7798

2.7 Results Comparison

This section shows the accuracy and AUC scores of the predictions made my 5 machine learning models discussed above. The accuracies and AUC scores are calculated based on the predictions made on the test dataset considering the probability threshold above 0.5 as successful.

2.7.1 Predictions Made on Train Dataset

Models	Accuracy_in_perct_train	AUC_scores_train
<chr>	<dbl>	<dbl>
Logistic Regression	90.46429	0.8794195
Decision Trees	90.45714	0.8796736
K-Nearest Neighbors	89.93571	0.8067439
Gradient Boosting	90.45000	0.8804134
Support Vector Machines	89.64286	0.7798907

When the accuracies and AUCs were calculated on the train set, all the models performed almost equally well. However, models such as Logistic Regression, Decision Trees and Gradient Boosting models have high accuracies as well as high AUC scores.

2.7.2 Predictions Made on Test Dataset

Models	Accuracy_in_perct_test	AUC_scores_test
<chr>	<dbl>	<dbl>
Logistic Regression	90.16667	0.6997270
Decision Trees	90.15000	0.6995627
K-Nearest Neighbors	90.08333	0.8062460
Gradient Boosting	90.18333	0.6997898
Support Vector Machines	89.93333	0.7843380

When the models were used to predict the test dataset, most of the models showed good results in the accuracies. However, the AUC scores varied a lot. Accuracies of some of the models were higher than the accuracies when predicted with train dataset. However, models such as Support Vector Machines and KNN show steady outputs and can be considered as the better models as they have high scores in both train datasets and testdata sets

Conclusion

In this report, mechanisms of 5 most commonly used classification machine learning methods was explained with its objective functions. The advantages and disadvantages of these models were also discussed. In the second part of the report, the benchmarking experiment using marketing dataset of a Portuguese banking institution was performed using the 5 different machine learning methods. The accuracy and AUC scores were calculated for both train and test datasets. The best performing models were identified as Support Vector Machines and K-Nearest Neighbors.

References

Logistic Regression

1. https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

Decision Trees

1. https://towardsdatascience.com/3-decision-tree-based-algorithms-for-machine-learning-75528a0f03d1
2. https://www.geeksforgeeks.org/gini-impurity-and-entropy-in-decision-tree-ml/?ref=rp

KNN

1. https://machinelearningmastery.com/distance-measures-for-machine-learning/

Gradient Boosting

1. https://towardsdatascience.com/gradient-boosting-classification-explained-through-python-60cc980eeb3d
2. https://catboost.ai/news/catboost-enables-fast-gradient-boosting-on-decision-trees-using-gpus
3. https://www.researchgate.net/figure/Schematic-diagram-of-a-boosted-ensemble-of-decision-trees_fig2_325632132

SVM

1. https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

Boruta

1. https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/
2. https://cran.r-project.org/web/packages/Boruta/Boruta.pdf