COSC2670 – PRACTICAL DATA SCIENCE
ASSIGNMENT – 1
DATA CLEANING AND SUMMARISING
PROJECT REPORT

AUTHORS: SURAJ KANNAN & JAIRAJ ALAGU PONNIAH
DATE OF PUBLICATION: 21/05/2018

CONTACT DETAILS:

STUDENT ID: S3668855
EMAIL ID: s3668855@student.rmit.edu.au

STUDENT ID: S3700757
EMAIL ID: s3700757@student.rmit.edu.au

TABLE OF CONTENTS

ABSTRACT

The main moto of this report is to examine and interpret the various models like decision tree, k-near neighbours and random forest and determine which is the most efficient and successful model for the chosen “Cars Evaluation” dataset. The dataset is perfectly cleaned and is ready for analysis i.e., is free from disruptions that hinder the data exploration process such as missing values, errors etc and hence doesn’t need any sanitary checks. Presuming there is no need for data cleaning, the three models namely, Decision tree, Random forest and K-nearest Neighbors algorithms are applied on the data to conclude which of the three machine learning models is most suitable for analysing the “Cars Evaluation” dataset. Cross-validation is done to the modelled data from all three techniques, as this factor plays a very prominent role in determining the precision of the models.

INTRODUCTION

The data which is to be modelled is taken from the UCI machine learning repository. The dataset is about the cars. Derived from simple hierarchical decision model, this dataset is useful for testing constructive induction and structure discovery methods. Car Evaluation Dataset was derived from a simple hierarchical decision model originally developed for the demonstration of DEX, M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.)
The model evaluates cars according to the following concept structure:

1. CAR: car acceptability

1.a. PRICE overall price

1.b. buying price

1.c. maint price of the maintenance

2. TECH technical characteristics

2.a. doors number of doors

2.b. persons capacity in terms of persons to carry

2.c. lug_boot the size of luggage boot

2.d. safety estimated safety of the car

Input attributes are printed in lowercase. Besides the target concept (CAR), the model includes three intermediate concepts: PRICE, TECH, COMFORT. The dataset contains 1727 counts in each column,6 features and 4 targets. The Car Evaluation Dataset contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes: buying, maint, doors, persons, lug_boot, safety. This report will narrate how the data is read, modelled using random forest, Support Vector Machine and decision tree algorithms. Since the dataset is checked for errors, missing values and sanitary checks, then is directly taken into modelling and visualising stage. This report will also show which of the three machine learning models is best suited for the cars evaluation dataset.

METHODOLOGY

Pandas feature in python is the main tool for analysis performed with the selected dataset. Pandas is a software library in python and is used in data manipulation and data analysis at a large scale. Pandas offers various features which includes data structures and operations for manipulating numerical tables and even time-series data. The Pandas library features include Dataframe, which is an object used for manipulation, tools for reading, writing data between different file formats, Data alignments, Data reshaping, indexing, subsetting and lots more.

Another feature of python which is closely integrated with pandas is the “Seaborn” which is used to build beautiful data visualisations. It is very effective for data exploration, especially in creating histograms, scatterplots, boxplots and heatmaps.

NumPy is another library for the python programming language, which is used to handle large multidimensional arrays and matrices. It also has an enormous collection of high-end mathematical functions to handle these arrays. The heart of NumPy’s functionality lies in the concept of the “Nd-array”, also known as n dimensional array.

Scikit learn is a library in python which features various types of classification, clustering and regressions techniques including random forests, gradient boosting and many more. Finally, the analysed data is pictured and visualised using a feature in python called as matplotlib. It is the plotting library in Python programming language.

The whole project starts with loading the precleaned data into a pandas dataframe. It is then separated into two datasets (training set and test set) and fed into the three machine-learning models. The modelled data is then cross-validated and compared with the precision value obtained from the modelled data. If the variance between the cross-validated value and the precision value is negligible then the model is considered good. However, if the variance is high then the process is repeated to finetune and model the data again, till the variance is negligible. Finally, the precision value from the three modelled data is compared and the modelled data with the highest precision value is awarded the perfect model for the “Cars Evaluation” dataset.

RESULTS

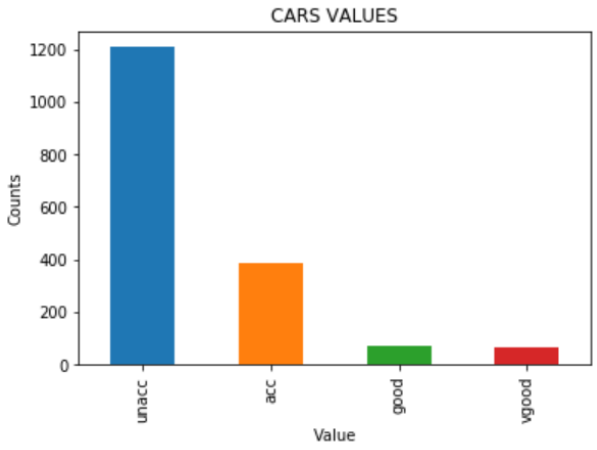
TASK 1: DATA EXPLORATION

- The pandas, NumPy, Scikit learn and other packages are imported and the dataset cars.txt is read into the “cars” dataframe.
- Appropriate descriptive statistics such as class types, shapes, data types and other information related to the dataframe are depicted.
- Pie charts and histograms with appropriate labels on the x-axis and y-axis, a title, and a legend are used to visualise each attribute of the dataset with appropriate colours applied wherever necessary.
- In addition, the data is encoded and applied on the dataset and is visualised in a heatmap. Finally, relationship between the pair of attributes is also pictured in a single graph containing multiple histograms.

.

a)

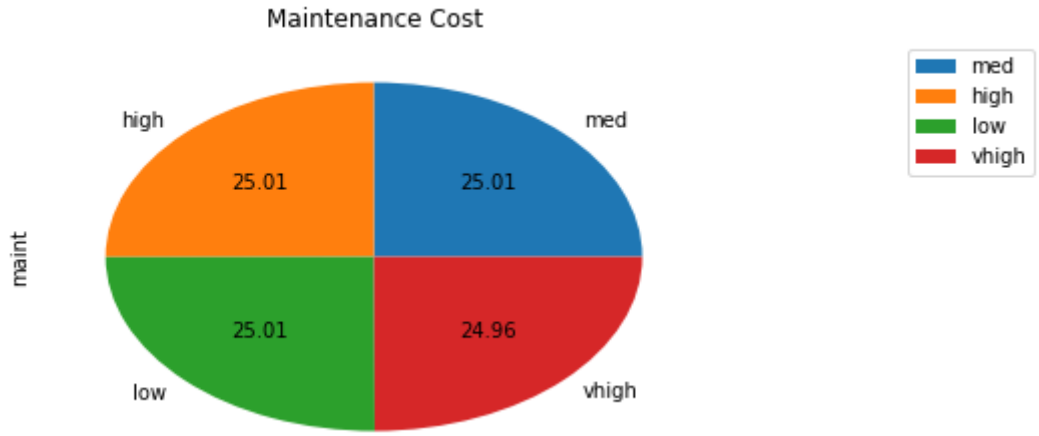
This graph is plotted for the frequency of price. We can observe the



b)

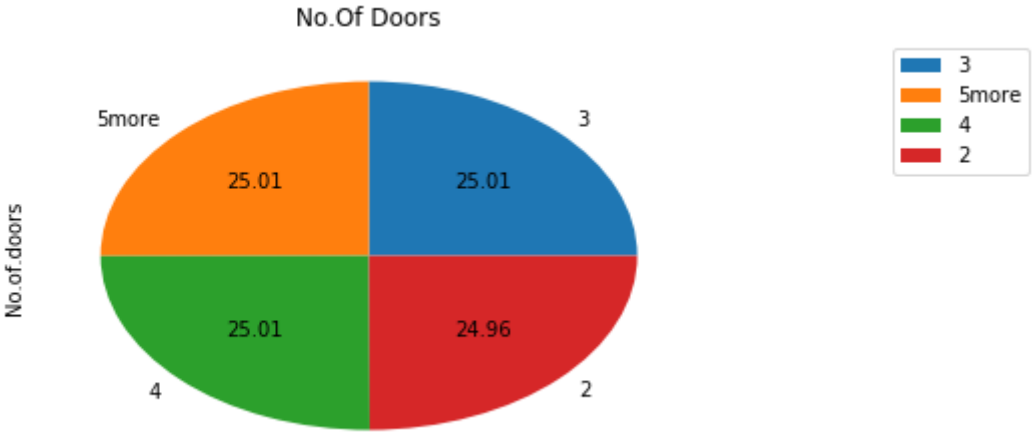
This graph is plotted for the frequency of the Cars Final Valuation .

.



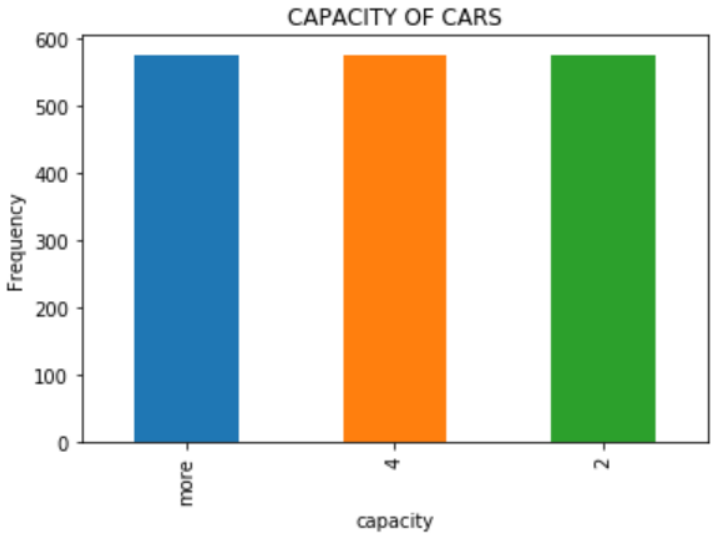
c)

This graph is plotted for the distribution of the Cars Maintenance cost.



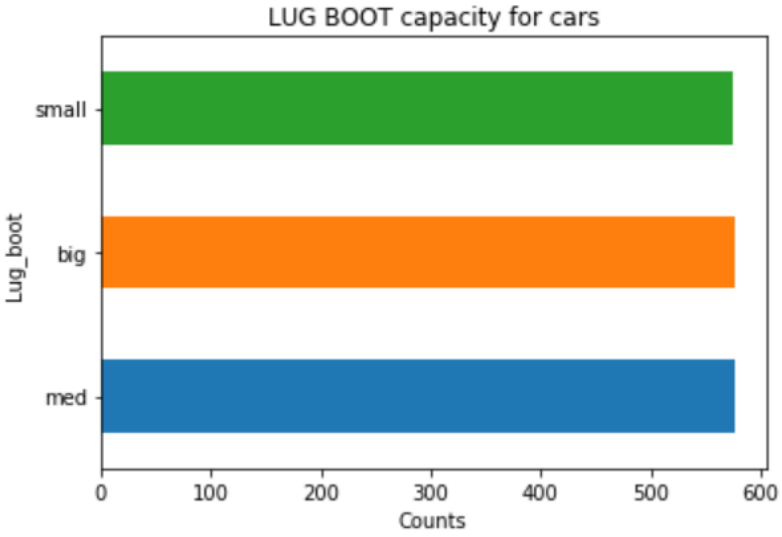
d)

This graph is plotted for the distribution of the Cars No.of Doors.



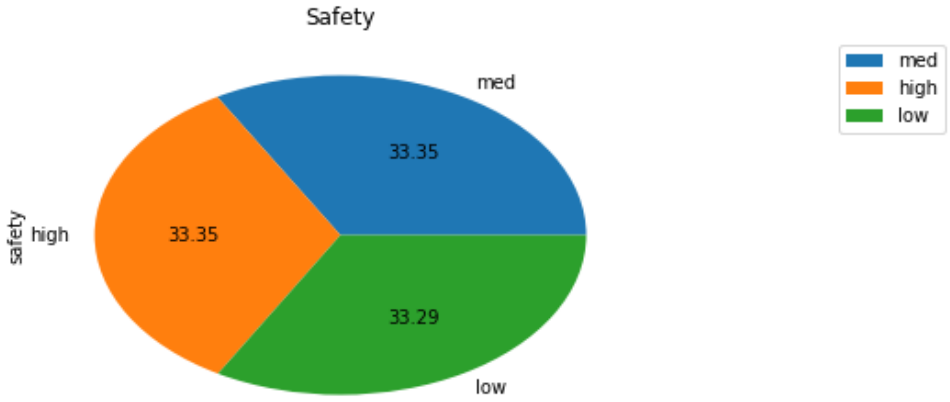
e)

. This graph is plotted for the frequency of the Car's capacity.



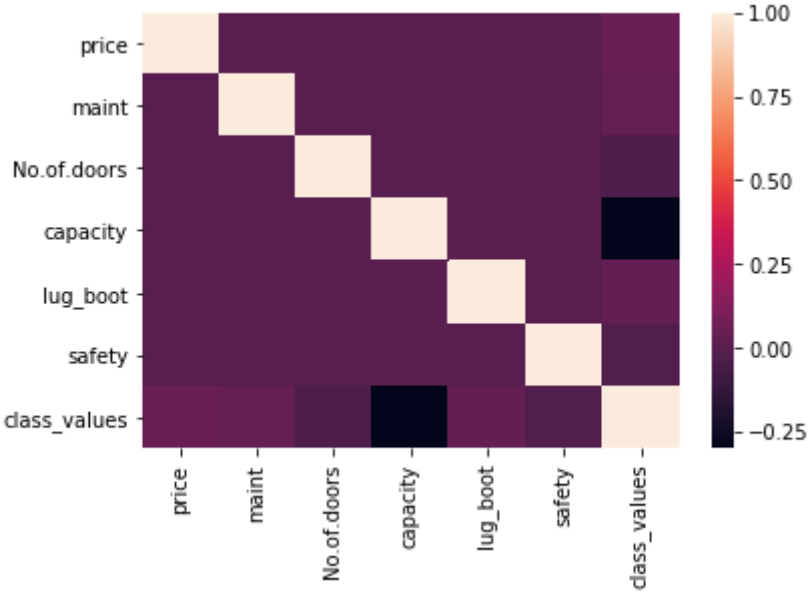
f)

This graph is plotted for the frequency of the Lug_boot space .



g)

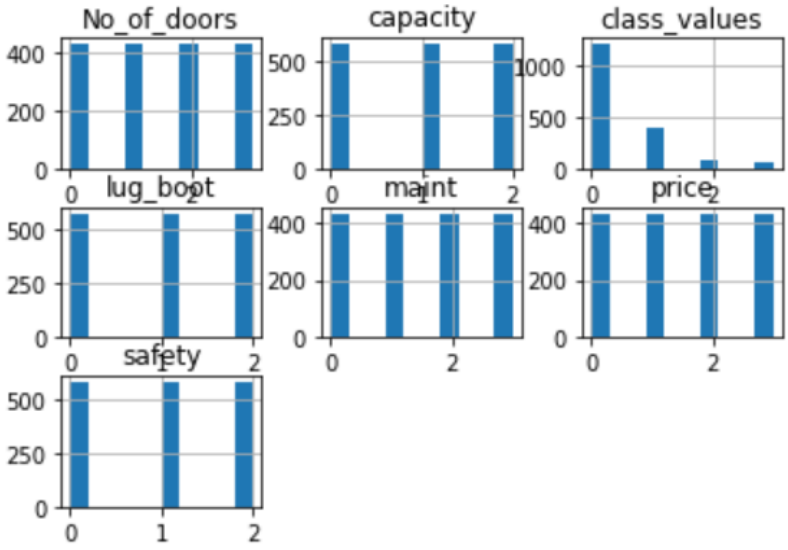
. This graph is plotted for the distribution of the Car's safety.



h)

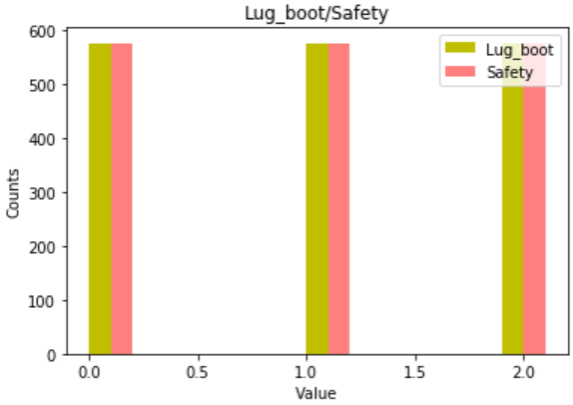
i) This graph gives the significance level of the target and features.

.



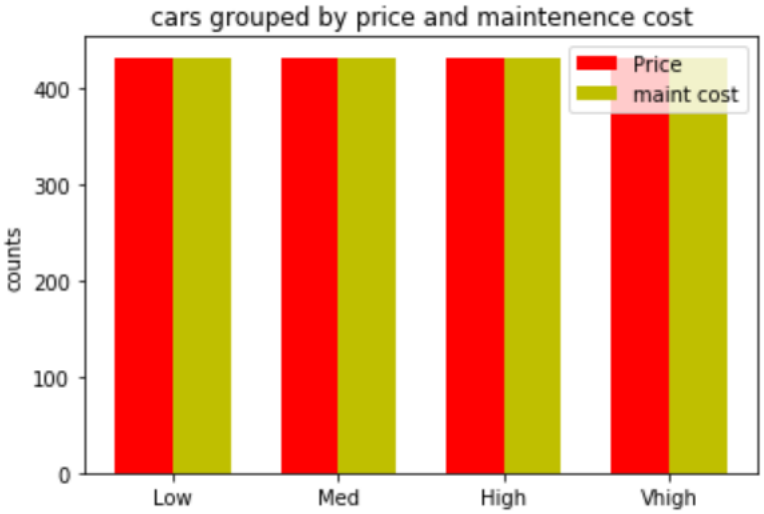
j)

k) This graph clearly tells us that there is no change in the distribution of the features but there is skewed distribution found in the target variable and hence we can conclude that the combination of features is responsible for the change in the target .



l) .

This graph is plotted for the combined distribution of the Car's Lug_boot and safety.

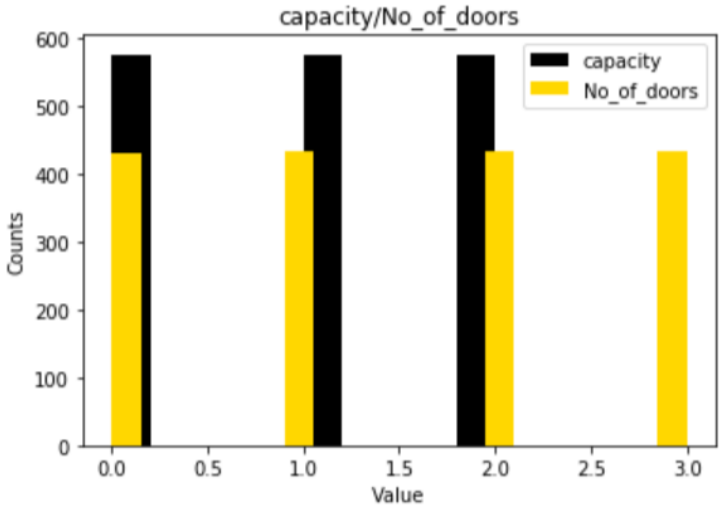


m)

n) This graph is plotted for the combined distribution of the Car's Price and Maintenance cost.

.

L)



This graph is plotted for the combined distribution of the Car’s Capacity and No.of doors..

TASK 2: DATA MODELLING

Depending upon the dataset previously selected the classification technique is used for modelling the dataset. Three classification models, namely Decision tree, Random forest and k-nearest Neighbors are chosen for data modelling. The following steps are done in order to model the given dataset:

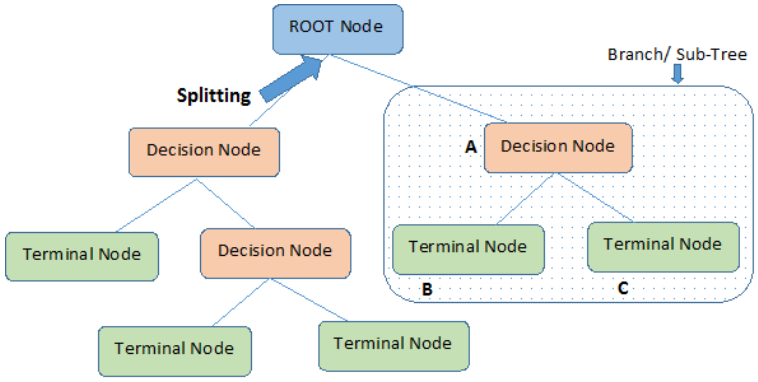
a) The dataset is split into “Test Dataset” and “Training Dataset” (70/30).
b) The model is trained using the test dataset.
c) The accuracy of the model on the test dataset is tested and the performance of the model is reported on the following factors:
    i. Confusion Matrix
    ii. Classification error rate
    iii. Precision
    iv. Recall
    v. F1-Score

1. Decision tree

To start with, Decision tree is one of the most popular and best predictive modelling approaches used in machine learning (supervised learning), data mining and in advances statistics. Decision tree algorithms come handy in solving any kind of real life data science problems. Even problems of classification and regression can easily be solved using this concept. Decision tree are of two types, namely Classification tree (when the predicted final result belongs to the class of the original data) and Regression tree (when the predicted result can be a real number). In this technique, the sample data is split into more than two sets/branches based on a most significant differentiator present in the input data. They are simple to compute and interpret and also can handle both categorical and numerical data. Also, very large amounts of data can be analysed and interpreted in a very short period of time which aids human decision making more precise than other techniques. Even though Decision tree approaches have a lot of advantages, they also have a considerable number of drawbacks such as they can be very non-robust i.e., a small change in the training data can make a very huge change in the tree and in the final predictions. Another major drawback is the major problem of overfitting. A lot of data mining software packages provide implementations of these decision tree algorithms, such as Salford Systems, Matlab, SAS Enterprise Miner and R .

Terminologies in Decision tree:

- **Root Node:** This depicts the whole population or data sample and it gets further classified into more than two homogenous data sets.
- **Splitting:** This is the process of classifying a node into two or more subdivided nodes
- **Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node.
- **Leaf/ Terminal Node:** Nodes do not split is called Leaf or Terminal node.



Note:- A is parent node of B and C.

- **Pruning:** The process of omitting/removing the subdivided nodes of a decision tree. This is the inverse process of splitting.
- **Branch / Sub-Tree:** The subdivided part of an entire decision tree.
- **Parent and Child Node:** When a node is subdivided into two more nodes it is the parent node, on the other hand, the subdivided nodes are the child.

Confusion Matrix

```
[[370   2   0   0]
 [  6 108   0   0]
 [  0   2  16   0]
 [  0   1   0  14]]
```

The above shows the confusion matrix attained from modelling the data using Decision tree.

	precision	recall	f1-score	support
0	0.98	0.99	0.99	372
1	0.96	0.95	0.95	114
2	1.00	0.89	0.94	18
3	1.00	0.93	0.97	15
avg / total	0.98	0.98	0.98	519

This is the classification report of the decision tree. This shows that the precision is 98% and the recall, f1- score are 98% and 98% respectively. The f1- score is calculated as the average of precision and recall.

Classification-error rate:

```
0.021194605009633882
```

In order to verify the accuracy score we use K-Fold cross validation(K=5).

```
[fold 0] score: 0.89595
[fold 1] score: 0.89595
[fold 2] score: 0.99130
[fold 3] score: 0.79130
[fold 4] score: 0.86667
```

As the K-Fold cross Validation result clearly tells us that the decision tree is overfitting. In order to prevent the overfitting of the model we change the parameters and tune the decision tree classifier.

Finally after changing the parameters criterion='gini', splitter='best', max_depth=5, min_samples_split=3, min_samples_leaf=2, min_weight_fraction_leaf=0.0, max_features=None, random_state=None,max_leaf_nodes=None,min_impurity_decrease=0.0, in_impurity_split=None, class_weight='balanced', presort=False, we get the confusion matrix as

```
array([[326,  42,   4,   0],
       [  0,  76,  27,  11],
       [  0,   0,  15,   3],
       [  0,   0,   0,  15]], dtype=int64)
```

Classification Report

	precision	recall	f1-score	support
0	1.00	0.88	0.93	372
1	0.64	0.67	0.66	114
2	0.33	0.83	0.47	18
3	0.52	1.00	0.68	15
avg / total	0.88	0.83	0.85	519

Classification Error Rate

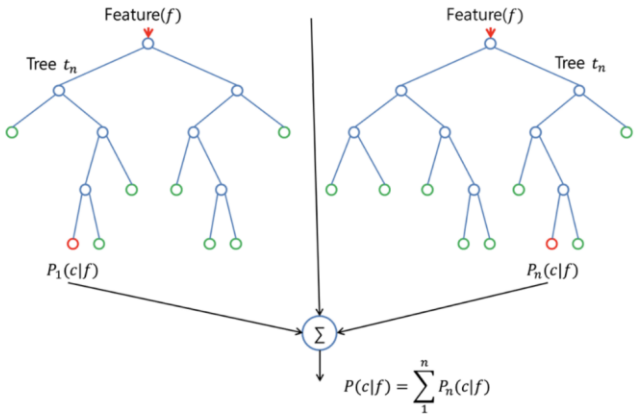
```
0.16763005780346818
```

Cross validation test report

```
[fold 0] score: 0.89595
[fold 1] score: 0.89595
[fold 2] score: 0.99130
[fold 3] score: 0.79130
[fold 4] score: 0.86667
```

2. Random forest

Random forest is capable of performing both regression and classification. It is a highly versatile machine learning technique and it can handle large number of features. It is also extremely accurate is estimating which of the selected variables is important in the data which is modelled. It can be used to solve any kind of data science problem with a combination of several models for a prediction. Below you can see how a random forest would look like with two trees:



Random Forests are also very hard to beat in terms of performance. Of course you can probably always find a model that can perform better, like a neural network, but these usually take much more time in the development. And on top of that, they can handle a lot of different feature types, like binary, categorical and numerical. Overall, Random Forest is a (mostly) fast, simple and flexible tool, although it has its limitations.

Confusion Matrix

```
array([[370,   2,   0,   0],
       [  6, 108,   0,   0],
       [  0,   1,  17,   0],
       [  0,   1,   0,  14]], dtype=int64)
```

Classification Report

	precision	recall	f1-score	support
0	0.98	0.99	0.99	372
1	0.96	0.95	0.96	114
2	1.00	0.94	0.97	18
3	1.00	0.93	0.97	15
avg / total	0.98	0.98	0.98	519

This is the classification report of the decision tree. This shows that the precision is 98% and the recall, f1- score are 98% and 98% respectively. The f1- score is calculated as the average of precision and recall.

```
[fold 0] score: 0.89595
[fold 1] score: 0.89595
[fold 2] score: 0.99130
[fold 3] score: 0.79130
[fold 4] score: 0.86667
```

As the K-Fold cross Validation result clearly tells us that the random forest is overfitting. In order to prevent the overfitting of the model we change the parameters and tune the random forest classifier. After tuning the parameters (n_estimators=10, criterion='gini', max_depth=5, min_samples_split=3, min_samples_leaf=3, min_weight_fraction_leaf=0.05, max_features='auto', max_leaf_nodes=None, bootstrap=True, oob_score=False, n_jobs=4, random_state=None, verbose=0, warm_start=False, class_weight='balanced'), we get the result as

Confusion Matrix

```
array([[315,  44,  13,   0],
       [  0,  96,  17,   1],
       [  0,   0,  12,   6],
       [  0,   0,   0,  15]], dtype=int64)
```

Classification Report

	precision	recall	f1-score	support
0	1.00	0.85	0.92	372
1	0.69	0.84	0.76	114
2	0.29	0.67	0.40	18
3	0.68	1.00	0.81	15
avg / total	0.90	0.84	0.86	519

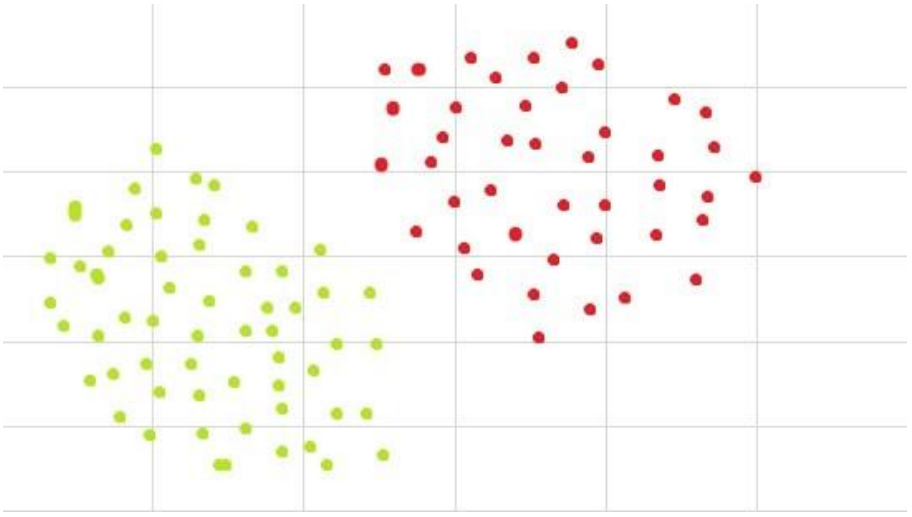
Cross validation test result

```
[fold 0] score: 0.67341
[fold 1] score: 0.83526
[fold 2] score: 0.73913
[fold 3] score: 0.74493
[fold 4] score: 0.69275
```

3. **Support Vector Machine**

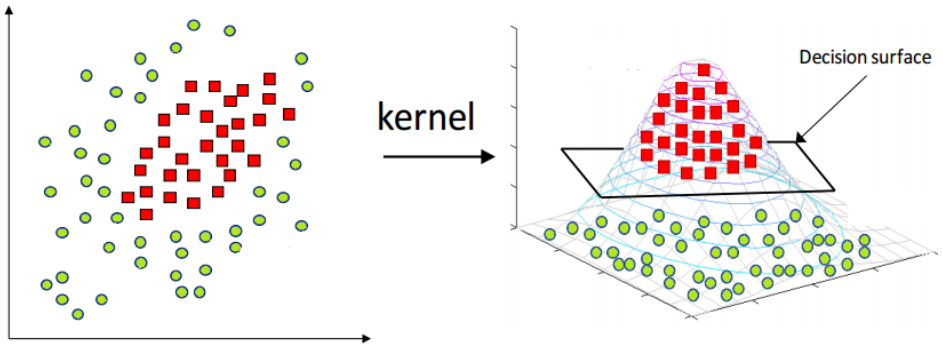
Support vector machines have extensive applications in the field of pattern classification and in nonlinear regressions. SVMs are used mainly in real world problems like categorizing text, images and also in hand written character recognition. It is also having its wide applications in the area of bioinformatics.

Support vector machine is a supervised learning algorithm, which can be very useful in solving the real-world problems of classification and regression. The technique used in SVM is the kernel trick, which transforms the data and based on these transformations, it finds an optimal boundary between the possible outputs. Below, is the example:



The main idea is to identify the optimal separating hyperplane which maximizes the margin of the training data.

The way to separate two classes of data is a line in case of 2D data and a plane in case of 3D data. This is accomplished using a kernel function which maps the data to a different space, where a linear hyperplane can be used to separate classes. The kernel function transforms the data into the higher dimensional feature space so that a linear separation is possible.



To summarise, SVM is an algorithm which is suitable for both linearly and nonlinearly separable data and also work well on small as well as high dimensional data spaces. Finally, SVMs can work effectively on smaller training datasets as they don't rely on the entire data.

After the parameter tuning of the svm classifier

```
SVC(C=2.0, cache_size=100, class_weight=None, coef0=0.0,
  decision_function_shape='ovo', degree=3, gamma='auto', ker
nel='rbf',
  max_iter=-1, probability=True, random_state=None, shrinkin
g=False,
  tol=0.001, verbose=False)
```

Confusion Matrix

```
array([[366,   5,   1,   0],
       [  4, 107,   3,   0],
       [  0,   1,  15,   2],
       [  0,   1,   0,  14]], dtype=int64)
```

Classification Report

	precision	recall	f1-score	support
0	0.99	0.98	0.99	372
1	0.94	0.94	0.94	114
2	0.79	0.83	0.81	18
3	0.88	0.93	0.90	15
avg / total	0.97	0.97	0.97	519

Error Rate:

```
0.032755298651252374
```

Cross validation test result

```
[fold 0] score: 0.98266
[fold 1] score: 0.96243
[fold 2] score: 0.92464
[fold 3] score: 0.87246
[fold 4] score: 0.85217
```

DISCUSSION

The main reason behind performing this type of comparative modelling analysis is to find which machine learning model fits perfectly and predicts the output “class_values” with high accuracy rate and extremely feeble variance in the cross-validation report. These two factors are one of the main factors in deciding whether a machine learning model is best suited for predicting the “targets” for the particular dataset. Since the dataset has been taken from the UCI repository, the dataset was already cleaned and without errors. The model upon successive tunning can be made into a perfect model for predicting future data with high accuracy.

The advantage of modelling is that, any kind of data can be predicted if the model is properly trained and tunned to perfection. It can be used to solve numerous real-world problems.

Take for instance, the above 3 models were trained and tunned to predict the target i.e., “class_values” using the train and test dataset. Finally, the SVM was successful in predicting the target values and the other two models were not so successful in predicting the values for this particular dataset.

CONCLUSION

Upon performing analysis on the 3 models the following are the conclusions:

- The accuracy rate of decision tree was 98%, but on seeing the cross validation report, we come to a conclusion that it overfits the model. After tunning the parameters of the model, the accuracy rate was found to be nearing 88%.
- The accuracy rate of random forest was 98%, but on seeing the cross-validation report, we conclude that it overfits the model. After tunning the parameters of the model, the accuracy rate was found to be nearing 90%.
- Finally, support vector machines the accuracy was 95%, upon further tunning the accuracy rate was 97% with a valid cross validation score.
- And hence, after the three models were analysed with the cars dataset, support vector machine model fits perfectly for analysis.

REFERENCES

- https://en.wikipedia.org/wiki/Decision_tree_learning
- https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/#three
- https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
- http://www.pybloggers.com/2016/11/random-forests-in-python/
- https://archive.ics.uci.edu/ml/datasets/Car+Evaluation
- https://en.wikipedia.org/wiki/Pandas_%28software%29
- https://www.quora.com/What-do-you-think-of-the-Seaborn-package-in-Python
- https://en.wikipedia.org/wiki/Matplotlib
- https://en.wikipedia.org/wiki/Scikit-learn
- https://www.hackerearth.com/blog/machine-learning/simple-tutorial-svm-parameter-tuning-python-r/
- http://scikit-learn.org/stable/modules/svm.html
- http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd