Machine Learning

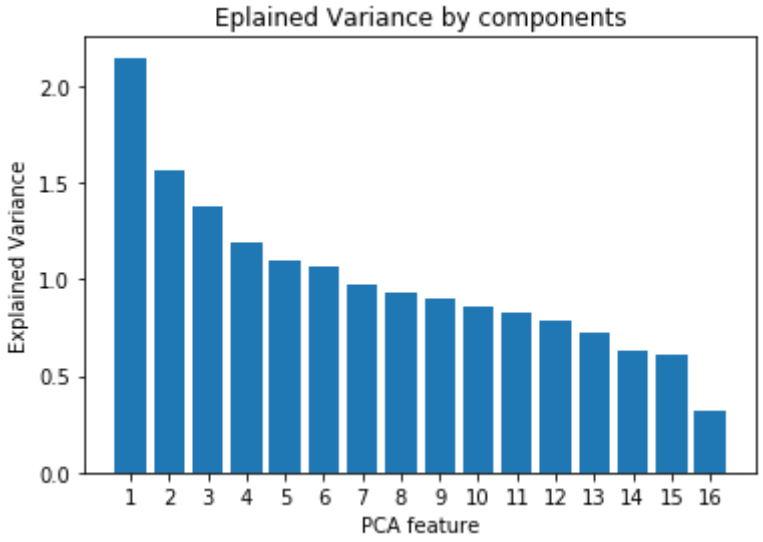
This section deals with building a predictive model for the client data for predicting whether or not the client subscribes for a term deposit.

It focuses majorly on:

- Principal component analysis (PCA)
- Upsampling the skewed data
- Base model selection from various models trained.
- Test metric
- Hyper-parameter tuning
- Predicting new data

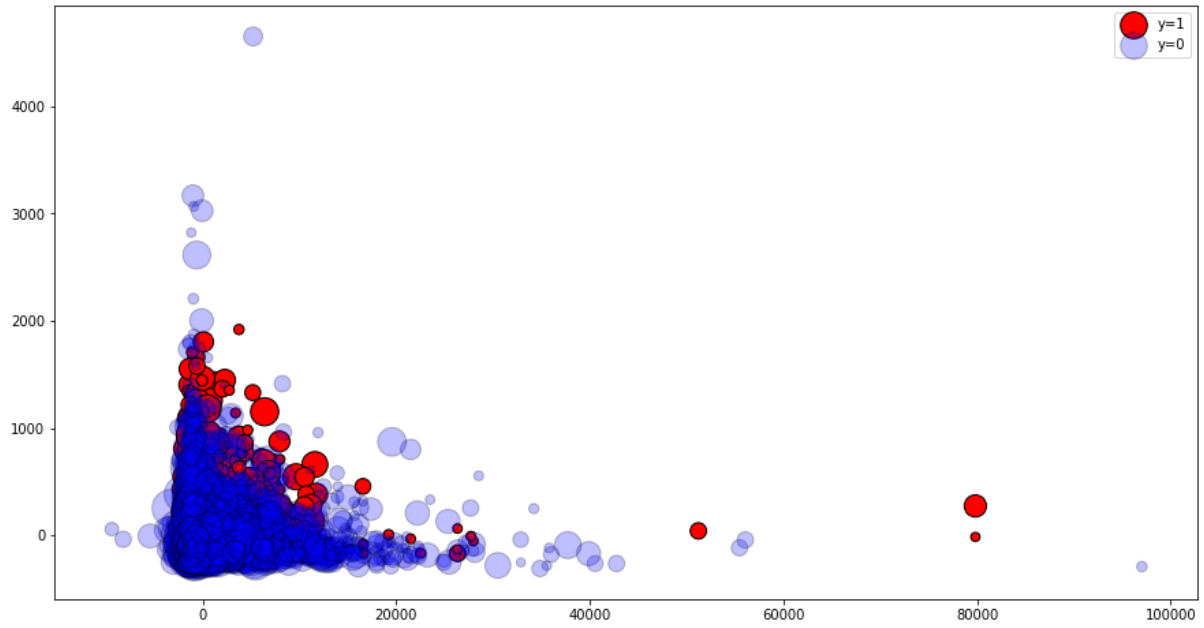
Principal component analysis (**PCA**)

Principal Component Analysis (PCA) is used to explain the variance-covariance structure of a set of variables through linear combinations. In this case, it is mainly done for two reasons. To understand the variance-covariance structure for the features and for visualization of the data that is multi-dimensional.



From the above barplot, the difference of explained variance is minimal after the number of features increases to 3. For that reason, feature reduction is done later after building the model during hyperparameter tuning.

PCA can also be used as a great visualization tool to get an idea about data we are about to predict. The scatter plot below shows how this data might look on a two-dimensional space.



Upsampling

As we can see from the visualization above that the data is highly skewed. For building a model for such skewed data, the minority class is upsampled by sampling from the existing data repetitively with repetition until the majority class and minority class have equal samples.
This is done using the package sklearn.utils.resample.

Base Model Selection from various models

The upsampled data is then used to build the following models:

- Linear regression
- Knn
- Support Vector Machine (SVM)
- Decision Tree
- Random Forest
- Extreme Gradient Boosting
- Gradient Boosting Classifier

The data is split into training and test data in the ratio of 4:1. The data is then scaled using the StandardScalar. Each model is trained using the training data and the model is then used to predict using the test data.

Test Metric

There are three test metrics computed on each of the models.

- Accuracy score
- Recall
- Mathew's correlation coefficient (MCC)

Mathew's correlation coefficient uses all the four components of a confusion matrix. It returns a value between -1 and 1 with -1 being a poorly fitted model and 1 being the rightly fitted model.

The entire data is split into training and testing datasets in the ratio of 4:1. The following models are trained using the training dataset and various test metrics are computed using the test dataset. The model is selected based majorly on MCC.

The MCC scores obtained after running various models on test data is as follows:

Models	MCC
Random Forest Classifier	0.941119
Decision Tree Classifier	0.927532
K-Near Neighbors	0.863852
Support Vector Machine	0.732137
Gradient Boosting	0.722849
XGBoost	0.718817
Logistic Model	0.607161

The Random Forest classifier performs much better than all the other models.

```
Training Accuracy score :  100.0
Confusion matrix :
 [[31995     0]
 [    0 31880]]
Recall - train :  1.0
MCC :  1.0
```

```
Testing Accuracy score :  97.0
Confusion matrix :
 [[7450  477]
 [   6 8036]]
Recall - test :  0.9992539169360856
MCC :  0.9411193633312515
```

Hyper-Parameter tuning

Hyper-Parameter is done on the base model selected to make sure that the model works well with the available data. Feature reduction is also done here.
From this step, it is understood that out of the 16 available features, it is best to use 6 features for this model.

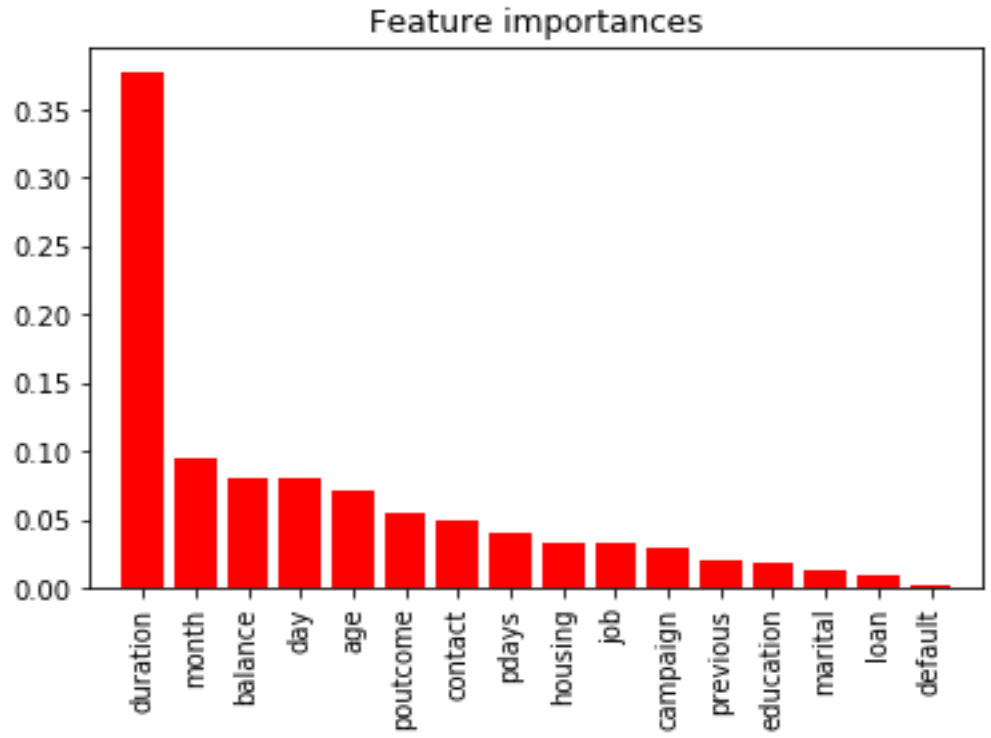
The test metric is computed again after hyperparameter tuning.

```
Training Accuracy score :  100.0
Confusion matrix :
 [[31995     0]
 [    0 31880]]
Recall - train :  1.0
MCC :  1.0

Testing Accuracy score :  97.0
Confusion matrix :
 [[7534  393]
 [  11 8031]]
Recall - test :  0.9985078338721711
MCC :  0.957430616160695
```

Feature importance

The features used in training the model have different importance associated. Few features are more important than the others.



Duration of the call made to the clients is very important in determining whether a client will subscribe to a term deposit or not. This is the most important feature of the entire data. This was discussed earlier in data storytelling section of this project.

Predicting using new data

The new data that is imported is again treated into categorical columns as the test data was treated earlier. StandardScalar is also applied to the new data to make the new data similar to the training data.
The Random forest classifier performs very well in predicting new data. Mathew's correlation coefficient is 0.95, which is as good as it works on test data.

```
Training Accuracy score :  0.9891616898916169
Confusion matrix :
 [[3952   48]
 [   1  520]]
MCC :  0.950000214141225
```

Conclusion

The Random Forest classifier has the highest performance with Mathew's correlation coefficient of 0.957 achieved with test data. Hyperparameter tuning of the model increased the MCC from 0.9411 to 0.9574. The best number of features to be used is also determined using the hyperparameter tuning. This model also has very low false negatives compared to other models.