Machine Learning Engineer Nanodegree

Contents

Capston Project

Project Name: Bank Marketing Prediction

Submitted By: Shashank Varshney

Project Overview

Domain Background

This project is related to the telemarketing. Telemarketing is a domain in which companies directly contact customers and try to sell the products which can help customer to achieve their goals.

In telemarketing, customer can be contacted by any means like e-mails, calls, etc. As part of telemarketing process, customer is contacted and provided information about a product and it can sometimes be required to contact customer more than one time.

The problem in telemarketing domain is company have to call all of the customers be it a potential customer or not. This become a lot of logistical work and lot of efforts are wasted on the customers which don't fit in the potential buyer category for that product.

This problem requires solution so that company can target only the potential customers which have higher probability to buy that product instead of wasting efforts on the customers which have lower probability of buying that product.

So past data can be used for the data analysis and ultimately machine learning techniques can be applied to predict the potential buyers as per various attributes related to customers.

This kind of problem has already been researched and following is the link of the same.

http://media.salford-systems.com/video/tutorial/2015/targeted_marketing.pdf

Problem Statement

This dataset has been taken from the UCI Machine learning repository. This data is related to direct marketing campaigns (phone calls) of Portuguese banking institution. The goal is to predict if client will subscribe a term deposit or not based on some attributes of the client so that potential clients can be targeted instead of wasting efforts on the clients which have very low probability of buying the term deposit.

Following steps will be used to solve this problem.

- Data will be collected.
- Data will be loaded.
- Various features will be analyzed to find if any strong relation between any of the feature and outcome or within feature themselves.

- Data will be pre-processed like numerical features will be transformed so that they should be on single scale and categorical features will be encoded to numerical by using one-hot encoding and feature engineering and feature selection will be performed.
- This is classification problem so various classification algorithms will be used like logistic regression, Support vector classifier, Random Forest Classifier, Ada-Boost Classifier, Gradient-Boost Classifier and K-Nearest Neigbours.
- Best classifier will be selected among mentioned classifiers on the basis of F score and various other parameters like training and testing times.
- Best classifier will be further tuned by tuning various hyper-parameters.
- Top features will be selected on the basis of feature importance and accuracy will be checked with and without feature selection and accordingly feature selection will be done.
- Stability of the classifier will be tested by using K-Fold.

Solution Statement

Solution of this problem is to create a model which can accurately predict whether client is going to subscribe to term deposit or not. Accurately predicting it can help the organization to put efforts towards the potential clients which are going to subscribe the term deposit instead of calling each client.

So, a predictive model should be created which can accurately predict whether client is going subscribe to term deposit or not.

Analysis

Data Exploration

Dataset contains information related to direct marketing campaigns of Portuguese banking institution. The marketing campaigns were based on phone calls and often more than one contact to the same client was required, to access if the product would be yes or no for the subscription.

Dataset contains 41188 examples and 20 input variables. Following are the input variables.

1. **Age:** Age is the factor which can impact client interest in the term deposit.
2. **Job:** This is type of job client have.
3. **Marital:** This is marital status of the client.
4. **Education:** This gives educational background of the client.
5. **Default:** Whether client has credit in default.
6. **Housing:** Client has housing loan or not.
7. **Lona:** Client has personal loan or not.
8. **Contact:** contact communication type. Cellular or telephone.
9. **Month:** Last contact month of the year.
10. **Day_of_week:** Last contact day of the week.

11. **Duration:** last contact duration in seconds. This attribute highly affects the output target (if duration = 0 then y = “no”). This input will only be included for the benchmarking purposes and will be discarded for predictive modelling.
12. **Campaign:** Number of contacts performed during this campaign for this client including last contact.
13. **Pdays:** Number of days that passed by after the client was last contacted from a previous campaign. 999 means client was not previously contacted.
14. **Previous:** number of contacts performed before this campaign and for this client.
15. **Poutcome:** Outcome of the previous marketing campaign.
16. **Emp.var.rate:** Employment variation rate - quarterly indicator.
17. **Cons.price.idx:** Consumer price index - monthly indicator.
18. **Cons.conf.idx:** Consumer confidence index - monthly indicator.
19. **Euribor3m:** Euribor 3 month rate - daily indicator.
20. **Nr.employed:** Number of employees - quarterly indicator.
21. **Y:** Output variable. Has client subscribed to term deposit or not?

All the above mentioned will be used for analysis and the prediction model building except the “Duration”.

- Data set contains total 10 numeric features and 10 categorical features.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
age               41188 non-null int64
job               41188 non-null object
marital           41188 non-null object
education         41188 non-null object
default           41188 non-null object
housing           41188 non-null object
loan              41188 non-null object
contact           41188 non-null object
month             41188 non-null object
day_of_week       41188 non-null object
duration          41188 non-null int64
campaign          41188 non-null int64
pdays             41188 non-null int64
previous          41188 non-null int64
poutcome          41188 non-null object
emp.var.rate      41188 non-null float64
cons.price.idx    41188 non-null float64
cons.conf.idx     41188 non-null float64
euribor3m         41188 non-null float64
nr.employed       41188 non-null float64
y                 41188 non-null object
dtypes: float64(5), int64(5), object(11)
memory usage: 6.6+ MB
```

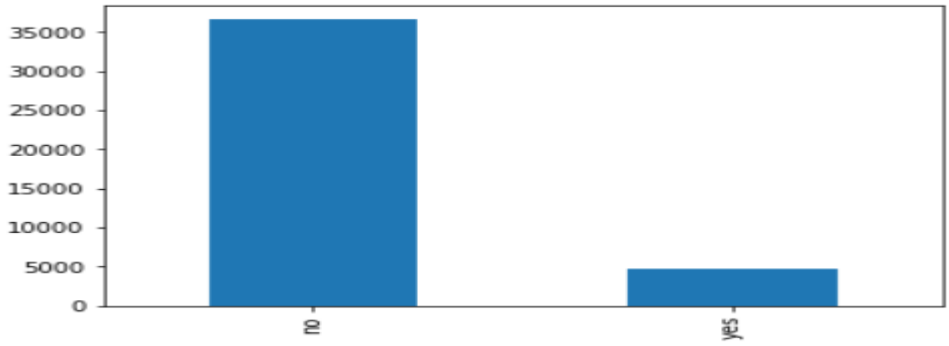
- Following are the stats of various numeric features. This denotes some outliers in various features which we will further check.

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000
mean	40.02406	258.285010	2.567593	962.475454	0.172963	0.081886	93.575664	-40.502600	3.621291	5167.035911
std	10.42125	259.279249	2.770014	186.910907	0.494901	1.570960	0.578840	4.628198	1.734447	72.251528
min	17.00000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.634000	4963.600000
25%	32.00000	102.000000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.344000	5099.100000
50%	38.00000	180.000000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000	5191.000000
75%	47.00000	319.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000	5228.100000
max	98.00000	4918.000000	56.000000	999.000000	7.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000

- Following is the dataset's first 5 observations. This gives basic feel of the dataset like how dataset is structured.

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	261	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
1	57	services	married	high.school	unknown	no	no	telephone	may	mon	149	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
2	37	services	married	high.school	no	yes	no	telephone	may	mon	226	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
3	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	151	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
4	56	services	married	high.school	no	no	yes	telephone	may	mon	307	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no

- This is the distribution of outcome variable and we can clearly see outcome is skewed towards "no" as compared to "yes".

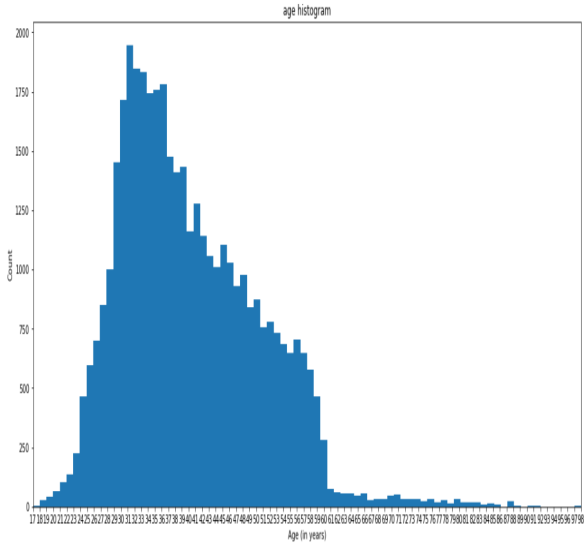


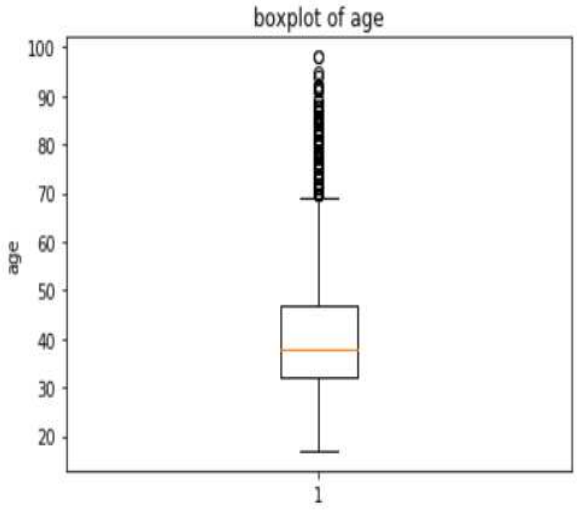
Exploratory visualization

In this section we will perform exploratory analysis of various features in the data set.

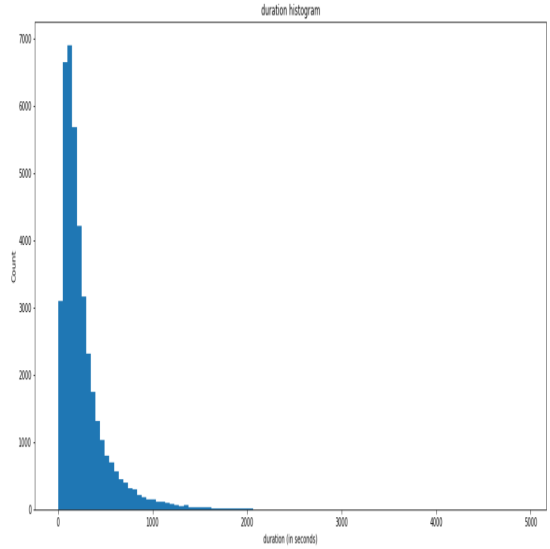
Numeric Features

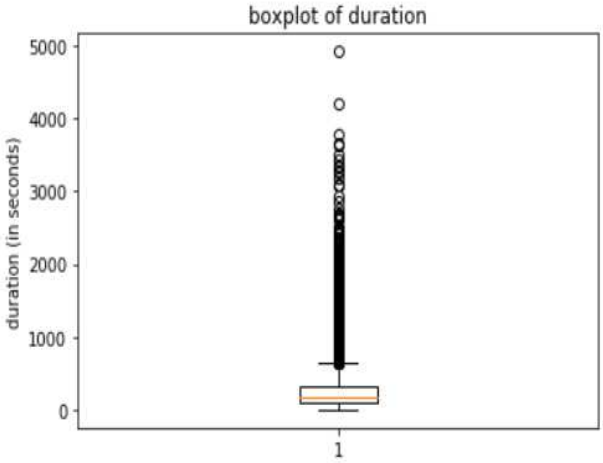
- Age is spread from 17 years to 98 years and there are lot of outliers as well. Following are the histogram and boxplot of age.



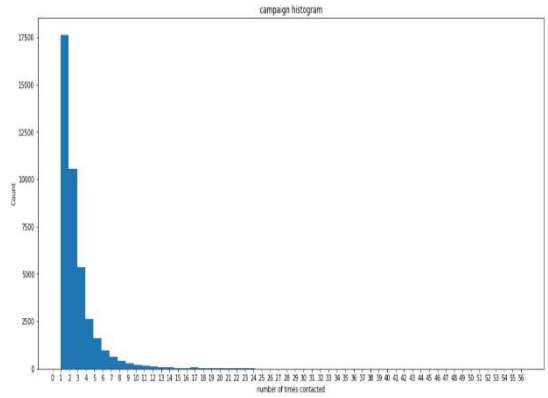


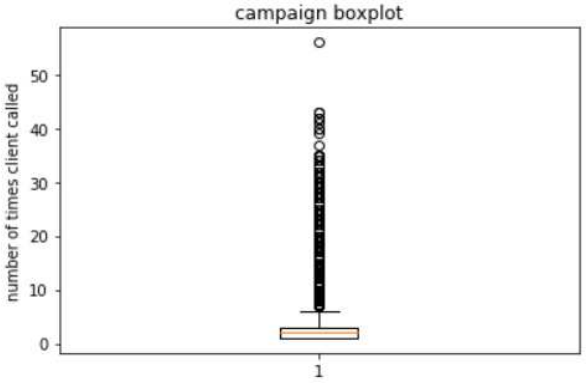
- Duration feature is also skewed, and lot of outliers are there.





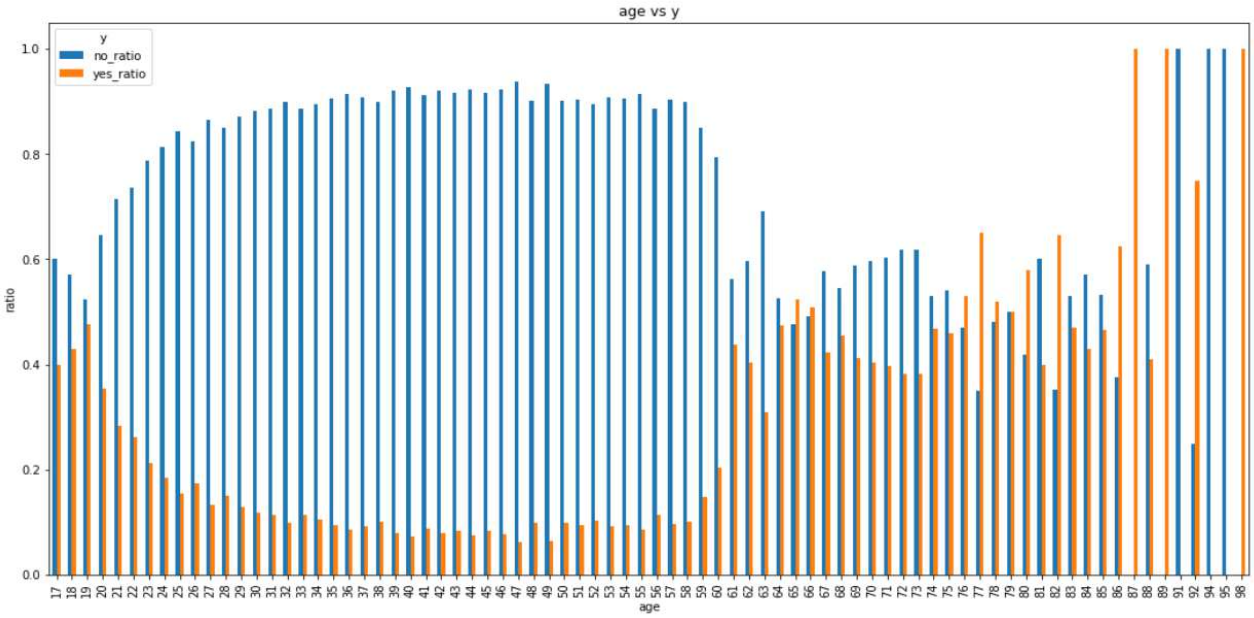
- Campaign feature is also showing the skewness and outliers.



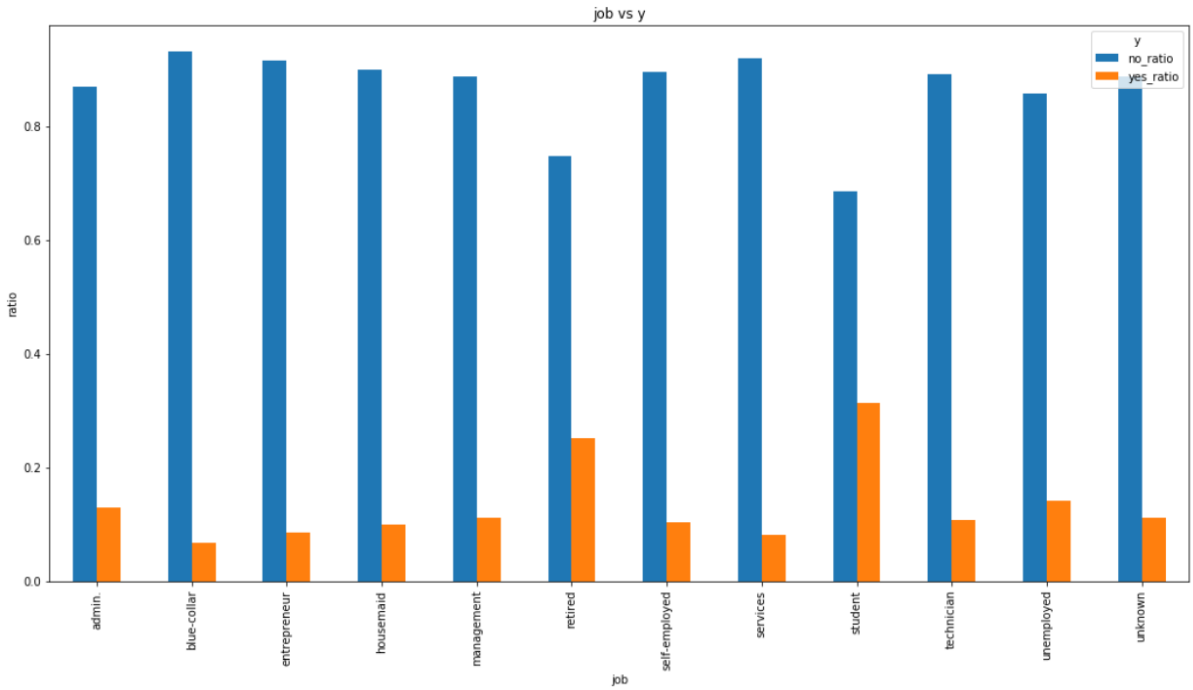


Feature value count to success ratio exploratory analysis

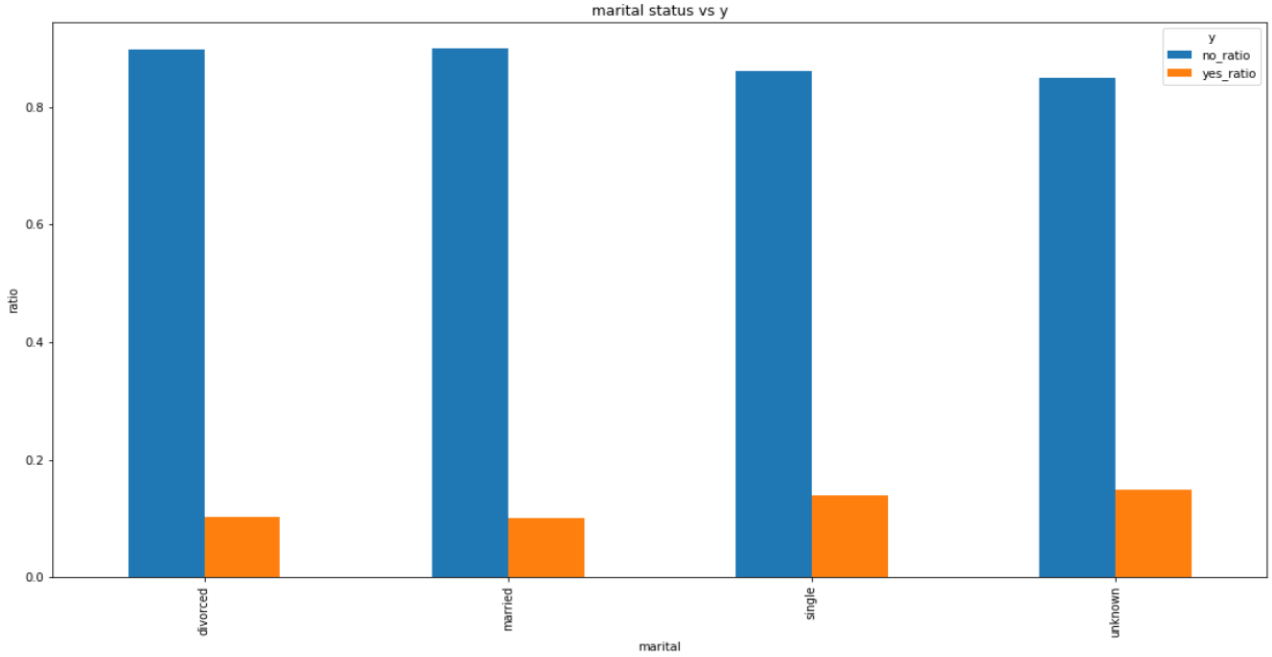
- Age vs y ratio: We can see from the above plot that success ratio is higher for the clients with age below 25 and above 60. In the mid age group success ratio is less.



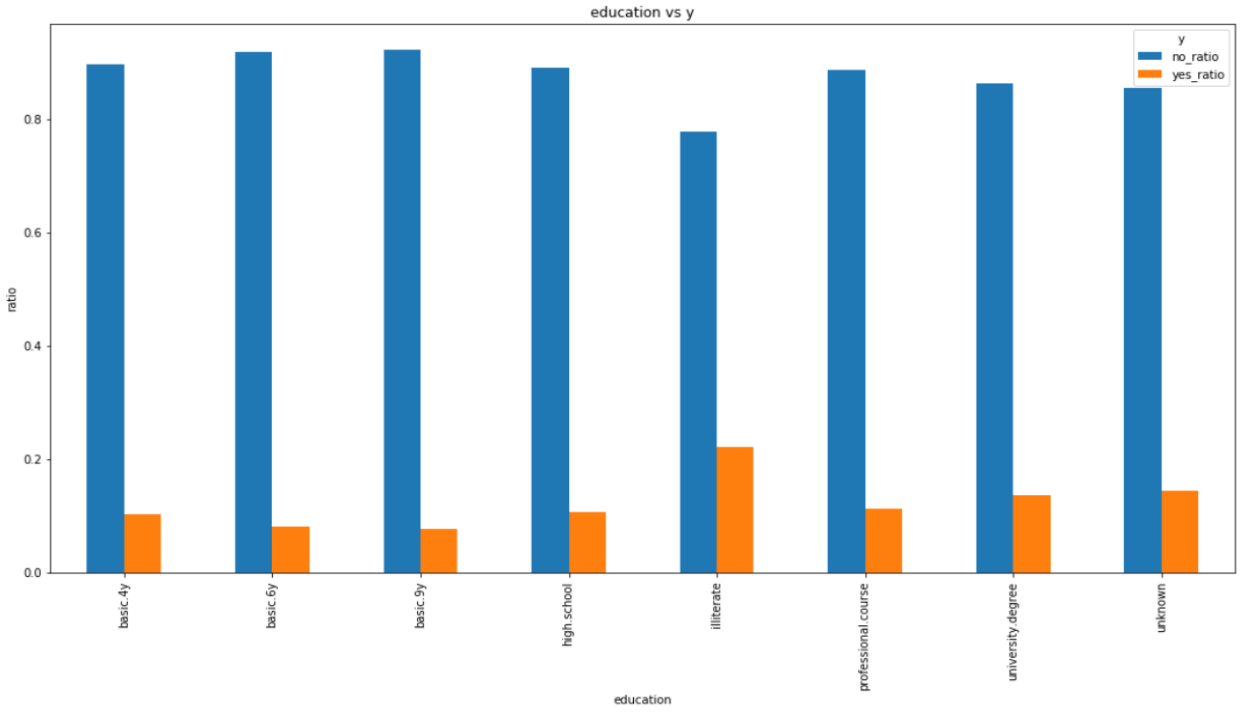
- Job type vs y ratio: Success ratio among retired and students are higher as compared to other professions.



- Marital status vs job: Not very high difference amonth the success ratios of different relationship types but it seems success ratio of Singles and unknowns are comparatively high.



- Education vs y ratio: Success ratio is higher for illiterate, but sample size is very small. There is not strong correlation between success and client education, but it seems higher the education better are the chances for success.



- Further all other variables were also analysed but not very high correlation was found among the variables and outcome variable y.

Algorithms and Techniques

This is a classification problem in which we need to identify whether client will opt for term deposit or not based on various other features.

We will use following 6 classification methodology and also brief of all of these techniques is also given.

- **Logistic regression:** Logistic regression training is all about the finding the relationship between independent variables or features to the class. Logistic regression tries to find which class is most likely to occur for the given values of features. This is done by finding the probability.

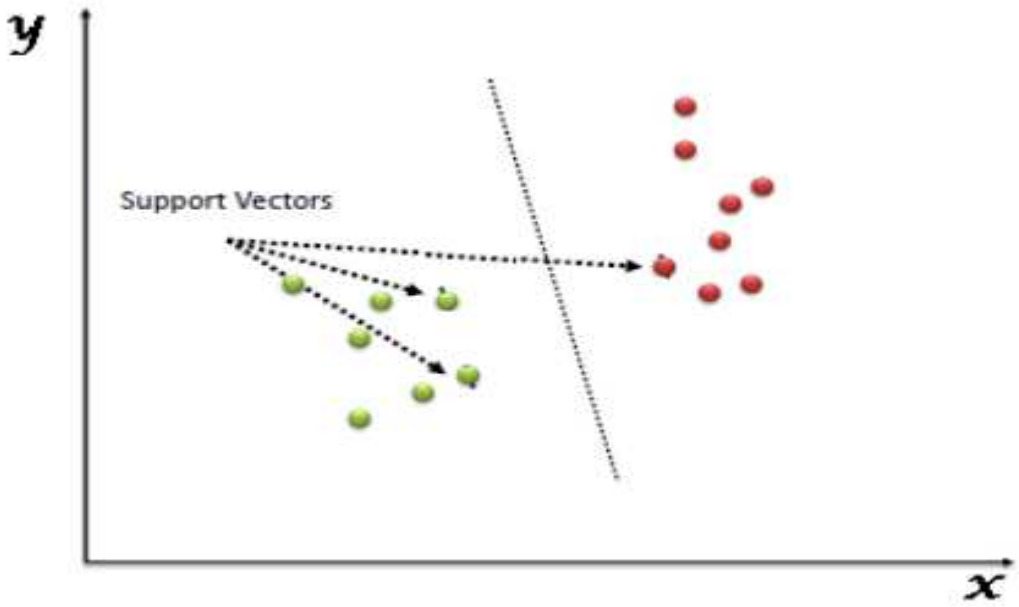
  In our example we have 2 classes, that is, individual will opt for term insurance or not. So, these are 2 classes. We have various independent variables or features like age, education, job, etc. Some of the features affect outcome more and some of the features affect income less.

  While training for the logistic regression, we tries to establish a relationship between all features and outcome to find the probability whether probability of outcome as yes is higher or no is higher.

  This relationship gives us a model or a mathematical relationship which calculates the probability of outcome as per given values of feature variables.

  Usually logistic regression is used for binary classification which means output variable contains only 2 classes like "yes/no", "true/false", etc but that can be extended to use logistic regression to classify data to more than 2 classes.
- **K-Nearest Neighbor:** In this technique outcome is found based on the nearest neighbors of the data points. K is the number of neighbors and outcome will be decided based on the number of neighbors belong to that class. Let's say any data point have more nearest neighbors belong to class "no" then outcome will be made as "no".
- **Random Forest classifier:** Random Forest is a supervised learning algorithm. Like you can already see from its name, it creates a forest and makes it somehow random. The "forest" it builds, is an ensemble of Decision Trees, most of the time trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result.

  Random forest method builds multiple decision trees and merges them together to get a more accurate and stable prediction.
- **Support Vector Classifier:** "Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).



Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

- **Ada-boost:** It is also like Random Forest Classifier is another ensemble classifier. (Ensemble classifier are made up of multiple classifier algorithms and whose output is combined result of output of those classifier algorithms).
  Ada-boost classifier combines weak classifier algorithm to form strong classifier. A single algorithm may classify the objects poorly. But if we combine multiple classifiers with selection of training set at every iteration and assigning right amount of weight in final voting, we can have good accuracy score for overall classifier.
- **Gradient Boosting Classifier:** Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.
  The objective of any supervised learning algorithm is to define a loss function and minimize it.
  We want our predictions, such that our loss function (MSE) is minimum. By using gradient descent and updating our predictions based on a learning rate, we can find the values where MSE is minimum.
  So, we are basically updating the predictions such that the sum of our residuals is close to 0 (or minimum) and predicted values are sufficiently close to actual values.

Benchmark Model

Simple Naive predictor will be used as the benchmark model which will consider that every client is going to subscribe the term deposit because this is how current process of the organization is working. Currently organization is calling every client.

Predictive model which will be used as a solution should have way higher accuracy than the benchmark simple Naive predictor.

Evaluation Metrics

Counts of clients who said “yes” is 4640 and those who said “no” is 36548. So this is clearly an unbalanced distribution. If we consider all “yes” which is usually the case and call to every client then we will get 11.27% of accuracy and same is the F1 score when beta = 0.

F-beta score and accuracy shall be used as evaluation metrics for the predictive model and same will be compared with the benchmark model. Following is the formula of the F-beta score.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

Accuracy measures how often the classifier makes the correct prediction. It’s the ratio of the number of correct predictions to the total number of predictions (the number of test data points).

Precision tells us what proportion of clients we classified as “yes” were actually “yes”.. It is a ratio of true positives (clients which classified as “yes”, and which are actually “yes”) to all positives (all clients classified as “yes”), in other words it is the ratio of

[True Positives/(True Positives + False Positives)]

Recall(sensitivity) tells us what proportion of clients are “yes” were classified as “yes”. It is a ratio of true positives (clients classified as “yes”, and which are actually “yes”) to all the words that were actually spam, in other words it is the ratio of

[True Positives/(True Positives + False Negatives)]

We don’t want to miss any client which can say “yes” so we would like to focus more on the Recall. So I would like to keep value of beta as 0 so that recall can be focused.

If we will go with beta = 0 then we will be working on accuracy which is same as F score in our scenario.

Methodology

Data Pre-processing

Algorithm which depends on distance based requires data transformation so that feature value shouldn’t be different from other feature value otherwise that feature will weigh more in that algorithm and will overshadow that other features.

This also increase training speed of the algorithm.

I have used Min-Max transformation on the numerical features which changes the values of the feature to the values between 0 to 1.

Following is the formula used for the same.

$$x' := (x - x_{\min})/(x_{\max} - x_{\min})$$

Also, most of the algorithm accepts the data in numerical format. So categorical features need to be converted to numerical values.

I have used One-Hot encoding technique for the same.

One-hot encoding creates a "dummy" variable for each possible category of each non-numeric feature. For example, assume someFeature has three possible entries: A, B, or C. We then encode this feature into someFeature_A, someFeature_B and someFeature_C.

	someFeature		someFeature_A	someFeature_B	someFeature_C
0	B		0	1	0
1	C	----> one-hot encode ---->	0	0	1
2	A		1	0	0

I have created one more feature for age_stage based on age. In this I have created 4 age stages as given below.

- teenage : age < 20
- young : age > 19 & age < 41
- mid_age : age > 40 & age < 61
- old : age > 60

Implementation

Following steps were used to implement the various algorithms.

- Data set was broken into 3 parts:
  - Training set
  - Validation set
  - Test set
- Training set will contain 60% part of the data set and will be used to train the data set.
- Validation set will be used for validation of model on unseen data and to visualize if model is overfitting or underfitting.
- Test set was used for further testing on the data and is used as a data for which outcome is not known.
- All 6-mentioned classification will be fitted on 10%, 50% and 100% training set.
- Then fitted model will be validated on model set and finally used for predicting on test set.

- Accuracy, training and testing times and F-beta score with beta = 0.001 is used because sklearn fscore metrics doesn't accept beta = 0.
- Then finally algorithm with highest F-score was selected.

Difficulty during implementation and possible complications

- Accuracy, training and testing times and F-beta score with beta = 0.001 is used because sklearn fscore metrics doesn't accept beta = 0.
- Function "train_predict" and "evaluation_graphs" are the core of the process where multiple classification algorithms will be tried for training and predicting. Both of these functions are related like output of "train_predict" will be used as input for "evaluation_graphs". If anything changes in one function can impact functionality of other function. So, one must be careful while changing these functions.
- During coding, algorithm was planned but while coding lot of too and for changes were done in the mentioned 2 functions to make it work.
- Any changes in the above mentioned functions can bring whole system on halt.

Refinement

From the below graphs we can clearly see Ada-boost performed best on validation and testing set.

Ada-boost also very less training and prediction time as compared to SVC which is high training time and KNN which has very high predicting time.

Further Ada-boost was refined more by using grid search. Grid search was done on "n-estimators" and "learning_rate" parameters of the Ada-boost classifier and F1-score was further improved.

Following values for “n_estimators” and “learning_rate” were used in grid search.

parameters = {'n_estimators': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 200],

'learning_rate': [0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 0.75, 1, 5, 10]}

Following shows the same.

```
Unoptimized model
------
Accuracy score on validation data: 0.8990
F-score on validation data: 0.6733
Accuracy score on test data: 0.8997
F-score on test data: 0.6525

Optimized Model
------
Final accuracy score on the validation data: 0.8992
Final F-score on the validation data: 0.7208
Final accuracy score on the test data: 0.8996
Final F-score on the test data: 0.6840
```

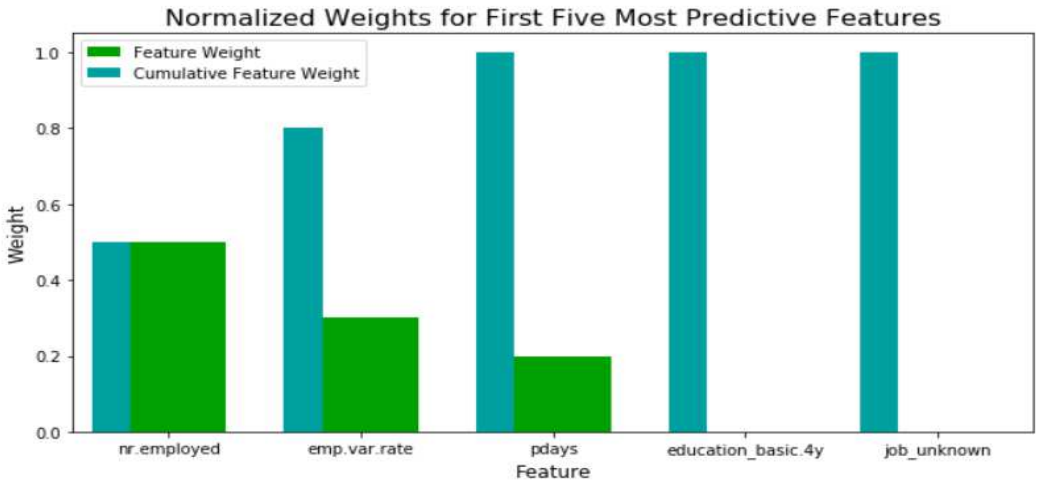
Following is the n_estimators and learning_rate gave the best result for Adaboost.

```
AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None,
          learning_rate=0.2, n_estimators=10, random_state=1
```

Feature Selection

As per feature importance of best Adaboost classifier, we have top 3 features which are covering 100% weightage. So, we just need to use only top 3 features. This will reduce the training and prediction time significantly.

Following is the graph of feature importance.



Further accuracy and f-score on reduced data set (only top 3 features) and full data set were tested, and both found because top 3 features are having 100% weightage.

```
Final Model trained on full data
------
Accuracy on testing data: 0.8992
F-score on testing data: 0.7208
Accuracy on testing data: 0.8996
F-score on testing data: 0.6840

Final Model trained on reduced data
------
Accuracy on testing data: 0.8992
F-score on testing data: 0.7208
Accuracy on testing data: 0.8996
F-score on testing data: 0.6840
```

Results

Model Evaluation and Validation

We can clearly see final model has lot of improvements as compared to benchmark Naïve Naive Model.

Naïve model has fscore is 0.112654271081 and accuracy is 0.112654171118.

Our final model has following accuracies and F1 score.

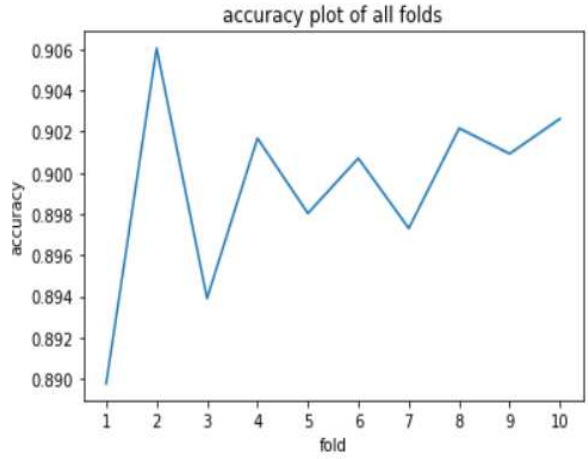
```
Unoptimized model
------
Accuracy score on validation data: 0.8990
F-score on validation data: 0.6733
Accuracy score on test data: 0.8997
F-score on test data: 0.6525

Optimized Model
------
Final accuracy score on the validation data: 0.8992
Final F-score on the validation data: 0.7208
Final accuracy score on the test data: 0.8996
Final F-score on the test data: 0.6840
```

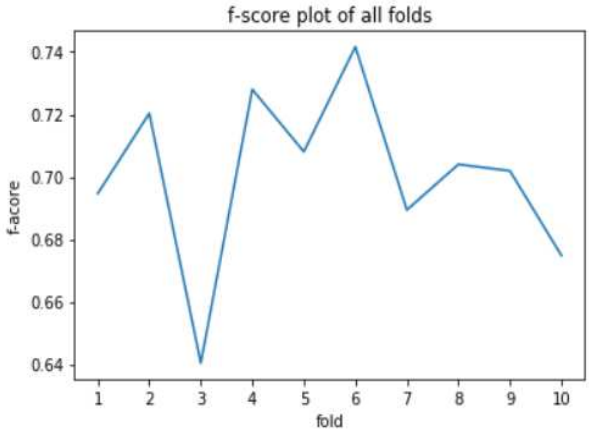
We can clearly see not very high difference has been observed among Validation data and test data accuracy and F score.

Further k-fold was used with K=10 to check if any high deviation in accuracy and f-score is observed for different folds but same couldn't be found. Mean and Standard deviation is very near to values given by our optimized model.

```
mean of accuracy is 0.899315453882 and standard deviation of accuracy i
s 0.00447535553146
mean of f-score is 0.700313644539 and standard deviation of f-score is
0.0271130901026
```

Following are the graphs showing the same.





So this proves that model generalized well and result from the model can be trusted.

Justification

We can clearly see final model has lot of improvements as compared to benchmark Naïve Model.

Naïve model has fscore is 0.112654271081 and accuracy is 0.112654171118.

Our final model has following accuracies and F1 score.

```
Unoptimized model
------
Accuracy score on validation data: 0.8990
F-score on validation data: 0.6733
Accuracy score on test data: 0.8997
F-score on test data: 0.6525

Optimized Model
------
Final accuracy score on the validation data: 0.8992
Final F-score on the validation data: 0.7208
Final accuracy score on the test data: 0.8996
Final F-score on the test data: 0.6840
```

Final model has significantly solved the problem because earlier F-score was very low and telemarketing team needs to call each customer. Now with this model, outcome can be predicted and only potential customers who have high probability of buying the term deposit will be called which will surely improve the productivity of the team.

Conclusion

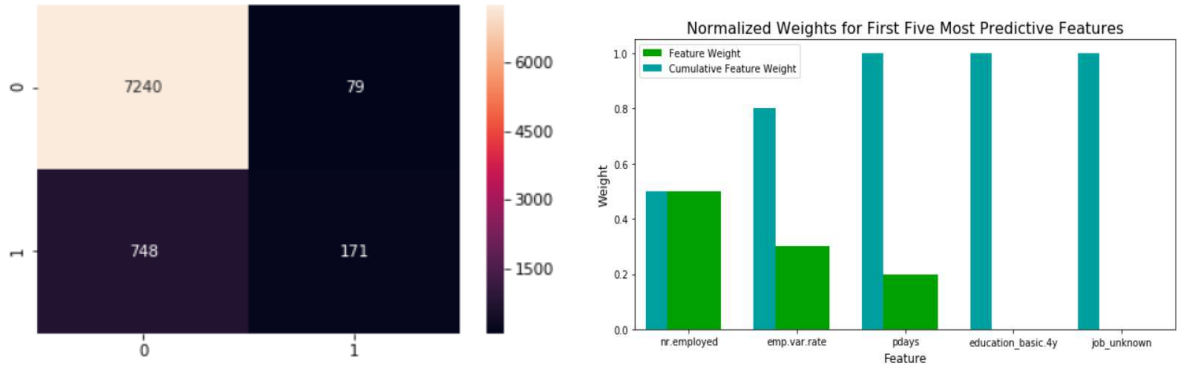
Free-form visualization

As discussed during the exploratory analysis, outcome is not conclusively related to any of the feature accept the duration feature which we didn't take into consideration because

that shouldn’t be taken into consideration as duration will be known only after calling to customer and if duration = 0 then surely outcome will be “no”.

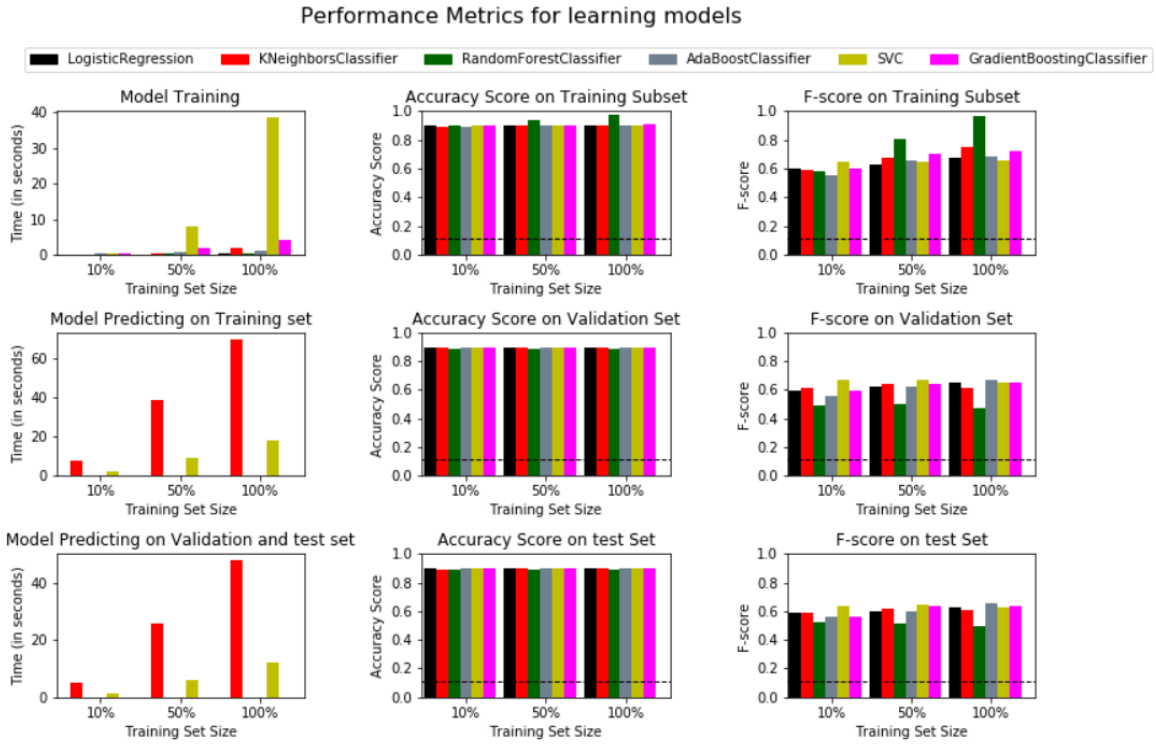
So, nothing conclusive very high dependency on any of the feature is found which could impact the outcome significantly so all features were required to be used to train the model.

We have following confusion matrix on test set and feature importance plot.

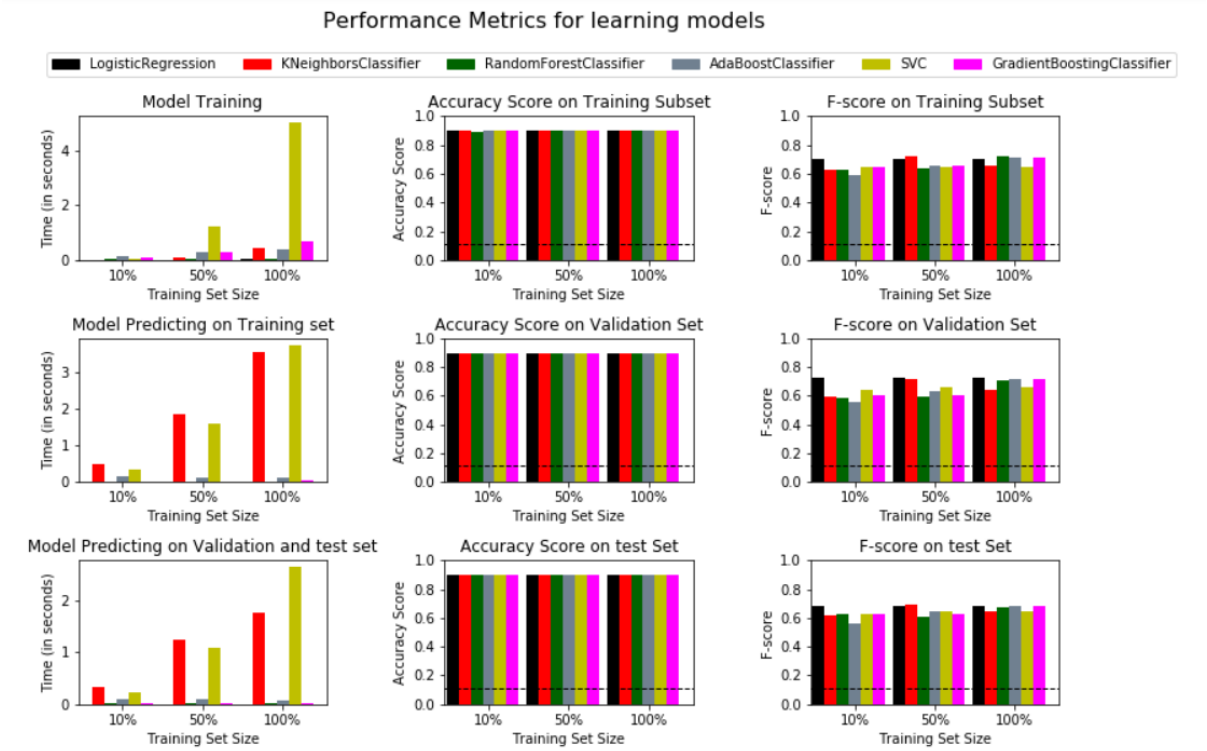


Time required for training and predicting was checked for reduced features data sets as well and it was found that time has reduced drastically with feature selection.

Graphs before feature selection on full data set.



Graphs after feature selection on reduced data set.



Reflection

Following is the summary of the entire project:

- Data was acquired and loaded in the python.
- Exploratory data analysis was done, and features were analyzed thoroughly.
- Data pre-processing and feature engineering was also done.
- Multiple models were tried, and accuracy and F score was checked on the validation and test set.
- Finally, best model among tried models was selected.
- Grid search was used to further improve the model by selecting the best hyper-parameter.
- Features were reduced by selecting top features and accuracy was checked on reduced feature models.
- K-fold cross-validation was used to check the model consistency and it was found that model generalized well.

Difficult part of this project was feature selection as features were not very strongly impacting the outcome.

Final model solves the problem and can be used for these kinds of issue provided features and distribution of features remains same. We can't fit same model to another problem be it similar type of issue or not.

Improvements

There is always scope of improving a model and same is true with final model as well. Following are some of the ways which can further improve the model:

- Feature engineering is one aspect which can be used to generate new features and further improve this model.
- Feature selection is another aspect which can be used to further improve the model after feature engineering.
- Hyper-parameter and base_estimator related changes in Ada-boost can also be used to further improve this model.
- After feature reduction, it was found that logistic regression was also performing good. Logistic regression can also be tried on reduced data set.

References

UCI Machine Learning repository

Google searches

Udacity courses Visualization and model fitting techniques and functions used in Nano Degree program.