Analysis and Classification of Adult dataset

Pavani Komati | CSE 5243 | 15rd Mar'18

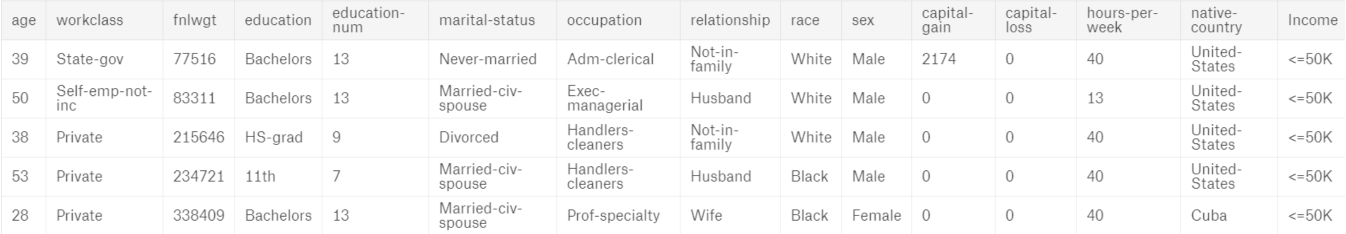
Table of Contents

1.INTRODUCTION

This report aims at processing and analyzing the adult data set from UCI database. The report walks through the preliminary data analysis, useful trends in the data, extracting useful features and building the model. Classification of the dataset is performed using multiple off-the-shelf classifiers. The results of all the classifiers are collected and compared to determine the best classifier for this model.

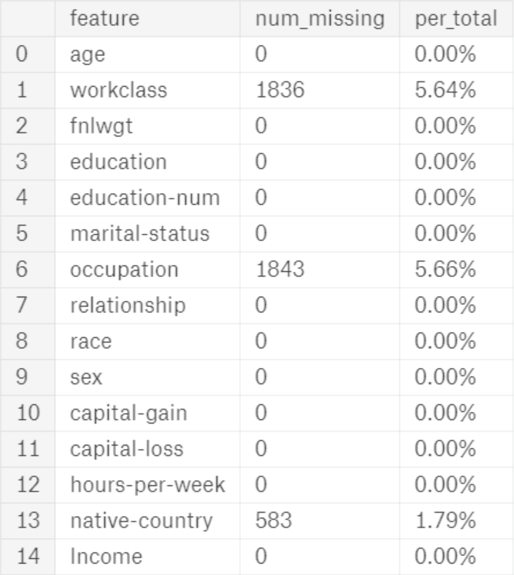
2.PRELIMINARY DATA ANALYSIS

The dataset consists of 32561 records with 14 features. In these features fnlwgt, education-num, capital-gain, capital-loss, hours-per-week are numerical attributes and the rest are categorical attributes.



age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	Income
39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

MISSING VALUES



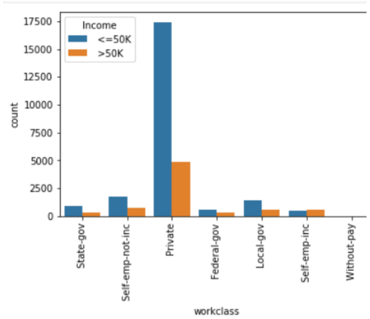
	feature	num_missing	per_total
0	age	0	0.00%
1	workclass	1836	5.64%
2	fnlwgt	0	0.00%
3	education	0	0.00%
4	education-num	0	0.00%
5	marital-status	0	0.00%
6	occupation	1843	5.66%
7	relationship	0	0.00%
8	race	0	0.00%
9	sex	0	0.00%
10	capital-gain	0	0.00%
11	capital-loss	0	0.00%
12	hours-per-week	0	0.00%
13	native-country	583	1.79%
14	Income	0	0.00%

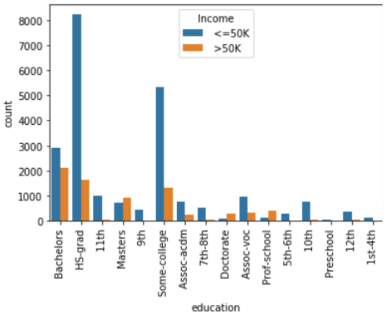
The given dataset consists of missing values represented as “?”. The current table shows the number and percentage of these missing features in each feature.

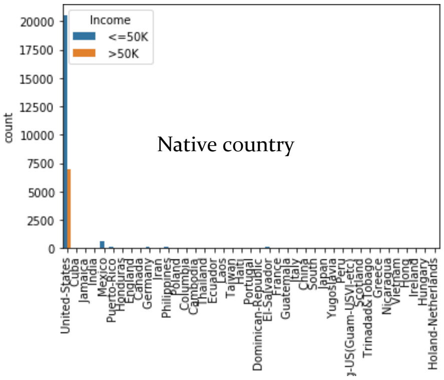
We can observe that, only three of the features i.e, workclass, occupation, and native-country consist missing values. In each of these features there are around 5% of missing values. Since these missing values covers small percentage of the dataset, we can eliminate these records to build an efficient model. After eliminating the records, the current dataset consists of 30,162 records. Around 7% of the total data is eliminated as missing records, which is corresponds to smaller number of records considering the huge dataset.

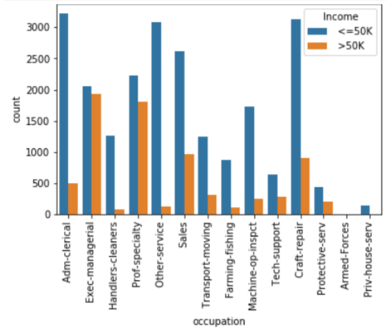
CATEGORICAL FEATURES

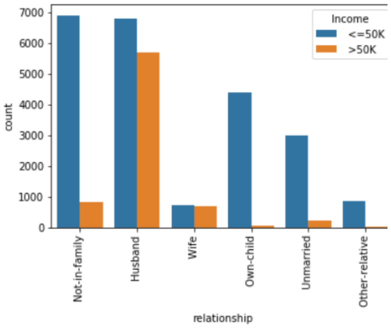
The distribution of various categorical features into multiple categories and their corresponding class label can be seen from the count plots for each feature.

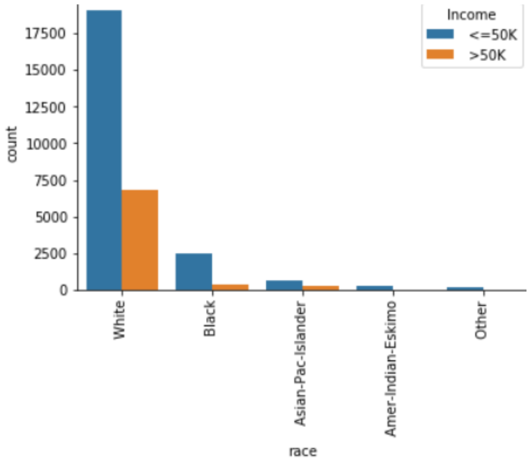


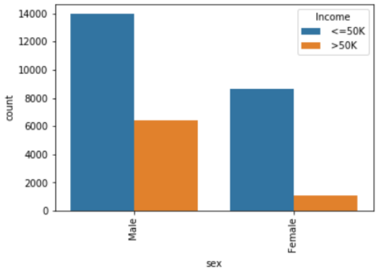


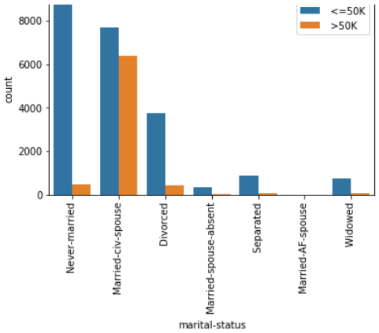








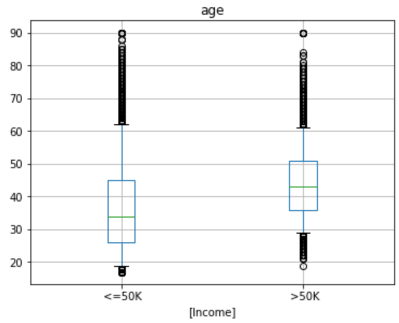


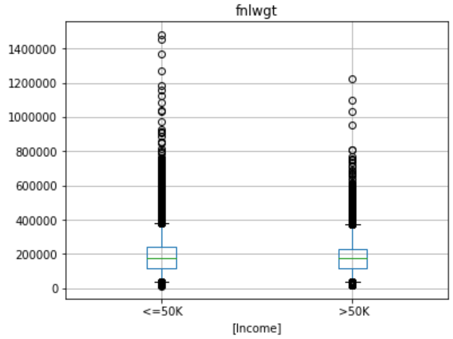


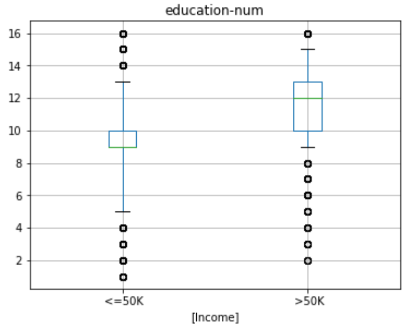
From these count plots charts, we can observe few prominent things.

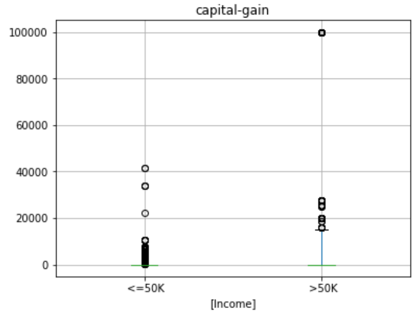
1. Majority of the race in "race" feature is white.
2. "Native-country" column has majority of the countries as United-States and very less percentage of other countries.

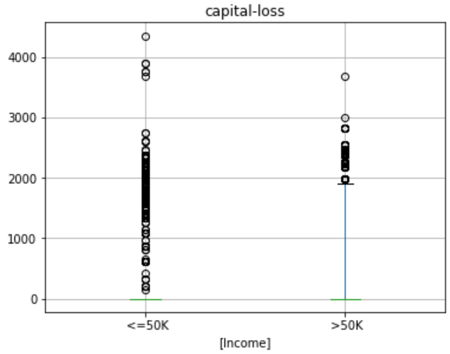
RELATION BETWEEN NUMERICAL FEATURES

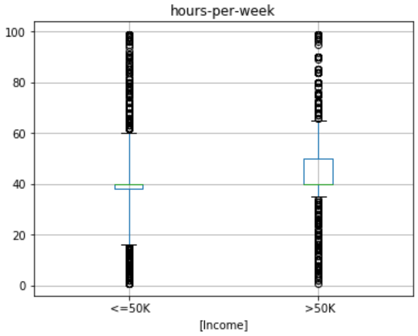




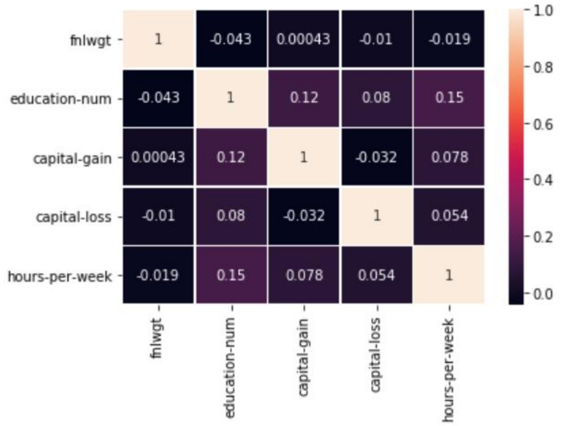






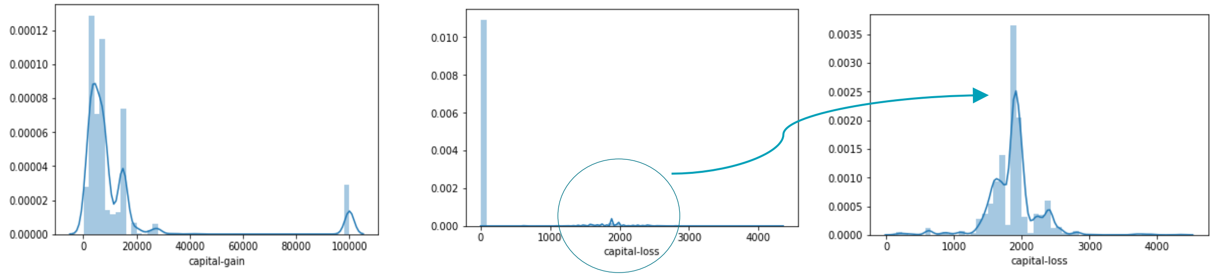


Correlation map

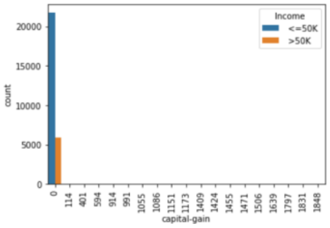


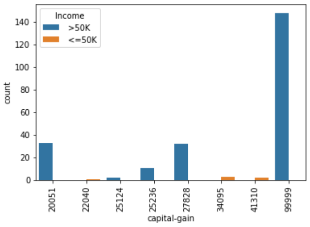
1. The correlation map between the numerical attributes shows that there is 0 to no correlation amongst these features.
2. The box plots show that, Most of the records with "education-num" >10 fall in the income category >=50k
3. "hours-per-week">40 fall into the >=50k income category.
4. "age" feature also shows a good amount of distinction indicating higher aged people fall into higher income category.
5. The boxplots show that capital-gain and capital-loss have most of the data as outliers.

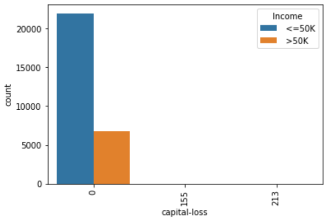
From the above observations, "education-num", "age" are "hours-per-week" could be the main deciding factors for the model building and classification.

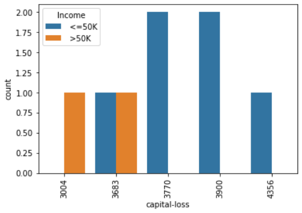


From the distribution plot, the majority of the values for capital gain and capital are distributed close to the 0 value. This is the reason why the above boxplots show us too many outliers for these features. These values cannot be considered as noise. Hence, they cannot be eliminated as outliers.
From the count plots for capital-gain it is observed that majority of the values corresponding to income class <=50 relate with capital-gain =0 and 99.5% of the values >25k correspond to income class >50k.
Similarly, for capital-loss majority of the values correspond to income class >=50k

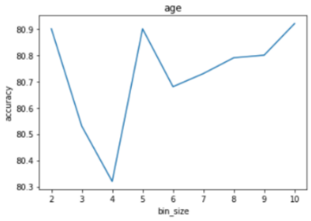






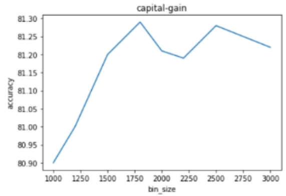


3.DATA TRANSFORMATIONS



Some data transformations like binning are performed on few of the numerical attributes and accuracy is compared with a default Random Forest classifier to check if the performance has improved or not. We have seen that the "AGE" feature is distributed evenly amongst all ages. In general, age as a range can be better factor for determining than taking a single value. Hence this category can be binned. To decide the binning value, various bin sizes are taken, and accuracy is predicted for various binning values. Binning value of 10 gave a better performance than the rest. The accuracy of the model is improved by 0.3%. Though the accuracies did not vary much, its practically a good idea to bin.

"CAPITAL-GAIN"

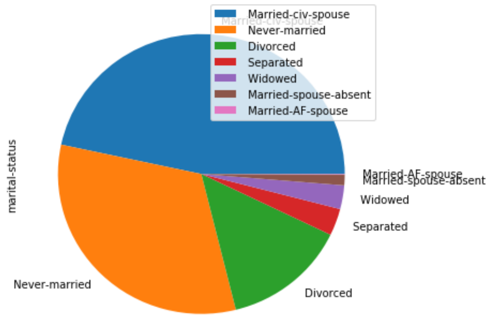


Similar to age factor, binning can be performed on this factor also. But, as the distribution of this data is not normally distributed, values above 20,000 (from the distribution plot) can be considered as one group and values below 20,000 are considered as another group and binned separately. By checking various bin sizes, bin size of 2000 for values less than 20,000 gave better performance (81.28%) compared to other sizes which varied by 0.1-0.2 %

"CAPITAL -LOSS" AND "HOURS-PER-WEEK"

Similar to the "capital-gain" feature, binning is performed by considering different bin sizes and accuracy is measured. Bin size of 500 gave the highest accuracy with 81.30% for capital-loss and a bin value of 18 gave an accuracy of 81.323% for hours-per-week feature.

“MARITAL-STATUS”



The marital-status feature is categorized into various sectors which seem to be unnecessary. This field can be simplified to contain lesser number of categories. But, the “relationship” feature gives the same information as the “marital-status”(i.e If a person is a wife, husband or son etc.,). Hence, I wish to drop this column in the data transformations section.

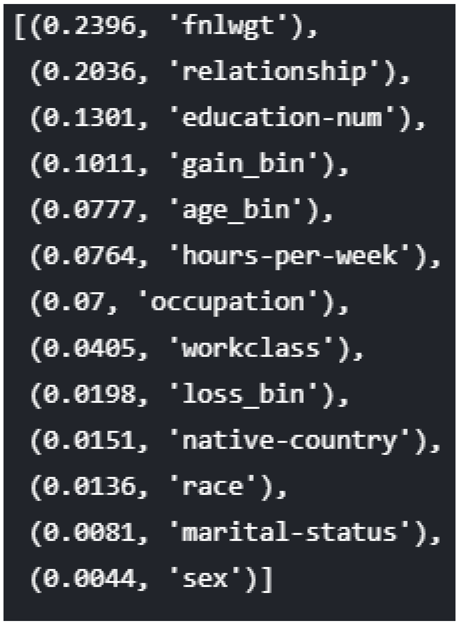
4.FEATURE SELECTION

“EDUCATION-LEVEL”

In the given dataset, “education-num” is the numerical value associated with the “education-level”. As they both give the same information, this column can be dropped.

“NATIVE COUNTRY” AND “RACE”

From the preliminary analysis, we have observed that, majority of the values in “native-country” belong to Unites-States and white “race”. Also, these features somewhat represent the same amount of feature. We can either select either of these features or none. But, we cannot decide this just by the distribution of these features because, they might contribute to the classification of some records accurately. Hence, the importance of features is measured by using Gini-Index with the Random Forest Classifier.



```
[(0.2396, 'fnlwgt'),
 (0.2036, 'relationship'),
 (0.1301, 'education-num'),
 (0.1011, 'gain_bin'),
 (0.0777, 'age_bin'),
 (0.0764, 'hours-per-week'),
 (0.07, 'occupation'),
 (0.0405, 'workclass'),
 (0.0198, 'loss_bin'),
 (0.0151, 'native-country'),
 (0.0136, 'race'),
 (0.0081, 'marital-status'),
 (0.0044, 'sex')]
```

The feature importance of the data model is classified and sorted based on decreasing order of Gini-Index. From the values, we can see that “sex”, “marital-status”, “race”, “native-country”, “capital-gain” (loss-bin) give the least values of gini-index.

After removal of these features accuracy has improved by 1.4%. Hence, with the improvement in accuracy and with the gini-index values, the above discussed features can be eliminated.

5.MODEL DEVELOPMENT

DECISION TREE

This model is built using the DecisionTreeClassifier from sklearn tree package. Without tuning the parameters, an accuracy of 79.8% is achieved. The criterion for the tree defines the quality of the split. This is chosen as 'gini' as the features are selected based on the gini index. By using various split sizes, split size of 3 gave the better accuracy. Also, for max_depth is calculated for values from 2-5 to avoid over fitting of the data as the number of features is not too high to have a higher depth. The accuracy did not improve with the change in these values. Hence building the model by considering gini criterion and max_depth = 3 gave an accuracy of 81.89%.

ARTIFICIAL NEURAL NETWORK

This model is built using MLPClassifier from sklearn.neural_network package. The model can be classified used various solvers for weight optimization. The current model is tested on 'lfgbs'- (optimizer in the family of quasi-Newton methods) and 'sgd' - stochastic gradient descent. However, 'sgd' has decreased the performance of the model to 71.2%. The different values of learning rate changing from 0.2 to 1, learning rate of 0.5 gave the best performance. With 1 hidden layer, the accuracy dropped down to 74%, this is because the model could not train itself efficiently. The model is built with three hidden layer varying the number of perceptrons from 2-10. Hidden Layer with 8 perceptrons gave the highest accuracy since it keeps a balance between too less and too many arguments avoiding underfitting and overfitting. With all the optimized parameters, NN model achieved an accuracy of 81.98%

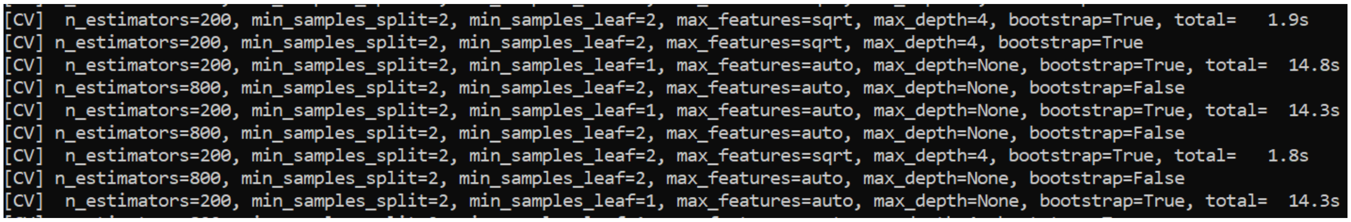
SUPPORT VECTOR MACHINE

This model is built using SVM classifier from sklearn package. The SVM classifier has various tuning parameters such as cache size, kernel, gamma etc. The accuracy of the model improved only by changing the gamma value. Gamma value controls the tradeoff between error due to bias and variance in the model. By using different gamma values this model from 0-15, gamma value of 12 has achieved the highest accuracy of 82.95%

RANDOMFOREST

This model is built using RandomForesrRegressor from sklearn.ensemble package. This model is built considering various tuning parameters using 3 fold cross-validation of the data and the parameters for the best fit is taken to predict the accuracy. The method of running all the combinations of parameters and cross-validating the data is done using RandomisedSearchCV from sklearn.model_selection pacakge.

- `{'n_estimators': [200, 400, 600, 800], 'max_features': ['auto', 'sqrt'], 'max_depth': [4, 6, 8, None], 'min_samples_split': [2, 3], 'min_samples_leaf': [2, 4], 'bootstrap': [True, False]}`



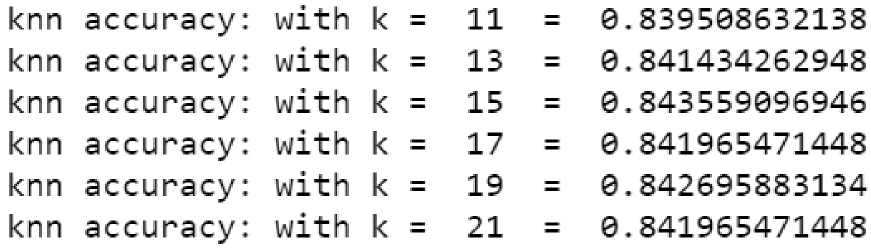
```
[CV]  n_estimators=200, min_samples_split=2, min_samples_leaf=2, max_features=sqrt, max_depth=4, bootstrap=True, total=   1.9s
[CV] n_estimators=200, min_samples_split=2, min_samples_leaf=2, max_features=sqrt, max_depth=4, bootstrap=True
[CV]  n_estimators=200, min_samples_split=2, min_samples_leaf=1, max_features=auto, max_depth=None, bootstrap=True, total=  14.8s
[CV] n_estimators=800, min_samples_split=2, min_samples_leaf=2, max_features=auto, max_depth=None, bootstrap=False
[CV]  n_estimators=200, min_samples_split=2, min_samples_leaf=1, max_features=auto, max_depth=None, bootstrap=True, total=  14.3s
[CV] n_estimators=800, min_samples_split=2, min_samples_leaf=2, max_features=auto, max_depth=None, bootstrap=False
[CV]  n_estimators=200, min_samples_split=2, min_samples_leaf=2, max_features=sqrt, max_depth=4, bootstrap=True, total=   1.8s
[CV] n_estimators=800, min_samples_split=2, min_samples_leaf=2, max_features=auto, max_depth=None, bootstrap=False
[CV]  n_estimators=200, min_samples_split=2, min_samples_leaf=1, max_features=auto, max_depth=None, bootstrap=True, total=  14.3s
```

n_estimators are the number of trees chosen, max_features are the number of features considered for splitting at every node, max_depth is the depth of the decision tree, min_samples_split is the minimum number of splits at each node etc.

The current model is tuned with 4*2*4*2*2*2 combinations of parameters in total and the best model is chosen to test the data.

KNN

This model is built using KNeighborsClassifier from sklearn.neighbours package. In this model, the category of the class is decided on the proximity measure. Current model uses Euclidean distance as proximity measure. As the distances between two vectors are calculated, the values in the vectors should be in the same range. Hence, normalization of the values is performed using preprocessing library from sklearn. The normalized data is tested for various k values ranging from (2 to 30) and the highest accuracy of 84.35% is achieved for k = 15 .



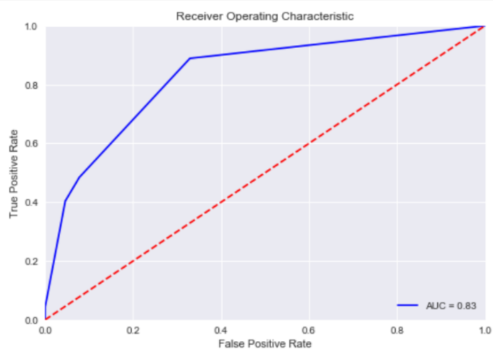
```
knn accuracy: with k =  11  =  0.839508632138
knn accuracy: with k =  13  =  0.841434262948
knn accuracy: with k =  15  =  0.843559096946
knn accuracy: with k =  17  =  0.841965471448
knn accuracy: with k =  19  =  0.842695883134
knn accuracy: with k =  21  =  0.841965471448
```

6.MODEL EVALUATION

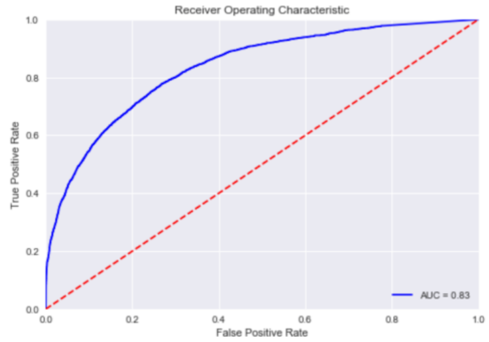
All the models are tested with the given data set of 16281 records. For each model, performance statistics such as True positive rate, false positive rate, true negative rate, false negative rate, precision, recall, f-measure, classification rate(i.e accuracy), error rate, area under curve(auc) are calculated and tabulated.

	Decision Tree	ANN	SVM	RandomForest	KNN
True Positive:	10840	10806	10884	10888	10384
False Positive:	2206	2159	2091	2036	1380
True Negative:	1494	1541	1609	1664	2320
False Negative:	520	554	476	472	976
Precision:	0.830906024835	0.833474739684	0.838843930636	0.84246363355	0.882692961578
Recall:	0.954225352113	0.951232394366	0.958098591549	0.958450704225	0.914084507042
F-Measure:	0.888306154224	0.888468653649	0.894514074378	0.896722121562	0.89811451306
Classification rate:	0.818990703851	0.819853917663	0.829548472776	0.833466135458	0.843559096946
Error rate:	0.181009296149	0.180146082337	0.170451527224	0.166533864542	0.156440903054
Auc	0.83064651	0.83459793	0.89459793	0.89870799	0.89142696

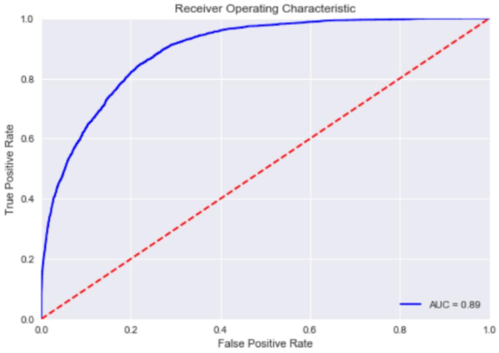
Decision Tree



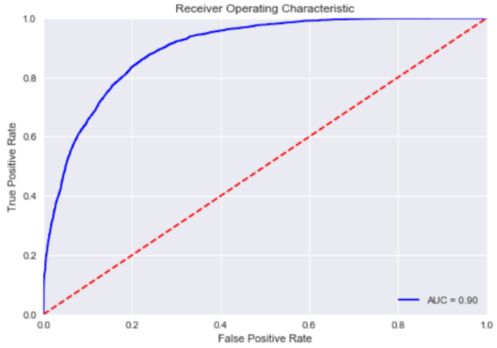
ANN



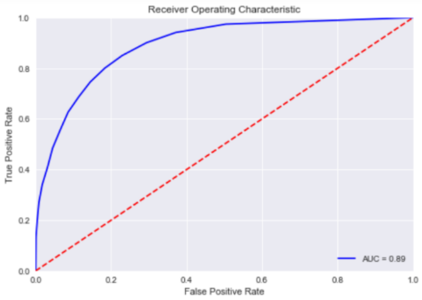
SVM



RandomForest



KNN



COMPARISON AND CONCLUSION OF MODELS:

- From the accuracy perspective, all the models have accuracies between 81-85%. The best accuracy is achieved by KNN model (84.35%) followed by Random Forest (83.34%) with not much difference in their accuracies.
- Also, with the F-Measure, precision and recall values for all the models, KNN has the highest values followed by Random Forest and SVM.
- Random Forest classifier has the highest Area Under ROC curve.
- For this dataset, all the models perform fairly well with slight differences in their accuracies. However, with the current optimizations, transformations and feature selections, KNN performs the best.
- But, there are a few drawbacks with the KNN model. For a KNN classifier, all the attributes should be converted to numerical value and should be normalized to compute the proximity measure. The current dataset has many categorical features which are converted to corresponding numerical values based on the category and passed onto KNN for the evaluation. When the given categorical values are converted to numerical attributes and normalized, they do not give the exact information compared to initial categories.
- When the dataset is having too many categorical attributes, a classifier like Random Forest or Decision Tree is an ideal classifier because, these models do not require normalization of the data. The modelling can be performed without changing the nature/information of datatype.
- As we can observe that there is minute difference in the accuracy and other measures, even though KNN has the highest accuracy by 0.01%, I would like to choose the Random Forest Classifier based on the above discussion.
- The only drawback of this model is the time taken to tune the model and select the parameters. Apart from this the models give a good accuracy of 83.34%