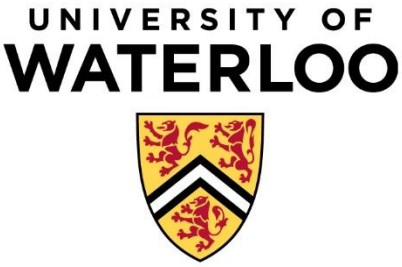
ECE-657A Data and Knowledge Modeling and Analysis
Winter 2018

Assignment 2
Classification

Prepared by

Samina Islam Eva
20704504
Souradip Dey
20741206
Md Rubayatur Rahim Bhuyian
20743420



Department of Electrical & Computer Engineering

Part-I

1.

At first we deleted all the samples having features with more than one unknown value. Then in the below manner we filled out the rest of the unknown values in different attributes. We did not remove the samples with unknown values because by doing that we would have lost ¼ th of the data.

Marital: We grouped the data points on their type and find the corresponding average age. So we filled the unknowns on the basis of age. The average age of unknown samples is almost equal to the average age of married persons. So all the unknowns have been filled out as married in marital attribute.

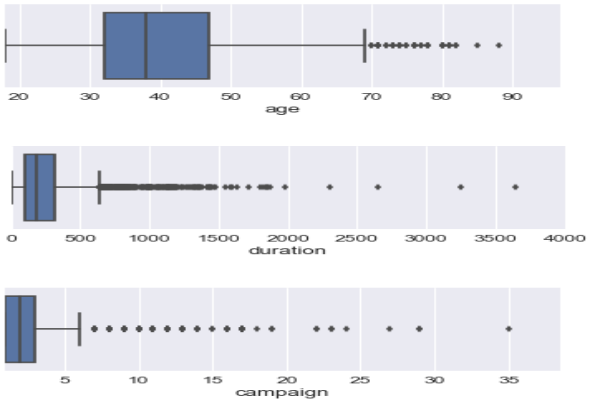
Education: At first the job category has been observed to find out the most frequent education type each job has. Then we filled out the unknowns of education on the basis of job. As an example we observed that most people doing blue-collar job has basix.9y education. So the unknown values in education attribute having blue-collar job was filled out by basic.9y. The table has been given below for the entire dataset.

Job	Education	Job	Education
admin	university.degree	self employed	university.degree
blue collar	basic.9y	service	high.school
entrepreneur	University.degree	student	High.school
housemaid	basic.4y	technician	professional.course
management	university degree	unemployed	University.degree
retired	basic.4y		

Default: All the values in default feature is no. So the unknowns in this attribute has been filled out with 'no'.

After going through above processing there were a marginal data point left with unknown values in the data set which could not be related. These were also deleted. Finally we had 3935 observations out of 4119 which is good enough to train and test our models.

Outlier: We don't have any significant outliers in our data set.



In the age feature most of our data points are under 70. But age can be more than 70 thus we cannot consider those as outliers. Similarly one can talk very long time after picking up the phone. And the duration was recorded only after picking up the phone thus the high values are not outliers. Campaign denotes number of contact performed during campaign which can be 30 or 40 times so these values also cannot be counted as outliers.

2.

We divided our data into 70 to 30 train test ratio. In our data set we have 4119 samples. Dividing it 70 to 30 ratio will give us enough data for train our model also the rest 30% of our data can be used for prediction and evaluation of our model. If we take more data say 95% it will over fit our model, the train accuracy might be high but the model will not be generalized. On the hand if we decide to choose high percentage of data for test it may result in under fitting. So, 70% to 30% or 80% to 20% is a good pick for training and testing the data.

3.

We applied classification using Decision Trees, Random Forests and Neural Networks. The classification properties has been given below.

Decision Trees:

We changed the max depth of the decision tree from 1 to 20 and at max depth of 5 we found max accuracy. That's why we chose depth of the tree as 5.

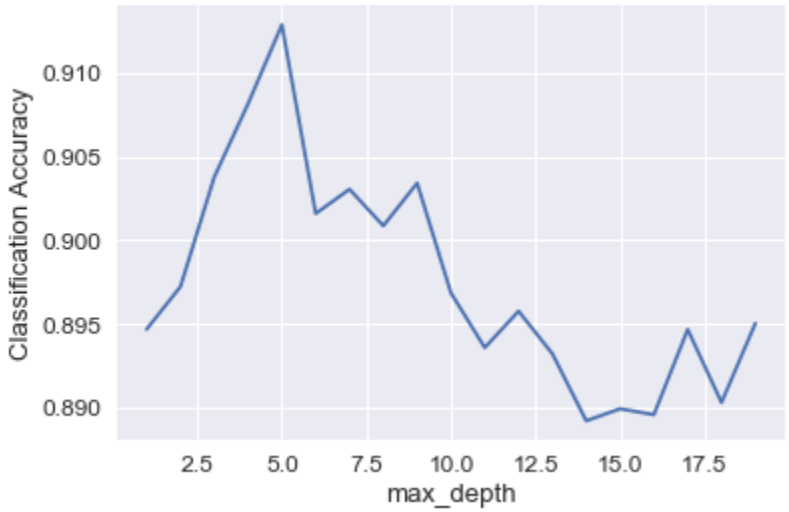


Figure: Depth of Decision Tree

Random Forest:

We varied the number of trees in our random forest model and found that the increment of trees after 20 don't impact the accuracy in a significant way. That's why we chose 20 as our number of tree.

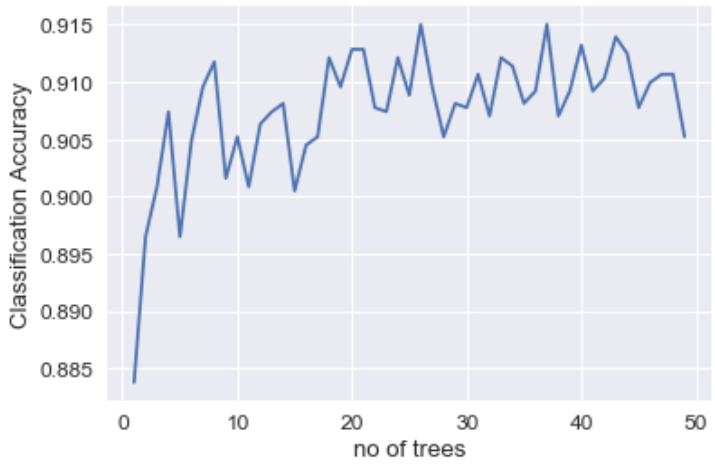


Figure: Accuracy vs number of Trees in Random Forest

Neural Network:

We choose 2 hidden layer because there is no significant change in accuracy if we increase the number of layer after that. To keep our training time in less we choose 2 hidden layer.

Figure: NN accuracy vs Hidden Layer

Decision Tree(Cart): The algorithm is based on Classification and Regression Trees A CART tree is a binary decision tree that is constructed by splitting a node into two child nodes repeatedly, beginning with the root node that contains the whole learning sample.

Random Forest: Random forest is an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Neural Network: Neural networks provide models of data relationships through highly interconnected, simulated “neurons” that accept inputs, apply weighting coefficients and feed their output to other “neurons” which continue the process through the network to the eventual output. Some neurons may send feedback to earlier neurons in the network. Neural networks are

"trained" to deliver the desired result by an iterative (and often lengthy) process where the weights applied to each input at each neuron are adjusted to optimize the desired output.

4.

Neural Network:

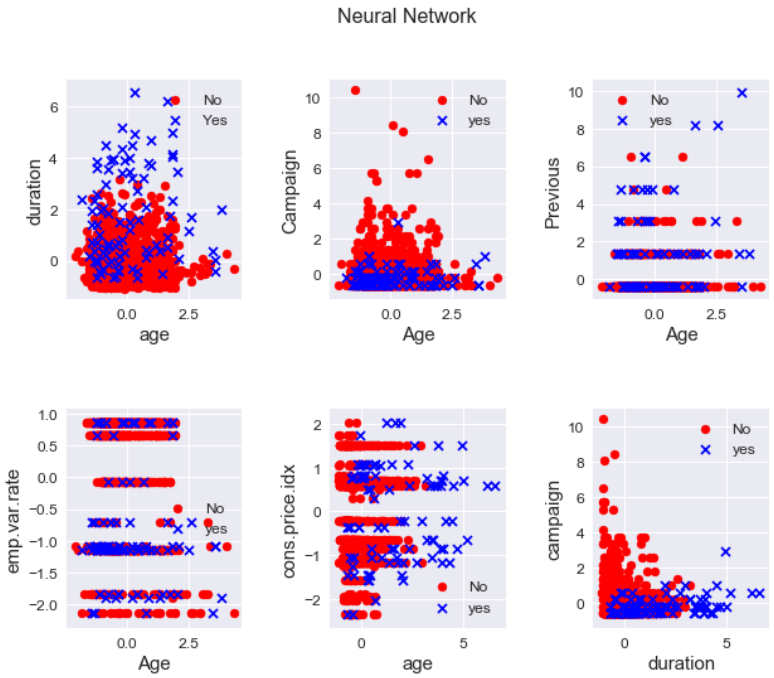


Figure: Two Dimension Plot for Random Forest

Random Forest:

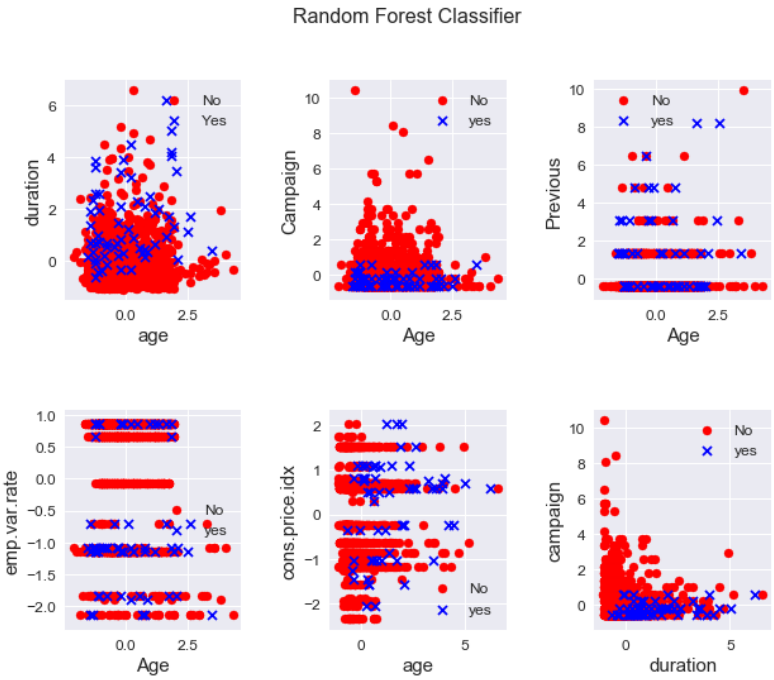


Figure: Two Dimension Plot for Random Forest

Decision Tree:

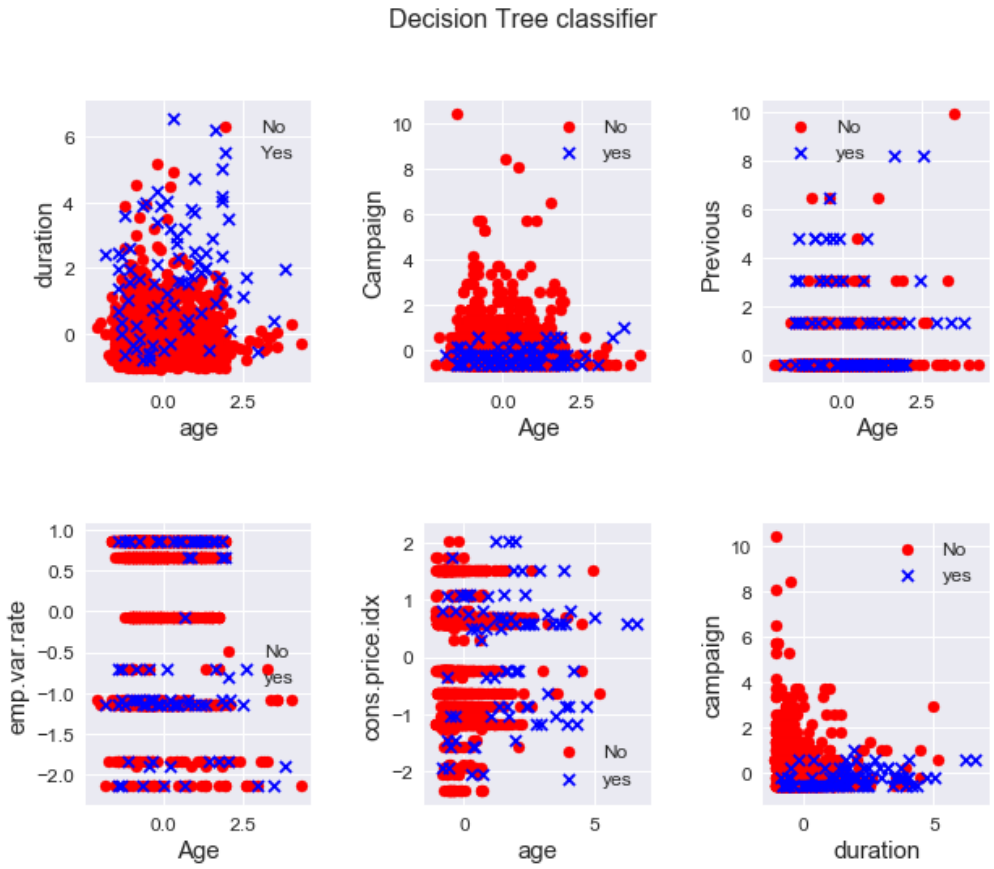


Figure: Two Dimension Plot for Decision Tree

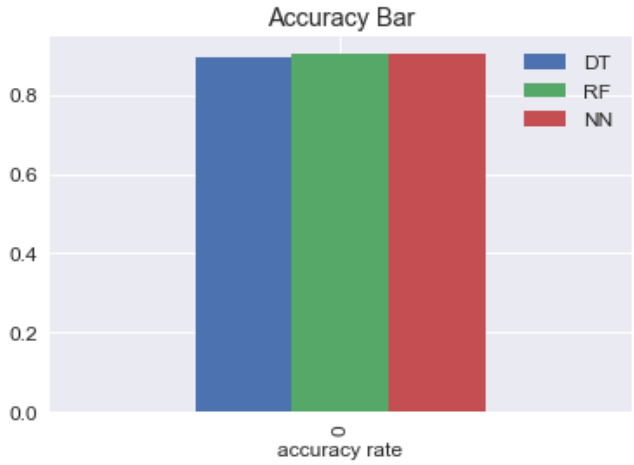
None of the plots showed great separation in our case.

5.

The confusion matrix of each algorithm and their accuracy is given below.

Index	TN	FP	FN	TP
Decision Tree	992	70	52	67
Random Forest	1030	32	84	35
Neural Network	1021	41	73	46

Index	DT	RF	NN
0	0.896698	0.901778	0.903472



6.

The accuracy of different algorithms is given below. Though Neural Network performs best the difference among the accuracies are really marginal.

```
Decision Tree Accuraccy: 0.891617273497
Random Forest Accuraccy: 0.8950042337
Nural Network Accuraccy: 0.900931414056
SVM Accuraccy: 90.0084674005
KNN Accuraccy: 88.3149872989
```

Neural networks provide models of data relationships through highly interconnected, simulated "neurons" that accept inputs, apply weighting coefficients and feed their output to other "neurons" which continue the process through the network to the eventual output. Some neurons may send feedback to earlier neurons in the network. Neural networks are "trained" to deliver the desired result by an iterative (and often lengthy) process where the weights applied to each input at each neuron are adjusted to optimize the desired output. That's why it performs better.

We applied SVM and k-NN in homework 5. SVM and Neural Network provides almost same accuracy as both deals with non-linear relationships with variable and there is not enough observations and features for NN.

7.

By counting the values of each class we have seen that our data is imbalanced. We have 3668 samples of class 'no' and 451 samples of class 'yes' which is comparatively low. So to tackle this discrepancy of the size of the classes we performed both Down-Sampling and Up- Sampling. At first we run Logistic Regression on our raw data which gives us almost 90% accuracy but the model is biased by the class 'no'. So we tried both down and up sampling. Down-sampling involves randomly removing observations from the majority class to prevent its signal from dominating the learning algorithm and Up-sampling is the process of randomly duplicating observations from the minority class in order to reinforce its signal. After performing Down Sampling we got 70.2% accuracy and by doing Up-Sampling we got 69.8% Accuracy. Though the accuracy is less but our model is not biased anymore. Another tactic we can consider is using tree-based algorithms. Decision trees often perform well on imbalanced datasets because their hierarchical structure allows them to learn signals from both classes.

Part: II

1.

We chose Z-score normalization over min max because min max normalization will encounter out of bound errors if a future input case for normalization fall outside of the original data range (any value except 1 to 4 in our case).

Moreover, in the DNA data set the actual minimum and the maximum of the attributes are unknown as the description of the attributes are not given. So it is better to use Z score normalization in this case.

Normalization is done to have the same range of values for each of the inputs in any model. This can guarantee stable convergence of weight and biases. Also we do normalization when we look for relations between the data points and identifying the outliers.

We normally split our data randomly for train and test set so that the estimate has lower bias. We want to avoid introducing any systematic differences between train and test. For example, we would never want to select the first half of the data for learning as there is a risk that the data has been ordered in a specific way.

The distribution of the +1, -1 classes in the dataset is given below.

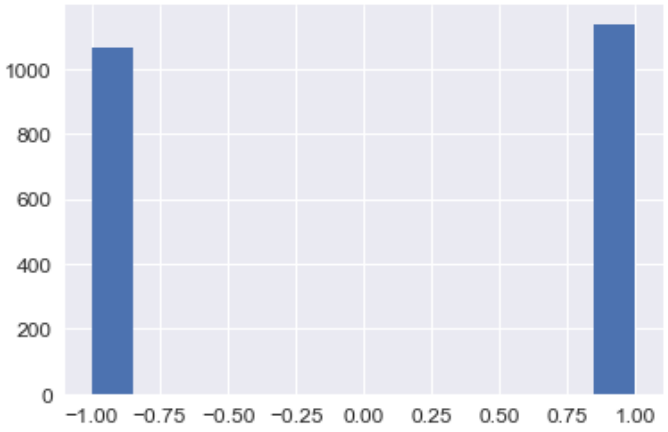


Figure: Distribution of the Classes

From the above distribution we can say that classes are almost equally distributed.

+1 Label ratio = 1137/2200 =0.5169

-1 Label ratio = 1063/2200 =0.4831

2. a)

From the below figure we can say that at K=15 we can achieve highest accuracy.

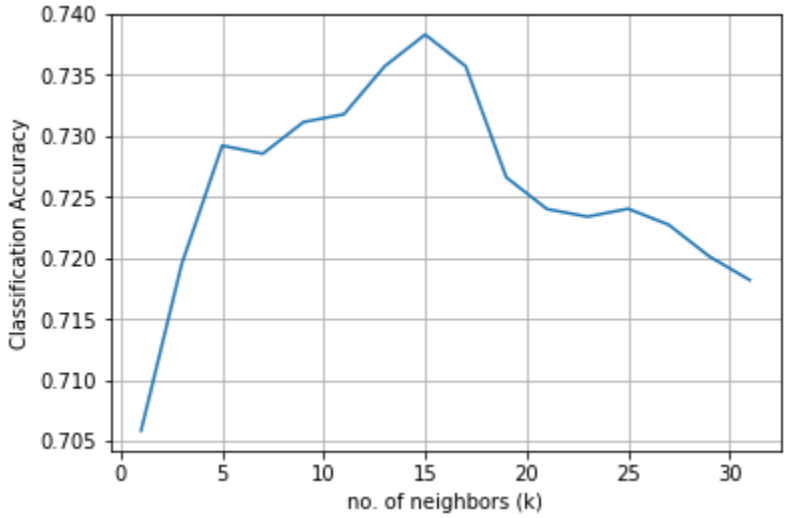


Figure: Accuracy vs K value.

The best K is the one that corresponds to the highest test accuracy rate, we carried out repeated measurements of the test accuracy for different values of K. If we evaluated directly on our test set then we are underestimating the true error rate since our model has been forced to fit the test set in the best possible manner (overfitting). Hence we didn't evaluate directly on the test set rather we used 5-fold cross validation.

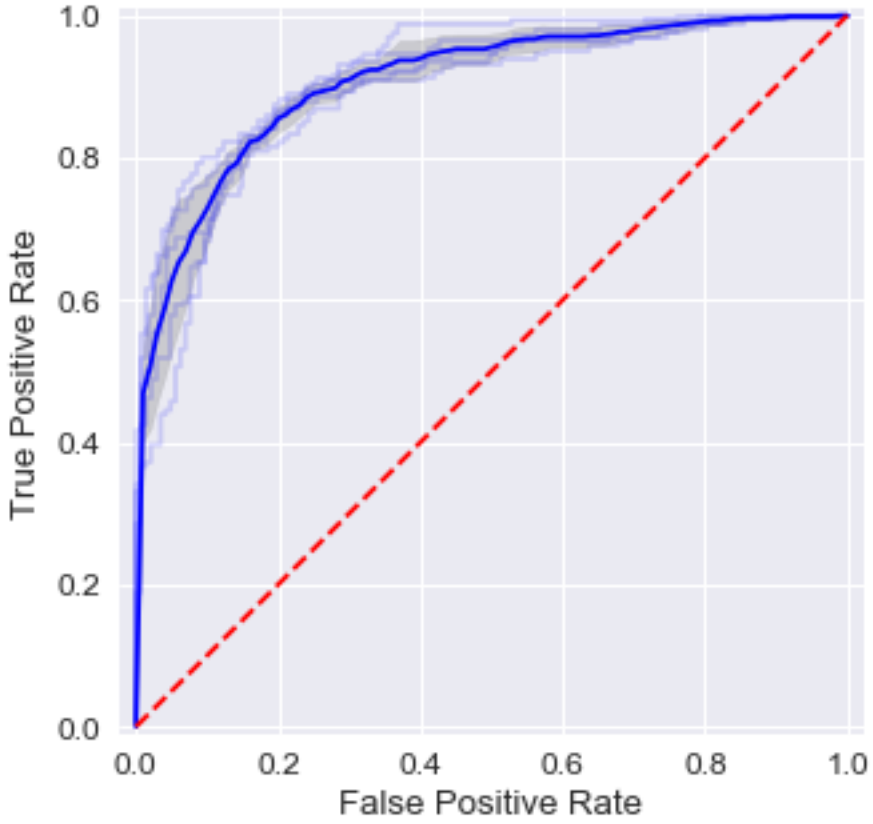
2. b)

For various values of C and Gamma we have calculated Accuracies for various combinations.

	0	1	2	3	4	5	6	7
0	0.528572	0.528572	0.528572	0.528572	0.528572	0.528572	0.828564	0.825328
1	0.540254	0.540254	0.540254	0.54155	0.545448	0.561686	0.879208	0.850651
2	0.580507	0.580507	0.580507	0.590247	0.604537	0.639609	0.891552	0.855846
3	0.580507	0.580507	0.580507	0.591546	0.606483	0.644804	0.897398	0.86558
4	0.580507	0.580507	0.580507	0.591546	0.606483	0.644804	0.903244	0.884407
5	0.580507	0.580507	0.580507	0.591546	0.606483	0.644804	0.902595	0.895439
6	0.580507	0.580507	0.580507	0.591546	0.606483	0.644804	0.902595	0.898699
7	0.580507	0.580507	0.580507	0.591546	0.606483	0.644804	0.902595	0.894786

For c=5 and Gamma= 0.02 we obtained the highest accuracy which is 0.903244.

Below is the ROC curve drawn for the optimum value of C=5 and Gamma=0.02.



3. a)

Following parameters has been chosen for the algorithms.

Algorithm	Parameters
K-NN	No. of neighbor=15
SVM	c=5 and Gamma= 0.02
Random Forest	n_estimators=100,min_samples_leaf=1
NN	hidden_layer_sizes=(10),max_iter=500,actvation="tanh",solver='sgd'

```
Accuracy_SVM: 0.918181818182
Accuracy_KNN: 0.748484848485
Accuracy_RF: 0.969696969697
Accuracy_NN: 0.85
```

3. b)

Random Forest

Model with rank: 1
Mean validation score: 0.9558)
Parameters: {'bootstrap': False, 'max_depth': 6, 'n_estimators': 300}

Model with rank: 2
Mean validation score: 0.9532)
Parameters: {'bootstrap': False, 'max_depth': 6, 'n_estimators': 100}

Model with rank: 3
Mean validation score: 0.9519)
Parameters: {'bootstrap': False, 'max_depth': 6, 'n_estimators': 200}

Model with rank: 4
Mean validation score: 0.9513)
Parameters: {'bootstrap': False, 'max_depth': 5, 'n_estimators': 200}

Model with rank: 5
Mean validation score: 0.9506)
Parameters: {'bootstrap': True, 'max_depth': 6, 'n_estimators': 200}

Model with rank: 6
Mean validation score: 0.9487)
Parameters: {'bootstrap': False, 'max_depth': 5, 'n_estimators': 300}

The parameters of Random Forest like depth and number of the tree was varied multiple times with both combination of bootstrap. As bootstrap works better with imbalance dataset we found that with bootstrap disabled, 6 depth and 300 number of tree give us the best accuracy. Though different combination provide different accuracy but it is not that much significant.

Neural Network

Model with rank: 1
Mean validation score: 0.8903)
Parameters: {'activation': 'relu', 'hidden_layer_sizes': 100, 'solver': 'lbfgs'}

Model with rank: 2
Mean validation score: 0.8786)
Parameters: {'activation': 'relu', 'hidden_layer_sizes': 100, 'solver': 'adam'}

Model with rank: 3
Mean validation score: 0.8721)
Parameters: {'activation': 'logistic', 'hidden_layer_sizes': (100, 100, 100), 'solver': 'lbfgs'}

Model with rank: 4
Mean validation score: 0.8695)
Parameters: {'activation': 'relu', 'hidden_layer_sizes': (100, 100, 100), 'solver': 'lbfgs'}

Model with rank: 5
Mean validation score: 0.8630)
Parameters: {'activation': 'relu', 'hidden_layer_sizes': (100, 100, 100), 'solver': 'lbfgs'}

Model with rank: 6
Mean validation score: 0.8513)
Parameters: {'activation': 'relu', 'hidden_layer_sizes': (100, 100, 100), 'solver': 'adam'}

The parameters of Neural Network like hidden layer, number of neuron in each layer and activation has been changed to find out the best combination for the accuracy of this dataset. We found out that with an activation function Rectified Linear Unit (ReLu) and a single layer with hundred neurons give us the best accuracy of 89.03%. We tried several combination of layer size, activation function and solver in Neural Network.

3. c)

The classification was repeated 20 times to get an estimate about the variance of the performance (accuracy, precision, recall, and F- Measure) of each algorithm.

Among all the classifier SVM achieved highest accuracy.

Index	KNN_Accuracy	RBF_SVM_Accuracy	RandomForest_Accuracy	NN_Accuracy
mean	0.74947	0.899545	0.802348	0.880985
std	0.0212515	0.0109584	0.0333212	0.0221154

Index	KNN_f_score	RBF_SVM_f_score	RandomForest_f_scorey	NN_f_score
mean	0.692918	0.897463	0.793016	0.869725
std	0.0293674	0.00877641	0.0271973	0.0257014

Index	KNN_precision	BF_SVM_precisio	RandomForest_precision	NN_precision
mean	0.943618	0.912776	0.775681	0.893479
std	0.0182883	0.0149519	0.0474028	0.0377948

Index	KNN_recall	RBF_SVM_recall	RandomForest_recall	NN_recall
mean	0.548971	0.882947	0.814866	0.850337
std	0.0392646	0.0141654	0.0445349	0.050727

Training Time	Nearest Neighbors	RBF SVM	Random Forest	Neural Net
mean	0.00435	0.1630205	0.015387	0.625361
std	0.001013657	0.00775985	0.001285068	0.231828916

Classification Time	Nearest Neighbors	RBF SVM	Random Forest	Neural Net
mean	0.119686	0.0519675	0.001825	0.0010245
std	0.019649428	0.003692633	0.000238506	0.000370006

For neural network we found that training time was highest where as it takes lowest time for classification. On the other hand K-NN takes lowest time for training but highest time for classification.

4)

	Accuracy	Training time	Testing time
KNN	74.947%	0.0035	0.11750
SVM	89.95%	0.166	0.053
Random Forest	80.2348%	0.015387	0.0018
Neural Network	88.0985%	0.625361	0.00010

For the given data set SVM performed best with moderate training and testing time.

PRO's and Con's of different algorithm are given below:

k-NN: Less training time high test time. If there is imbalance presence in the dataset prediction will be biased. Computational time is high if the data set is large. Training time is fast as in training period we basically introduce the labels to the classifier.

SVM: Powerful classifier, maps to higher dimensional to make nonlinearly separable into linearly separable. It uses the kernel trick, so we can build in expert knowledge about the problem via engineering the kernel as we did here by varying the gamma and cost value.

Random Forest: Runs efficiently on large dataset, provides estimate of what variables are important in a dataset, it has methods for balancing error in class population unbalanced data sets. It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

Neural Network: It's difficult to interpret. For data including categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.

As SVM classifier, maps to higher dimensional to make nonlinearly separable into linearly separable. It was expected that SVM would perform better on our data set as we did not had a linearly separable data. Though the accuracy of neural network is quite close to SVM it takes much longer time to train and will work better with huge data set with lots of attributes. So SVM should be the ultimate choice for these type of data set.

5)

If we had to remove one feature from the dataset we had to find out the mutual information of each feature and rank them accordingly. Then the feature which having less mutual information can be removed. One other way could be, we can find the correlation between the features and if two feature provide us the same information for classification we can get rid of one. If we classify based on only two dimension there is a possibility that we lose valuable information which will lead to less accuracy of the model.

For the given dataset we calculated the mutual information and obtained the following:

```
Top Feature ranking:
1. feature 27 (0.146350)
2. feature 28 (0.118761)
3. feature 30 (0.089779)
4. feature 29 (0.076379)
5. feature 26 (0.037991)
6. feature 32 (0.029496)
7. feature 33 (0.027695)
8. feature 31 (0.027159)
9. feature 20 (0.018621)
10. feature 24 (0.018493)
```

……..

```
50. feature 10 (0.006515)
51. feature 54 (0.006478)
52. feature 50 (0.006451)
53. feature 55 (0.006113)
54. feature 5 (0.006029)
55. feature 51 (0.005867)
56. feature 56 (0.005618)
57. feature 1 (0.005316)
```

In our case we would have removed feature 1 if we had to.

Also reduction in feature gave lower accuracy.

```
Accuracy before dimension reduction is:
0.925757575758
Accuracy after dimension reduction is:
0.534848484848
```