Applied Machine Learning Project Report

Nandini Goswami
Indiana University Bloomington
goswamin@iu.edu

Sarita Bhateja
Indiana University Bloomington
sbhateja@iu.edu

ABSTRACT

This project report demonstrates the use of various classifiers like decision tree, logistic regression etc for prediction on two data sets.

1. INTRODUCTION

The project demonstrates the implementation of various machine learning algorithms on two datasets. The results have been analysed and corresponding graphs have been plotted.

2. DATA SET DESCRIPTION

1. Dataset 1 Title: Predict an employee's access needs, given his/her job role
The objective of this dataset is to build a model, learned using historical data, that will determine an employee's access needs, such that manual access transactions (grants and revokes) are minimized as the employee's attributes change over time. The model will take an employee's role information and a resource code and will return whether or not access should be granted.

Attribute Information:
ACTION- ACTION is 1 if the resource was approved, 0 if the resource was not
RESOURCE- An ID for each resource
MGR ID- The EMPLOYEE ID of the manager of the current EMPLOYEE ID record; an employee may have only one manager at a time
ROLE ROLLUP 1- Company role grouping category id 1 (e.g. US Engineering)
ROLE ROLLUP 2- Company role grouping category id 2 (e.g. US Retail)
ROLE DEPTNAME- Company role department description (e.g. Retail)
ROLE TITLE- Company role business title description (e.g. Senior Engineering Retail Manager)
ROLE FAMILY DESC- Company role family extended description (e.g. Retail Manager, Software Engineering)
ROLE FAMILY- Company role family description (e.g. Retail Manager)
ROLE CODE- Company role code; this code is unique to each role (e.g. Manager) [1]

2. Dataset 2 Title: Contraceptive Method Choice
This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of interview. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics.

Attribute Information:

1. Wife's age- This contains numerical data that denotes age of the women.
2. Wife's education-This is categorical data where 1 denotes low education and 4 denotes high education.
3. Husband's education-This is same as wife's education
4. Number of children ever born-This is numerical data.
5. Wife's religion-This is categorical where 0 is for Non Islam and 1 is for Islam.
6. Wife's now working?-This is categorical where 0 is for Yes and 1 is for No.
7. Husband's occupation - This is categorical having values 1,2,3,4.
8. Standard-of-living index -This is categorical where 1 is for low and 4 is for high.
9. Media exposure-This is categorical where 0 is for Good and 1 is for Bad.
10. Contraceptive method used -This is the class label and has values 1 for no use, 2 for long term and 3 for short term. [2]

3. FEATURE SELECTION AND PREPROCESSING

For feature selection we ran random forest for evaluating feature importance. As the Figure1 shows the column 1 and 0 have very high importance whereas the others have low importance comparatively.We have removed the features that hardly contribute to making of decision.

Apart from that in prepossessing we have removed all null values and normalised the features to ensure better predictions by the various classifiers.

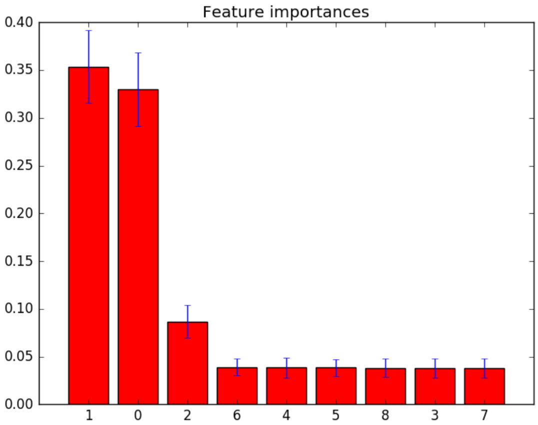


Figure 1: Histogram for feature importance for contraceptive dataset

The column with Resource ID was removed from Amazon dataset because it has any unique IDs and there is no linkage between Resource ID and access. This feature was hampering the accuracy and results of all the algorithms.

Additionally, the data sets were unbalanced and skewed. To avoid the case of bias, under sampling is done in order to get better results.

4. ALGORITHMS USED

For both of the datasets, we have used the following algorithms:

1. Decision Tree
2. Logistic Regression
3. Adaboost
4. Support Vector Machine
5. k-Nearest Neighbor 6. Random Forest

5. RESULTS AND ANALYSIS

Dataset1:
Decision Tree:
The Figure 2 shows Graphical representation of Depth vs Accuracy. When depth is small, the accuracy is low because of underfitting. As depth grows, the accuracy increases and when depth is significantly higher, the accuracy drops because of overfitting. The graph is not representing overfitting because it is getting overfit at high depth greater than 40.

The Figure 3 shows the confusion matrix of decision tree when accuracy is highest.

The Figure 4 shows the ROC curve of Decision tree.

k-Nearest Neighbour:
The Figure 5 shows the graph between k and accuracy. When k is small, the accuracy drops at first due to the case if overfitting, as k increases, the accuracy also increases and

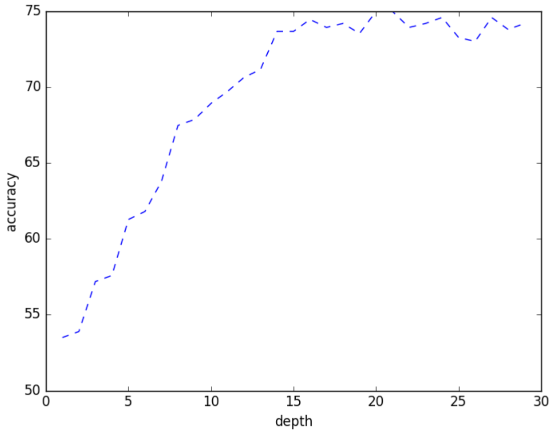
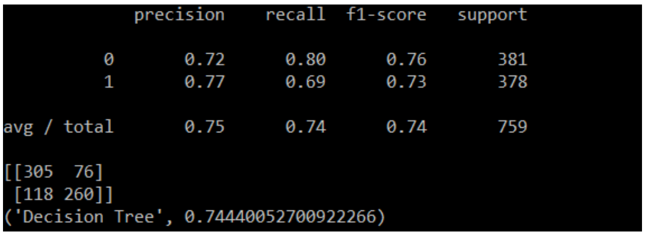


Figure 2: Graph for Depth vs Accuracy (Dataset1)



```
             precision    recall  f1-score   support

          0       0.72      0.80      0.76       381
          1       0.77      0.69      0.73       378

avg / total       0.75      0.74      0.74       759

[[305  76]
 [118 260]]
('Decision Tree', 0.74440052700922266)
```

Figure 3: Confusion matrix for Decision tree (Dataset 1)

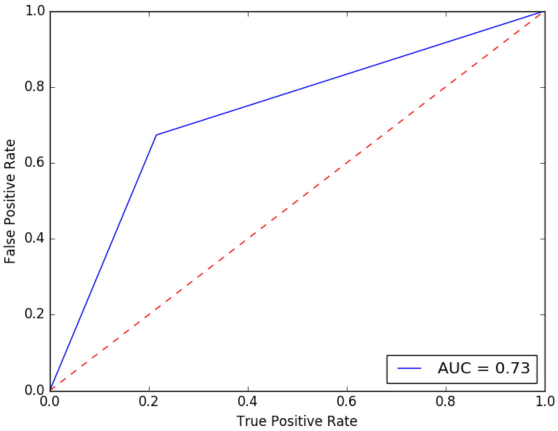


Figure 4: ROC Curve for Decision tree (Dataset 1)

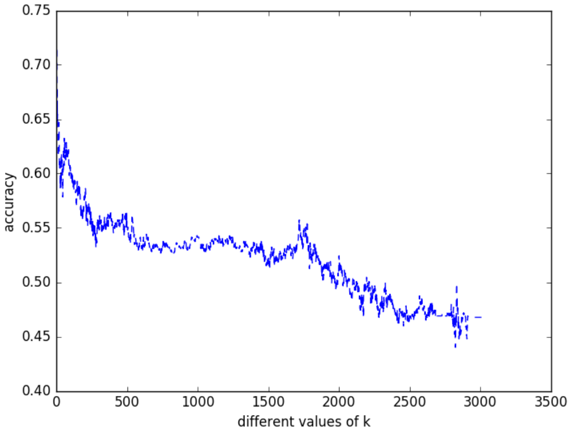
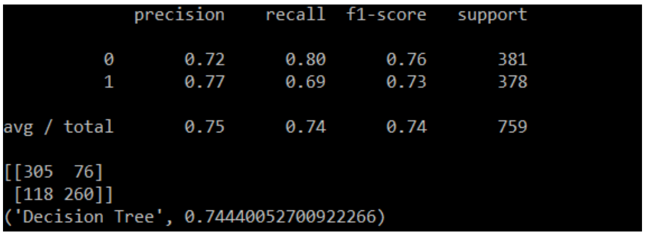


Figure 5: Graph for Accuracy vs k (Dataset1)



```
              precision    recall  f1-score   support

          0       0.72      0.80      0.76       381
          1       0.77      0.69      0.73       378

avg / total       0.75      0.74      0.74       759

[[305  76]
 [118 260]]
('Decision Tree', 0.74440052700922266)
```

Figure 6: Confusion matrix for k-Nearest Neighbour (Dataset 1)

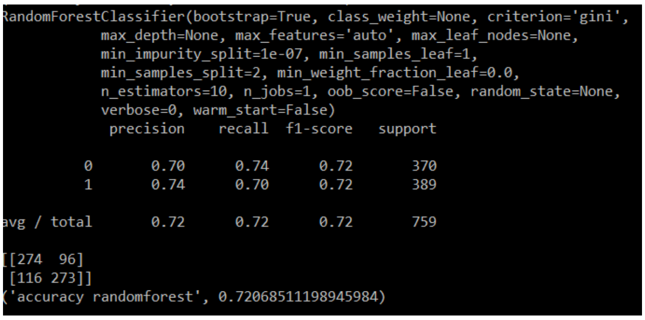
the fluctuation is low. When k becomes quite large, the accuracy drops steeply due to the case of underfitting.

The Figure 6 shows the confusion matrix of decision tree when accuracy is highest.

Random Forest:
The Figure 7 shows the confusion matrix for Random forest.

The Figure 8 shows the ROC curve for Random forest.

SVM:
The Figure 9 shows the confusion matrix for SVM.



```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            n_estimators=10, n_jobs=1, oob_score=False, random_state=None,
            verbose=0, warm_start=False)
             precision    recall  f1-score   support

          0       0.70      0.74      0.72       370
          1       0.74      0.70      0.72       389

avg / total       0.72      0.72      0.72       759

[[274  96]
 [116 273]]
('accuracy randomforest', 0.72068511198945984)
```

Figure 7: ROC curve for Random Forest (Dataset 1)

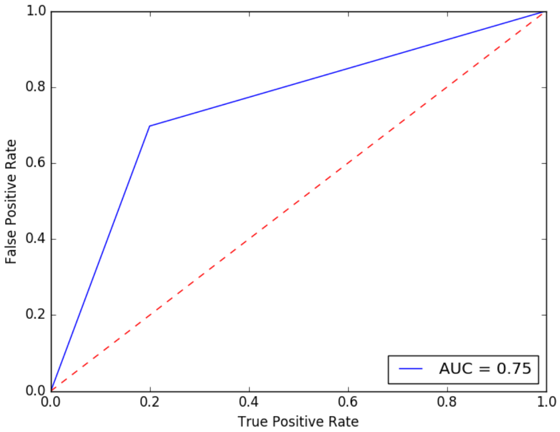
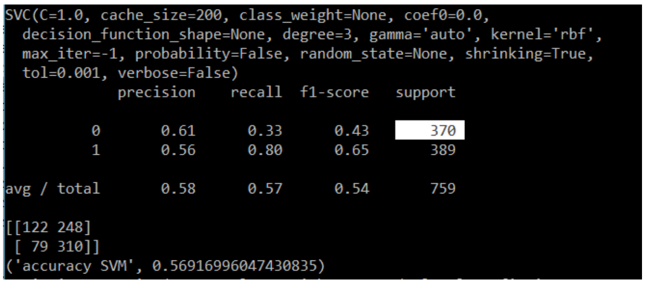


Figure 8: ROC curve for Random Forest (Dataset 1)



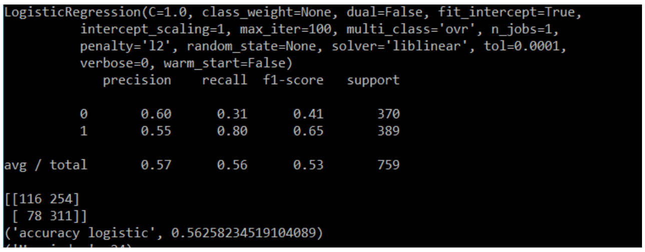
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
             precision    recall  f1-score   support

          0       0.61      0.33      0.43       370
          1       0.56      0.80      0.65       389

avg / total       0.58      0.57      0.54       759

[[122 248]
 [ 79 310]]
('accuracy SVM', 0.56916996047430835)
```

Figure 9: Confusion matrix for SVM (Dataset 1)



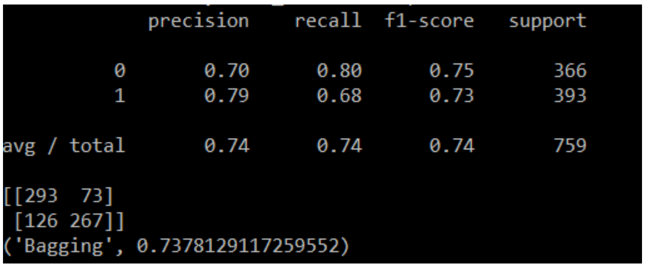
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
             precision    recall  f1-score   support

          0       0.60      0.31      0.41       370
          1       0.55      0.80      0.65       389

avg / total       0.57      0.56      0.53       759

[[116 254]
 [ 78 311]]
('accuracy logistic', 0.56258234519104089)
```

Figure 10: Confusion matrix for Logistic Regression (Dataset 1)



```
             precision    recall  f1-score   support

          0       0.70      0.80      0.75       366
          1       0.79      0.68      0.73       393

avg / total       0.74      0.74      0.74       759

[[293  73]
 [126 267]]
('Bagging', 0.7378129117259552)
```

Figure 11: Confusion matrix for Bagging (Dataset 1)

Logistic Regression:
The Figure 10 shows the confusion matrix for Logistic Regression.

Bagging:
The Figure 11 shows the confusion matrix for ensemble Bagging.
The Figure 12 shows the ROC curve for ensemble Bagging.

Dataset2: Decision Tree:

The Figure 13 shows Confusion Matrix and accuracy for decision tree. The Figure 14 shows Accuracy vs depth in Decision tree. When depth is small, the accuracy is low because of underfitting. As depth grows to a certain point, the accuracy increases and then again, it starts to decline. This is the case of overfitting. So the tree provides best results at depth=5.

The Figure 15 shows the ROC curve for Decision tree (depth=5).

Logistic Regression: The Figure 17 shows the ROC curve for Logistic Regression.

Support Vector Machine: The Figure 18 shows the confusion matrix and accuracy for Support Vector Machine

The Figure 19 shows the the effect on accuracy when the c is changed.

The Figure 20 shows the ROC curve for Support Vector Machine.

Knearest neighbor: The Figure 21 shows the confusion ma-

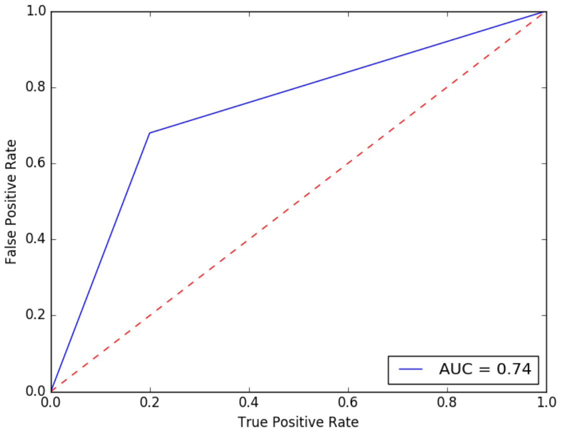
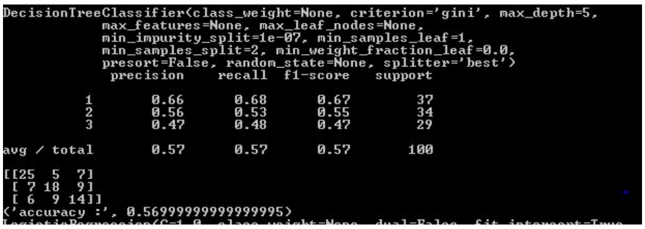


Figure 12: ROC curve for Bagging (Dataset 1)



```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=5,
            max_features=None, max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            presort=False, random_state=None, splitter='best')
             precision    recall  f1-score   support

          1       0.66      0.68      0.67        37
          2       0.56      0.53      0.55        34
          3       0.47      0.48      0.47        29

avg / total       0.57      0.57      0.57       100

[[25  5  7]
 [ 7 18  9]
 [ 6  9 14]]
('accuracy :', 0.56999999999999995)
```

Figure 13: Dataset2: Confusion Matrix and accuracy for decision tree

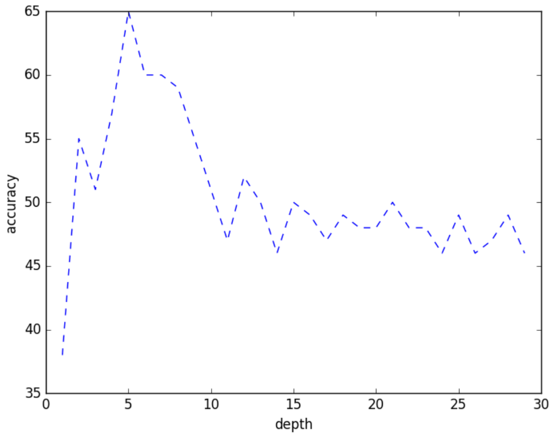


Figure 14: Dataset2:Graph for Depth vs Accuracy

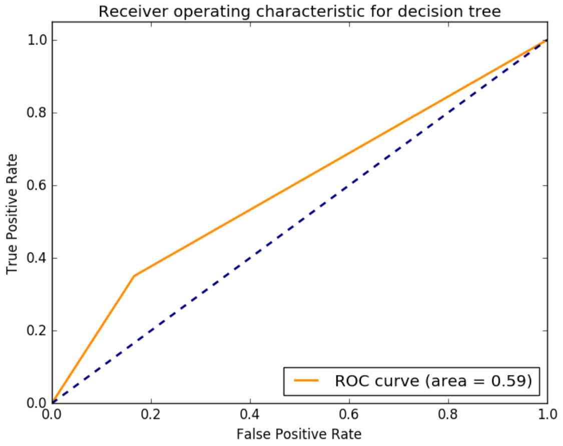
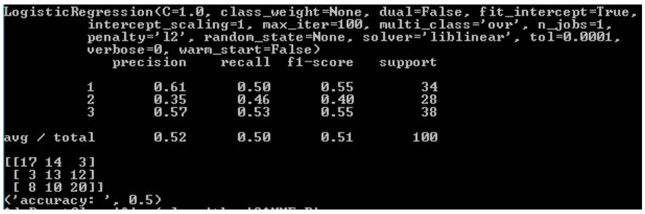


Figure 15: Dataset2:ROC curve for decision tree



```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
             precision    recall  f1-score   support

          1       0.61      0.50      0.55        34
          2       0.35      0.46      0.40        28
          3       0.57      0.53      0.55        38

avg / total       0.52      0.50      0.51       100

[[17 14  3]
 [ 3 13 12]
 [ 8 10 20]]
('accuracy: ', 0.5)
```

Figure 16: Dataset2: Confusion Matrix and accuracy for Logistic Regression

regression graph.png

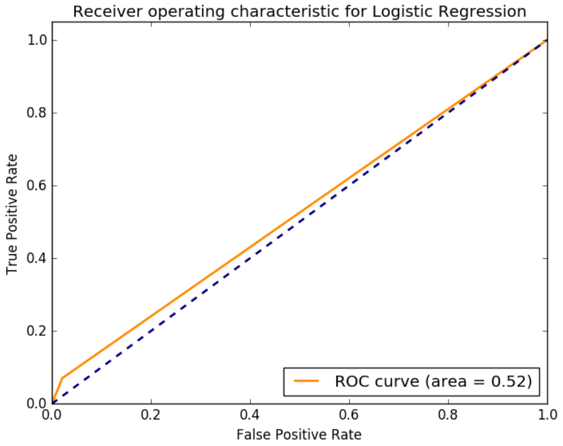
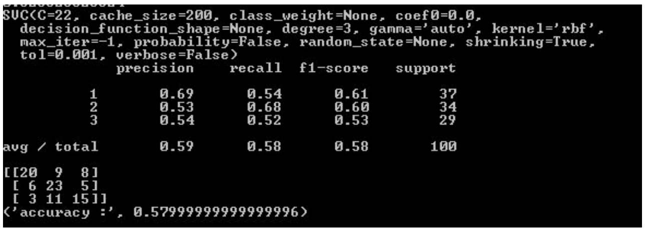


Figure 17: Dataset2:ROC curve for logistic regression



```
SVC(C=22, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
             precision    recall  f1-score   support

          1       0.69      0.54      0.61        37
          2       0.53      0.68      0.60        34
          3       0.54      0.52      0.53        29

avg / total       0.59      0.58      0.58       100

[[20  9  8]
 [ 6 23  5]
 [ 3 11 15]]
('accuracy :', 0.57999999999999996)
```

Figure 18: Dataset2: Confusion Matrix and accuracy for SVM

graph.png

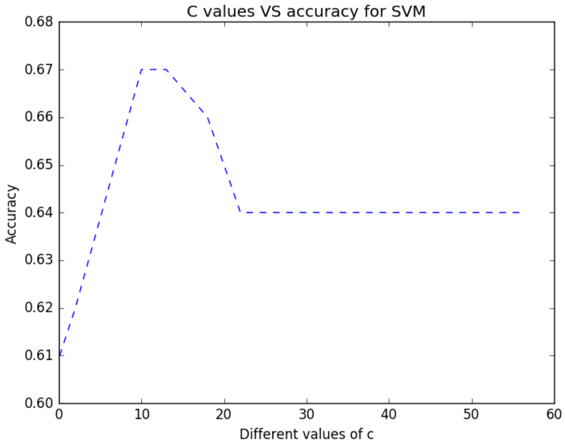


Figure 19: Dataset2:Graph for accuracy with varying c

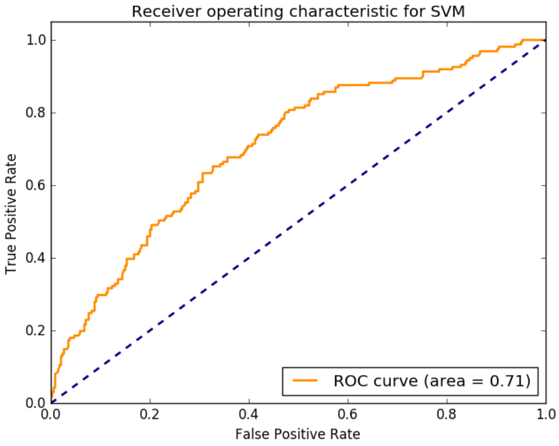
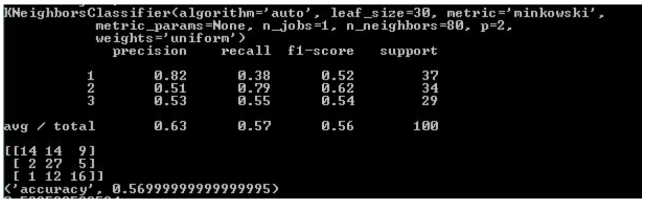


Figure 20: Dataset2:ROC curve for support vector machine



```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=80, p=2,
           weights='uniform')
             precision    recall  f1-score   support

          1       0.82      0.38      0.52        37
          2       0.51      0.79      0.62        34
          3       0.53      0.55      0.54        29

avg / total       0.63      0.57      0.56       100

[[14 14  9]
 [ 2 27  5]
 [ 1 12 16]]
('accuracy', 0.56999999999999995)
```

Figure 21: Dataset2: Confusion Matrix and accuracy for knearest neighbors

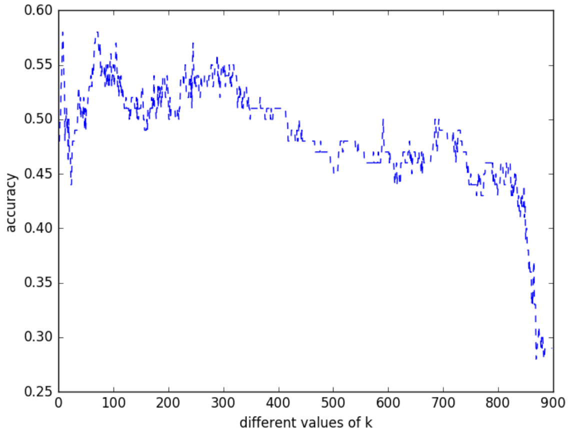


Figure 22: Dataset2:Graph for accuracy with varying k values

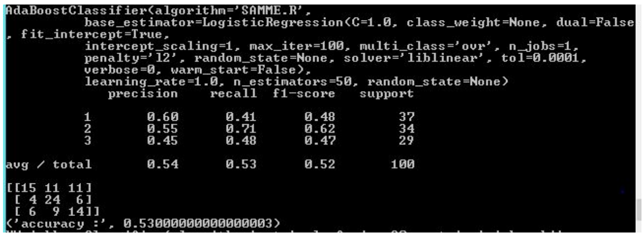
trix and accuracy for knearest neighbor. The Figure 22 shows that when k is small, the accuracy drops at first due to the case if overfitting, as k increases, the accuracy also increases and the fluctuation is low. When k becomes quite large, the accuracy drops steeply due to the case of underfitting.

Adaboost: Figure 23 shows Confusion matrix and accuracy on adaboost. The base classifier for adaboost is logistic regression.

6. REFERENCES

[1] Amazon.com - employee access challenge. Webpage.

[2] Contraceptive method choice data set. Webpage.



```
AdaBoostClassifier(algorithm='SAMME.R',
          base_estimator=LogisticRegression(C=1.0, class_weight=None, dual=False
, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False),
          learning_rate=1.0, n_estimators=50, random_state=None)
             precision    recall  f1-score   support

          1       0.60      0.41      0.48        37
          2       0.55      0.71      0.62        34
          3       0.45      0.48      0.47        29

avg / total       0.54      0.53      0.52       100

[[15 11 11]
 [ 4 24  6]
 [ 6  9 14]]
('accuracy :', 0.53000000000000003)
```

Figure 23: Dataset2: Confusion Matrix and accuracy for Adaboost