Document version: 1.1 (2015-11-15)

Curtin University – Department of Computing

Assignment Cover Sheet / Declaration of Originality

Complete this form if/as directed by your unit coordinator, lecturer or the assignment specification.

Last name:	Mahmood	Student ID:	18747612
Other name(s):	Qaiser		
Unit name:	Data Mining	Unit ID:	COMP3009
Lecturer / unit coordinator:	Sonny Pham	Tutor:	Eric
Date of submission:	14/10/2019	Which assignment?	(Leave blank if the unit has only one assignment.)

I declare that:

- The above information is complete and accurate.
- The work I am submitting is *entirely my own*, except where clearly indicated otherwise and correctly referenced.
- I have taken (and will continue to take) all reasonable steps to ensure my work is *not accessible* to any other students who may gain unfair advantage from it.
- I have *not previously submitted* this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I understand that:

- Plagiarism and collusion are dishonest, and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I plagiarise or collude, I risk failing the unit with a grade of ANN ("Result Annulled due to Academic Misconduct"), which will remain permanently on my academic record. I also risk termination from my course and other penalties.
- Even with correct referencing, my submission will only be marked according to what I have done myself, specifically for this assessment. I cannot re-use the work of others, or my own previously submitted work, in order to fulfil the assessment requirements.
- It is my responsibility to ensure that my submission is complete, correct and not corrupted.

Signature: Qaiser Mahmood Date of signature: 14/10/2019

(By submitting this form, you indicate that you agree with all the above text.)

DATA MINING ASSIGNMENT

Qaiser Mahmood (18747612)

Contents

ASSIGNMENT SUMMARY

In this data mining assignment, I was given two tasks. First task was to prepare the data correctly for classification and second task was to select classifier, train and tune it and make predictions for unknown data.

For task 1 I prepared data using the techniques learned during the lectures and tutorials. The detail of every step has been described later in the report.

For task 2 I used three classifiers that we studied during the workshops. The selected classifiers are KNN, Decision Tree and Naïve Bayes. The detailed discussion on selection criteria and performance evaluation of these classifiers has been given in the later parts of this report.

I got 71% accuracy on validation set and it was within ±5% of estimated accuracy.

What went well during the assignment?

It was very good learning experience. The assignment gave me the opportunity to apply the concepts learned during the lectures and tutorials.

What went wrong during the assignment?

Mostly this assignment went very well at good pace. No major issues / surprises happened. Initially I got very high accuracy but that was due to information leakage from validation set to training set. Although that gave me good opportunity to debug the issues in data preprocessing but it took some time to fixe and I did not have time at the end to use the methods for feature selection to improve the final accuracy.

What could be done differently to improve?

I am getting 71% accuracy on validation set with an estimated accuracy of ±5%. I think it could be improved further if we use methods like Recursive Feature Elimination (RFE) or backward elimination for feature selection.

Similarly we can create an ensemble of two or more classifiers to reduce the error between predicted and validation accuracy.

TASKS COMPLETED

DATA PREPARATION

- ### Irrelevant Attributes:

Following attributes are irrelevant in the data set.

Attribute ID

Type: Numeric.

Issue(s) found: No useful information for data classification.

Decision: Removed from the data set.

Reason: ID is just for the ordering of data and does not have any relationship with the class information of the data.

Attribute att14

Type: Nominal.

Issue(s) found: No change.

Decision: Removed from the data set.

Reason: Same value throughout the data set.

Attribute att17

Type: Numeric.

Issue(s) found: No change.

Decision: Removed from the data set.

Reason: Same value throughout the data set.

- ## Missing Entries:

Following attributes have missing entries in the data set.

Attribute att13

Type: Nominal.

Issue(s) found: Missing entries.

Decision: Removed from the data set.

Reason: 935 missing entries out of 1000.

Attribute att19

Type: Numeric.

Issue(s) found: Missing entries.

Decision: Removed from the data set.

Reason: 937 missing entries out of 1000.

Attribute att3

Type: Nominal.

Issue(s) found: Missing entries.

Decision: Missing values replaced with most occurring value in the attribute.

Reason: Only 2 values are missing out of 1000. The number of missing values is much less than the available values in the attribute, therefore, we can fill the missing values using the available data. For nominal data we can calculate the frequency of the categories and lack any other information about the data, therefore, mode is the most suitable method to fill the missing values.

Attribute att9

Type: Nominal.

Issue(s) found: Missing entries.

Decision: Missing values replaced with most occurring value in the attribute.

Reason: Only 4 values are missing out of 1000. The number of missing values is much less than the available values in the attribute, therefore, we can fill the missing values using the available data.

For nominal data we can calculate the frequency of the categories and lack any other information about the data, therefore, mode is the most suitable method to fill the missing values.

Attribute att25

Type: Numeric.

Issue(s) found: Missing entries.

Decision: Missing values replaced with mean of the attribute values.

Reason: Only 2 values are missing out of 1000. The number of missing values is much less than the available values in the attribute, therefore, we can fill the missing values using the available data.

For numeric data we can calculate the mean of the available data, therefore, I am using mean of the attribute values to fill the missing values.

Attribute att28

Type: Numeric.

Issue(s) found: Missing entries.

Decision: Missing values replaced with mean of the attribute values.

Reason: Only 4 values are missing out of 1000. The number of missing values is much less than the available values in the attribute, therefore, we can fill the missing values using the available data.

For numeric data we can calculate the mean of the available data, therefore, I am using mean of the attribute values to fill the missing values.

- ## Duplicates:

Attribute att8

Type: Nominal.

Issue(s) found: Duplicate.

Decision: Removed from the data set.

Reason: Duplicate of att1.

Attribute att24

Type: Numeric.

Issue(s) found: Duplicate.

Decision: Removed from the data set.

Reason: Duplicate of att18.

- ## Data Type:

Attribute att1 to att12

Type: Nominal.

Issue(s) found: No numeric information.

Decision: Changed from nominal to categorical and assigned a numeric value to each category.

Reason: In the absence of numeric value we cannot calculate statistical properties like normalization, mean etc of the data which are useful for classification.

Attribute att13 to att14

Type: Nominal.

Issue(s) found: No numeric information.

Decision: Changed from nominal to binary.

Reason: In the absence of numeric value we cannot calculate statistical properties like normalization, mean etc of the data which are useful for classification.

- ## Scaling and Standardisation:

Attribute att1 to att30

Type: att1 - att14 (Nominal), att15 – att30 (Numeric).

Issue(s) found: Different scales of values.

Decision: Scaled between 0 and 1 by normalization.

Reason: If we have different scales of values then data can be misclassified due to different weights of the values.

- ## Feature Engineering:

Attribute att1 to att30

Type: att1 - att14 (Nominal), att15 – att30 (Numeric).

Issue(s) found: No domain knowledge.

Decision: Did not use.

Reason: In the absence of any domain knowledge it's very difficult and probably not very effective to create new features from the existing ones.

- ## Data Instances:

Instances: 100 instances throughout the data set.

Type: Not applicable.

Issue(s) found: Duplicates.

Decision: Removed from the data.

Reason: Duplicated instances do not help / provide any extra information which is useful for classification. In fact they can cause over fitting and use computational resources.

Data Imbalance:

Instances: 200 instances of class 0 and 600 instances of class 1.

Type: Not applicable.

Issue(s) found: Class Imbalance.

Decision: Class 0 balanced by using SMOT.

Reason: Class imbalance can create bias in the classifier. The class with more number of instances can be classified more by the classified. The provided test data is also balanced. I used over sampling to remove the imbalance from the data.

Training, Validation and Test Sets:

Test Set Instances: 100 instances from ID 1001 to 1100.

Type: Not applicable.

Issue(s) found: No class information

Decision: Separated as Test Set.

Reason: As specified in the assignment and this data does not have any class information. Therefore, it can serve as our Test Set.

Validation Set Instances: 100 instances randomly selected from ID 1 to 1000.

Type: Not applicable.

Issue(s) found: Not applicable.

Decision: Separated as Validation Set.

Reason: To check the accuracy of classification we need data whose class information is already known to us. We have separated 100 instances from the available data before doing any training. This data is preprocessed separately just like Test data. We can feed this data into the trained classifier and compare the predictions done by classifier with the ground truth class values to calculate the accuracy of the classification.

Training Set Instances: Remaining instances of data after separating the Test and Validation Sets.

Type: Not applicable.

Issue(s) found: Not applicable

Decision: Separated as Training Set.

Reason: In supervised machine learning problem we need data whose class information is already known to us. That data then can be used to train the classifier. This data is preprocessed to make it suitable for classification and fed to the classifier to train it.

DATA CLASSIFICATION

- ## Classifier Selection:

In our data set we have instances and know about class labels, therefore, our problem of classification is related to Supervised Learning. But we don't know about the distribution of data or any domain knowledge of data, therefore, we need a classifier that can handle these problems.
KNN is suitable for this kind of application.
Decision Tree is also a good candidate for binary class labels.
Naïve Bayes is a bit complex than both of the above but is very efficient and accurate for this kind of problems.

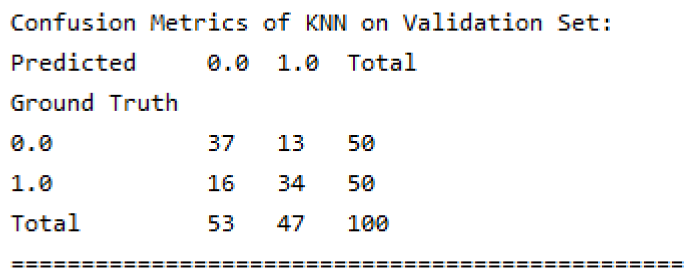
- ## Cross Validation:

To measure the effectiveness of classifier we can split the training data into validation set and training set. After training the classifier on training data, we feed the validation set into the trained classifier and compare the predicted results with the ground truth class labels of validation set. The accuracy of model on validation set gives us the performance of model on unseen data.
I selected the validation set that closely resembled with the test data. My test set and validation set both were balanced. They both contain 50 class labels from each class. But my training data was highly imbalanced. That contained 200 instances from class 0 and 600 instances from class 1. Class imbalance can create bias in the classifier. The class with more number of instances can be classified more by the

classified. I used Synthetic Minority Over-sampling Technique (SMOT) to create class 0 instances and remove the imbalance from the data.

I used K-Fold cross validation because it generally results in a less biased model compare to train-test-split method. Because it ensures that every observation from the original dataset has the chance of appearing in training and test set. This gives best performance if we have limited data.

KNN Confusion Metrics:



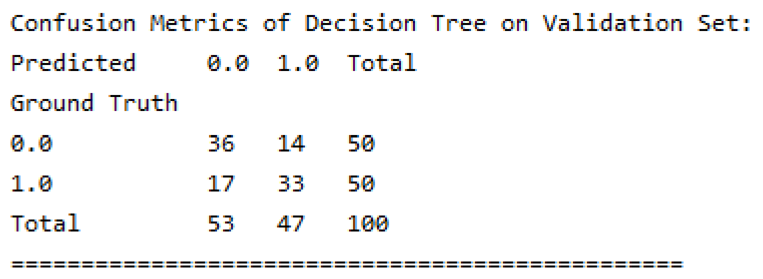
```
Confusion Metrics of KNN on Validation Set:
Predicted      0.0  1.0  Total
Ground Truth
0.0            37   13   50
1.0            16   34   50
Total          53   47   100
================================================
```

Interpretation:

Correct classification is along the diagonal which is 37 for class 0 and 34 for class 1. Misclassification is along the off diagonal which is 13 for class 0 and 16 for class 1. Class 0 was classified 53 times instead of 50 that shows the model is a little biased towards class 0

The validation set accuracy is 37 + 34 = 71%

Decision Tree Confusion Metrics:



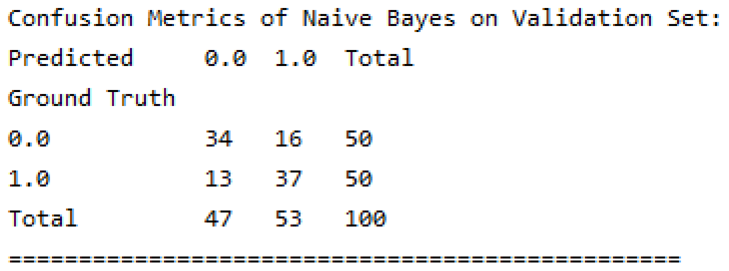
```
Confusion Metrics of Decision Tree on Validation Set:
Predicted      0.0  1.0  Total
Ground Truth
0.0            36   14   50
1.0            17   33   50
Total          53   47   100
================================================
```

Interpretation:

Correct classification is along the diagonal which is 36 for class 0 and 33 for class 1. Misclassification is along the off diagonal which is 14 for class 0 and 17 for class 1. Class 0 was classified 53 times instead of 50 that shows the model is a little biased towards class 0

The validation set accuracy is 36 + 33 = 69%

Naïve Bayes Confusion Metrics:



```
Confusion Metrics of Naive Bayes on Validation Set:
Predicted     0.0  1.0  Total
Ground Truth
0.0           34   16   50
1.0           13   37   50
Total         47   53   100
================================================
```

Interpretation:

Correct classification is along the diagonal which is 34 for class 0 and 37 for class 1.

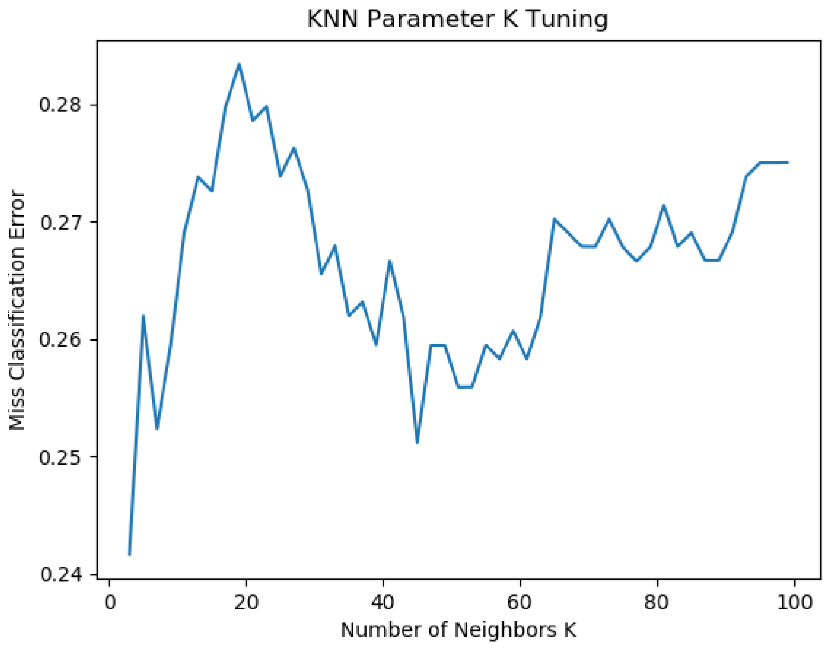
Misclassification is along the off diagonal which is 16 for class 0 and 13 for class 1.

Class 1 was classified 53 times instead of 50 that shows the model is a little biased towards class 1

The validation set accuracy is 34 + 37 = 71%

- ## Classifier Hyper Parameters Tuning:

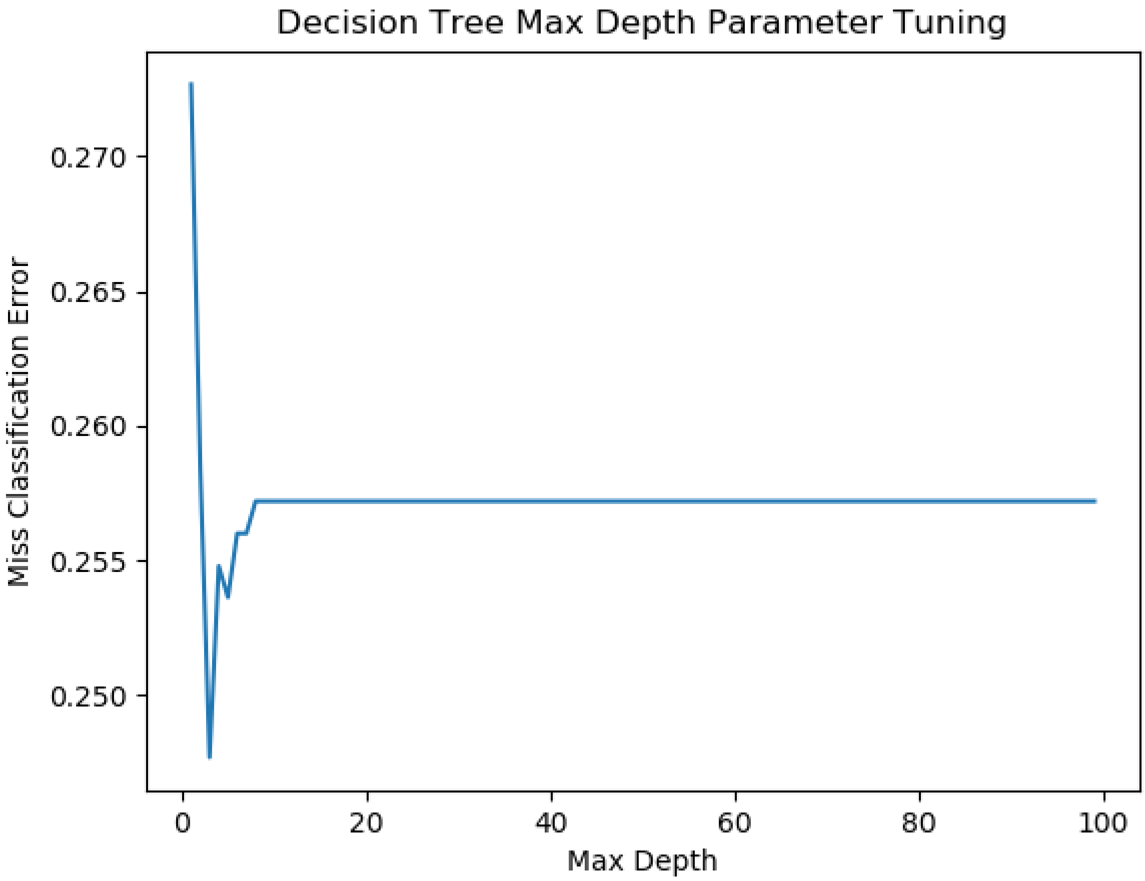
For **KNN** the parameter k (which is the number of nearest neighbors) was tuned to get better performance. The following graph shows the effect of k on misclassification error. The value of k that has the minimum error was chosen as optimal k which 3.



Decision Tree Hyper Parameters Tuning:

Decision tree uses a lot of hyper parameters. I experimented with minimum samples split (which is the minimum number of samples to be split), minimum samples leaf (which is the minimum number of leaves of a sample split) and maximum depth (which is the maximum number of levels of the tree). The following table shows that there was no major effect of minimum samples split on accuracy, therefore, that was left on default value. But a value of 18 for minimum samples leaf and optimal depth of 3 levels gave me better performance.

The tuning data is shown in the table and graph.



Decision Tree Parameters Tuning Table

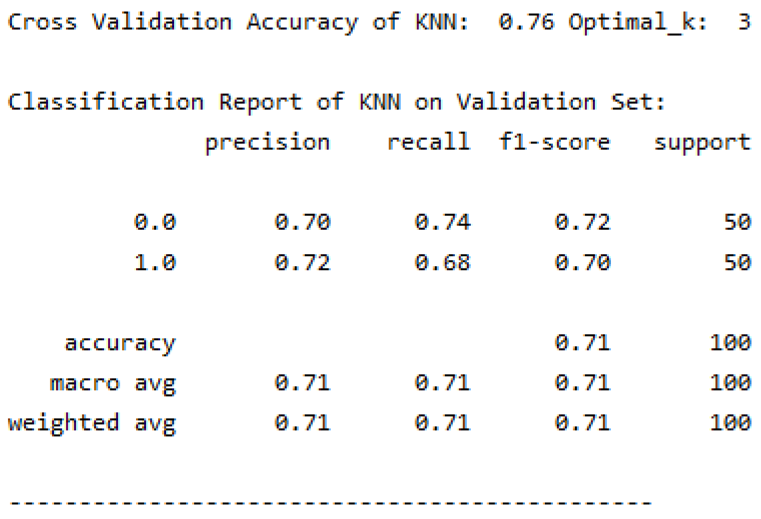
Minimum Samples Split Parameter	**Estimated Accuracy**	**Validation Set Accuracy**	**Estimated Accuracy Error**
2	0.76	0.60	0.16
4	0.76	0.60	0.16
6	0.76	0.67	0.09
8	0.77	0.58	0.19
10	0.77	0.56	0.21
15	0.77	0.58	0.19
20	0.75	0.62	0.13
25	0.75	0.63	0.12
30	0.75	0.63	0.12
Minimum Samples Leaf Parameter	**Estimated Accuracy**	**Validation Set Accuracy**	
2	0.76	0.55	0.21
4	0.76	0.55	0.21
6	0.76	0.63	0.13
8	0.76	0.63	0.13
10	0.76	0.67	0.09
15	0.75	0.63	0.12
17	0.76	0.63	0.13
18	0.75	0.69	0.06
19	0.75	0.68	0.07
20	0.75	0.68	0.07
25	0.76	0.67	0.09
30	0.76	0.69	0.07

- ## Classifier Comparison:

We can interpret the Precision as "How useful the results are" and Recall as "How complete the results are" [1].

F1-measure or F1-score is the harmonic mean of precision and recall.

KNN Classification Report:



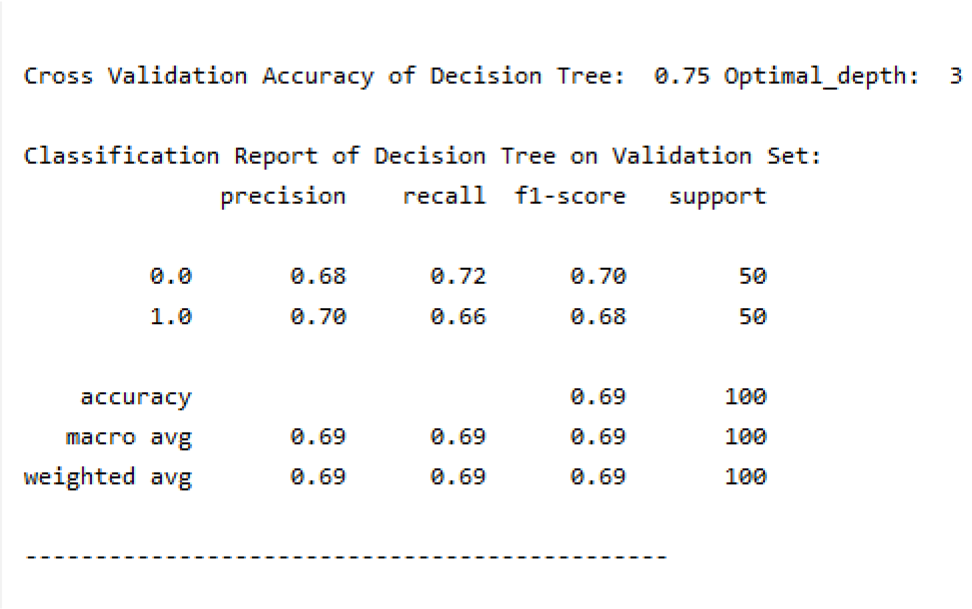
```
Cross Validation Accuracy of KNN:  0.76 Optimal_k:  3

Classification Report of KNN on Validation Set:
              precision    recall  f1-score   support

         0.0       0.70      0.74      0.72        50
         1.0       0.72      0.68      0.70        50

    accuracy                           0.71       100
   macro avg       0.71      0.71      0.71       100
weighted avg       0.71      0.71      0.71       100

----------------------------------------------
```

Report Interpretation:

As we can see from above figure that KNN's classification results are 71% useful or precise and 71% complete.

Its validation set accuracy is 0.71 but its cross-validation accuracy was 0.76. Therefore, its estimate of accuracy is within ±5 % (76 - 71 = 5)

Decision Tree Classification Report:



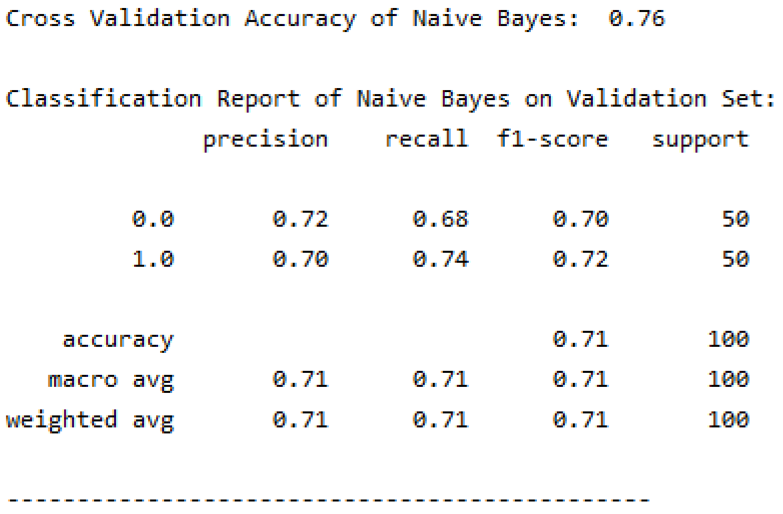
```
Cross Validation Accuracy of Decision Tree:  0.75 Optimal_depth:  3

Classification Report of Decision Tree on Validation Set:
              precision    recall  f1-score   support

         0.0       0.68      0.72      0.70        50
         1.0       0.70      0.66      0.68        50

    accuracy                           0.69       100
   macro avg       0.69      0.69      0.69       100
weighted avg       0.69      0.69      0.69       100

----------------------------------------------
```

Report Interpretation:

As we can see from above figure that Decision Tree classification results are 69% useful or precise and 69% complete.

Its validation set accuracy is 0.69 but its cross-validation accuracy was 0.75. Therefore, its estimate of accuracy is within ±6 % (75 - 69 = 6)

Naïve Bayes Classification Report:



```
Cross Validation Accuracy of Naive Bayes:  0.76

Classification Report of Naive Bayes on Validation Set:
              precision    recall  f1-score   support

         0.0       0.72      0.68      0.70        50
         1.0       0.70      0.74      0.72        50

    accuracy                           0.71       100
   macro avg       0.71      0.71      0.71       100
weighted avg       0.71      0.71      0.71       100

----------------------------------------------
```

Report Interpretation:

As we can see from above figure that Naïve Bayes classification results are 71% useful or precise and 71% complete.

Its validation set accuracy is 0.71 but its cross-validation accuracy was 0.76. Therefore, its estimate of accuracy is within ±5 % (76 - 71 = 5)

Conclusion:

Based on the F1-measure and Confusion Metrics analysis we can see that our best choice of two models could be KNN and Naïve Bayes. Because KNN is little biased towards class 0 and Naïve Bayes is little biased towards class 1 we can create ensemble of both of these models to remove the overall biasness.

- ## Prediction:

Test Set Predictions Table

ID	KNN PREDICTIONS	NAÏVE BAYES PREDICTIONS	ID	KNN PREDICTIONS	NAÏVE BAYES PREDICTIONS	ID	KNN PREDICTIONS	NAÏVE BAYES PREDICTIONS
1001	1	1	1023	1	1	1045	1	0
1002	0	0	1024	0	1	1046	0	1
1003	1	0	1025	0	0	1047	0	0
1004	0	0	1026	0	1	1048	1	0
1005	1	0	1027	0	0	1049	1	1
1006	1	0	1028	1	0	1050	0	0
1007	0	1	1029	1	1	1051	1	1
1008	0	1	1030	1	1	1052	1	0
1009	1	0	1031	1	1	1053	0	0
1010	0	1	1032	1	1	1054	1	1
1011	1	1	1033	1	1	1055	1	1
1012	0	1	1034	0	0	1056	1	1
1013	0	0	1035	0	0	1057	1	1
1014	1	1	1036	1	1	1058	1	1
1015	0	1	1037	0	0	1059	1	0
1016	1	0	1038	1	1	1060	0	0
1017	0	0	1039	0	1	1061	1	1
1018	1	0	1040	0	0	1062	0	1
1019	1	1	1041	1	1	1063	0	0
1020	0	0	1042	1	0	1064	1	1
1021	1	1	1043	1	1	1065	0	1
1022	1	1	1044	1	1	1066	1	0
1067	1	1	1079	0	1	1091	0	0
1068	0	1	1080	1	1	1092	0	1
1069	1	1	1081	1	1	1093	1	1
1070	0	1	1082	0	1	1094	1	0
1071	1	0	1083	0	1	1095	1	1
1072	1	0	1084	1	1	1096	1	0
1073	0	0	1085	1	0	1097	0	0
1074	1	1	1086	1	1	1098	0	0
1075	0	0	1087	0	1	1099	0	0
1076	1	1	1088	0	0	1100	1	1
1077	1	1	1089	1	1			
1078	1	1	1090	1	1			

References:

1 - https://en.wikipedia.org/wiki/Precision_and_recall