CSE 601: DATA MINING AND BIOINFORMATICS
FALL 2018

PROJECT 3: CLASSIFICATION ALGORITHMS REPORT

SIDDHARTH SELVARAJ #50247317
ALLEN DANIEL YESA #50246827
ANUSH RAVINDRA SHETTY #50247204

1.KNN

K nearest neighbors is a simple algorithm that stores all training data and classifies new test data based on a similarity measure like Euclidian distance.

The classification of a record is based on the neighbors.

A) ALGORITHM IMPLEMENTATION:

1. In this we first read the input data and split the data into k fold using cross validation technique which is 90% training data and 10% testing data
2. This technique is applied k times; in our code we have used it 10 times
3. Further we choose the best k neighbors for better performance and accuracy.
4. One good mechanism is to either sort the data to get k minimum value or use a heap based mechanism with a size k. In our code we used the heap based technique which will reduce the time complexity
5. For calculating the distance between two data points we have used Euclidean distance and through this we choose k nearest neighbors.
6. Repeating step 4 and 5 over all the test data will provide us with a predicted test value.

B) CHOOSING K:

1. A good technique for obtaining better accuracy and precision value is to select a better k
2. If we choose the k-value too small, then only limited set of points will be considered thereby not giving an accurate information
3. If we choose high value of k then we would consider many possibilities of data which would include outliers and would misclassify our data.
4. Hence depending on the data we would require to reiterate the data with different values of k.

C) CALCULATING EUCLIDIAN DISTANCE:

1. For calculation of k neighbors we first would require to calculate the distance between two data points.
2. We do this using Euclidean distance which is calculated as follows:
3. $dist((x, y), (a, b)) = \sqrt{(x - a)^2 + (y - b)^2}$
4. For categorical data, we calculate the Euclidean distance on the basis of their equality, i.e. if we have a data1 = ‘Present’ and data2=’Absent’ and if the two data points we are about to calculate are different, we put the distance as 1, otherwise we add the distance as 0.

D) RESULT ANALYSIS:

Dataset Used: project3_dataset1.txt

```
K=5
Average Accuracy: 92.96679197994987
Average Precision: 92.48897840674618
Average Recall: 87.83455119194744
Average F_measure: 0.9001336687970283
```

```
K=10
Average Accuracy: 93.1422305764411
Average Precision: 94.48798814878275
Average Recall: 86.41052127142385
Average F_measure: 0.9020575957077037

K=15
Average Accuracy: 92.96679197994987
Average Precision: 94.27721450089872
Average Recall: 86.3319502535739
Average F_measure: 0.9007911340953305

K=30
Average Accuracy: 92.44047619047619
Average Precision: 95.80379362988059
Average Recall: 83.11844262830155
Average F_measure: 0.8884920037638562

K=50
Average Accuracy: 91.56015037593986
Average Precision: 96.00605930017696
Average Recall: 80.14913963375312
Average F_measure: 0.8725870544779047
```

K	Accuracy	Precision	Recall	F1 Measure
5	92.96679197994987	92.48897840674618	87.83455119194744	0.9001336687970283
10	93.1422305764411	94.48798814878275	86.41052127142385	0.9020575957077037
15	92.96679197994987	94.27721450089872	86.3319502535739	0.9007911340953305
30	92.44047619047619	95.80379362988059	83.11844262830155	0.8884920037638562
50	91.56015037593986	96.00605930017696	80.14913963375312	0.8725870544779047

Dataset Used: project3_dataset2.txt

```
K=5
Average Accuracy: 62.1322849213691
Average Precision: 45.45787545787546
Average Recall: 30.149866215655692
Average F_measure: 0.35075237357978345

K=10
Average Accuracy: 65.59204440333026
Average Precision: 52.82967032967033
Average Recall: 18.301858229489806
Average F_measure: 0.26457129610043867
```

```
K=15
Average Accuracy: 67.32192414431083
Average Precision: 55.19047619047619
Average Recall: 26.895786961576437
Average F_measure: 0.3551006906899356

K=30
Average Accuracy: 65.56891766882516
Average Precision: 47.64285714285714
Average Recall: 15.306493944651839
Average F_measure: 0.22578739018769003

K=50
Average Accuracy: 66.44310823311747
Average Precision: 63.0
Average Recall: 10.46515984015984
Average F_measure: 0.17519505284951511
```

K	Accuracy	Precision	Recall	F1 Measure
5	62.1322849213691	45.45787545787546	30.149866215655692	0.35075237357978345
10	65.59204440333026	52.82967032967033	18.301858229489806	0.26457129610043867
15	67.32192414431083	55.19047619047619	26.895786961576437	0.3551006906899356
30	65.56891766882516	47.64285714285714	15.306493944651839	0.22578739018769003
50	66.44310823311747	63.0	10.46515984015984	0.17519505284951511

- Using K-NN we can say that the accuracy and precision for dataset1 is higher than that in dataset2.
- Also since dataset1 contains only continuous values and dataset2 contains both continuous and categorical values, this implies that kNN would work better on just continuous data rather than combination of data
- Also for and optimal value of k we can have better accuracy

E) ADVANTAGES:

- Easy to implement
- Suitable for handling both continuous and discrete data
- Robust to noisy data
- Boundaries can be of arbitrary shape

F) DISADVANTAGES:

- Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
- Classifying unknown records are relatively expensive
- Choosing the value of K- If k is too small, sensitive to noise points/ If k is too large, neighborhood may include points from other classes

2.DECISION TREE

A decision tree is a predictive model which builds classification or regression models in the form of a tree based on branching a series of Boolean tests that use specific facts to make conclusions. They can handle both categorical and numerical data. The decision tree consists of the decision nodes and the leaf nodes. The decision nodes have branches based on condition and the leaf nodes represent the classification.

A) ALGORITHM IMPLEMENTATION:

1. The algorithm is designed for both numerical and categorical input variables. The categorical variables when detected are converted to numerical values.
2. The dataset is split into 10 data chunks for the 10-fold cross validation and the respective train data and test data is determined.
3. The split point is determined using the function **get_split_point()** which uses the gini index that has been calculated from the **gini_calculation()** function to get the split point.
4. The decision tree is built by creating the root node and calling the **build_decision_tree()** function that then calls itself recursively to build the entire tree. The **classify_node()** function is used to get the classification at the node.
5. The test data is then classified based on the decision tree using the **prediction()** function which recursively calls itself with left and right nodes depending on the split.
6. From the predicted class and the actual class, the true positive, true negative, false positive and false negatives are determined using the **calc_measures()** function.
7. The accuracy, precision, recall and f1 score are calculated from the above determined measures.

B) CHOICE DESCRIPTION:

- **CATEGORICAL FEATURE:**

  If the current attribute is equal to the split value, it is appended to the left sub list else if the current attribute is not equal to the split value, it is appended to the right sub list.

- **CONTINUOUS FEATURE:**

  If the current attribute is not a categorical attribute and if it is less than or equal to the split value, it is appended to the left sub list else it is appended to the right sub list.

- **BEST FEATURE:**

  The feature which minimizes the gini index the most is the best feature.

- **POST PROCESSING:**

  The post processing of the decision tree is done as follows:

  1. For the fully grown decision tree, the nodes of the tree are trimmed in a bottom-up manner.

2. After trimming the tree, the sub-tree is replaced by a leaf node if the generalization error is improved.
3. The leaf node is classified based on the majority class of instances in the sub-tree.

C) RESULT ANALYSIS:

Dataset Used: project3_dataset1.txt

```
Enter the filename: project3_dataset1.txt
Accuracy: 92.25877192982455
Precision: 89.25079958740548
Recall: 91.46354511600778
F1 Measure: 0.9034362543936167
```

Dataset Used: project3_dataset2.txt

```
Enter the filename: project3_dataset2.txt
Accuracy: 61.87326549491213
Precision: 45.01894441987321
Recall: 50.46652397310292
F1 Measure: 0.4758733817922902
```

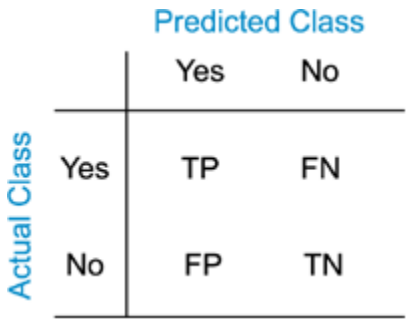
- Performance metrics of decision tree were comparable with other algorithms but it took lot more time than KNN and Naive Bays.

D) ADVANTAGES:

- Decision trees are easy to understand and interpret.
- Suitable for handling both continuous and discrete data.
- They are universal for solving classification and regression problems.
- Easy and inexpensive to construct.

E) DISADVANTAGES:

- Decision trees can be unstable even with a small change in the input data.
- It is complex and time-consuming to construct large decision trees with many branches.
- Needs to be pruned to avoid over-fitting.
- A highly complicated decision tree tends to have low bias.



The performance measures are calculated using the below formulas:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1 Measure = 2* (Precision * Recall) / (Precision + Recall)

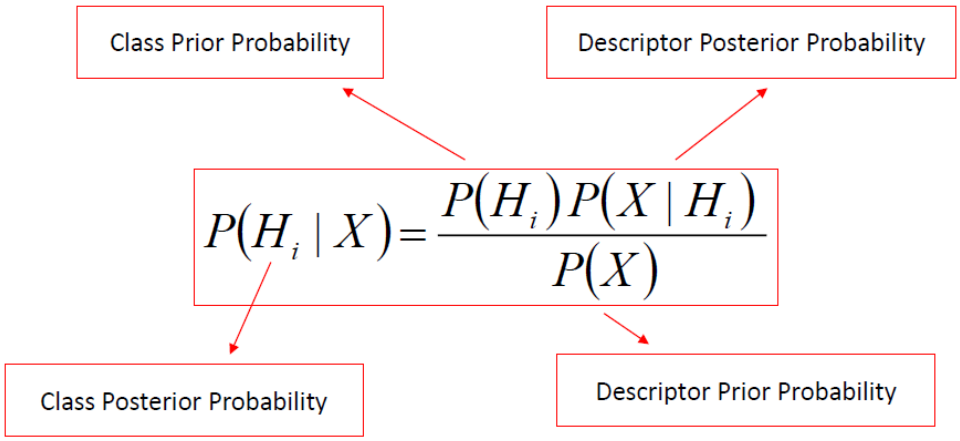
3.NAIVE BAYES

Naïve Bayes classifier is a probabilistic classifier based on Bayes theorem. Naïve Bayes assumes that the attributes are independent of each other

Based on Bayes theorem we estimate posterior probability

Bayes Theorem is as follows:

To classify means to determine the highest $P(H_i|X)$ among all classes C1, C2, …. Cm. Let X be a data sample whose class label is unknown. Let H_i be the hypothesis that X belongs to a particular class Ci. Calculate $P(H_i|X)$ using the Bayes theorem



A) ALGORITHM IMPLEMENTATION:

1. The algorithm takes in input as both continuous data and categorical data and would be applied to a k fold cross validation
2. The dataset is split into 10 data chunks for the 10-fold cross validation and the respective train data and test data is determined.
3. For the categorical training data we calculate the posterior probability using the above formula
4. For continuous training data we calculate the posterior probability using the formula
5. $$f(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
6. Further on the basis of training labels each of the count of each data is calculated
7. Using this count the descriptor posterior probability is calculated for the categorical data for respective classes (0 or 1)

B) HANDLING CONTINOUS FEATURES:

We have used following method for handling continuous features:

Probability density function:

1. Depending on the distribution of continuous values, we can estimate the posterior probability using the probability density function.
2. We calculate the standard deviation and mean of the data which is then applied to the PDF
3. In the below figure σ is the standard deviation and μ is the mean of the entire distribution, x is the current value in the distribution for which PDF is calculated.

C) HANDLING ZERO PROBABLITY:

1. As mentioned in the slide, the descriptor posterior probability may go to 0 if any of the probability goes to 0:

$$P(X \mid H_i) = \prod_{j=1}^{d} P(x_j \mid H_i)$$

2. This can be used in the following way:
   For example: If we have a dataset with 100 tuples for a class C, age = low(12), age = medium(50) and age= high(70)
3. Then by using the laplacian correction: we add 1 to each of the case
   Hence Prob(age=low|H) = 1/103
   Prob(age = medium|H) = 51/103
   Prob(age=high|H)=71/103

D) RESULT ANALYSIS:

Dataset Used: project3_dataset1.txt

```
Average Accuracy: 93.31453634085214
Average Precision: 91.31925736801898
Average Recall: 90.44426356826324
Average F_measure: 0.9078725938618137
```

Dataset Used: project3_dataset2.txt

```
Average Accuracy: 67.7659574468085
Average Precision: 53.11920462074727
Average Recall: 68.06706962628016
Average F_measure: 0.5906398590355034
```

- Using Naive Bayes we can see that the classifier is better off with the dataset1 as compared to dataset2. Further as the dataset1 contains continuous value and dataset 2 contains continuous and categorical value which might increase the runtime of the overall algorithm.
- Naive bayes impacts the accuracy, precision, recall and f-measure as for multi nomial attributes since the values of dataset would be more dependent.

E) ADVANTAGES:

- Efficient when applied to large databases
- Comparable in performance to decision trees
- Suitable for handling both continuous and discrete data
- They are suitable for large size of data

F) DISADVANTAGES:

- Attributes are not always independent they are correlated.
- It will not work if there are new labels in test which is not present in train

4. RANDOM FOREST

Random Forest is an extension of decision tree. It is used to classify more accurately than decision trees by using more than one decision tree.

It is an ensemble learning technique which uses k classifiers (decision trees) and classify the data based on the majority voting among all the trees.

A) ALGORITHM IMPLEMENTATION:

1. The file name and number of trees to be grown are read as input from the user.
2. The algorithm is designed for both numerical and categorical input variables. The categorical variables when detected are converted to numerical values.
3. The dataset is split into 10 data chunks for the 10-fold cross validation and the respective train data and test data is determined.
4. For each tree we choose 20% of the features from the given feature set.
5. For each tree we choose training set of size of the given training set but with replacement so the train data can repeat.
6. Then we feed the test data to the trees and get the classification and we do the final classification based on the majority voting.

B) RESULT ANALYSIS:

Dataset Used: project3_dataset1.txt

```
Enter the filename: project3_dataset1.txt
Enter the number of trees: 2
Accuracy: 92.43734335839598
Precision: 89.05215573500915
Recall: 90.26279753546481
F1 Measure: 0.8965338982166601

Enter the filename: project3_dataset1.txt
Enter the number of trees: 3
Accuracy: 93.14223057644111
Precision: 89.60397659167683
Recall: 91.71701516681303
F1 Measure: 0.90648183647835
```

```
Enter the filename: project3_dataset1.txt
Enter the number of trees: 4
Accuracy: 94.01629072681705
Precision: 91.1665512426382
Recall: 93.43905561998129
F1 Measure: 0.9228881610927547


Enter the filename: project3_dataset1.txt
Enter the number of trees: 5
Accuracy: 95.06578947368422
Precision: 94.30372073177946
Recall: 92.19773332144423
F1 Measure: 0.9323883654834452

Enter the filename: project3_dataset1.txt
Enter the number of trees: 6
Accuracy: 95.41979949874687
Precision: 94.20358868184955
Recall: 94.02624585612787
F1 Measure: 0.941148337262339

Enter the filename: project3_dataset1.txt
Enter the number of trees: 7
Accuracy: 95.95238095238095
Precision: 95.24482417224351
Recall: 93.90716720479115
F1 Measure: 0.9457126582511609
```

Number of trees	Accuracy	Precision	Recall	F1 Measure
2	92.43734335839598	89.05215573500915	90.26279753546481	0.8965338982166601
3	93.14223057644111	89.60397659167683	91.71701516681303	0.90648183647835
4	94.01629072681705	91.1665512426382	93.43905561998129	0.9228881610927547
5	95.06578947368422	94.30372073177946	92.19773332144423	0.9323883654834452
6	95.41979949874687	94.20358868184955	94.02624585612787	0.941148337262339
7	95.95238095238095	95.24482417224351	93.90716720479115	0.9457126582511609

Dataset Used: project3_dataset2.txt

```
Enter the filename: project3_dataset2.txt
Enter the number of trees: 2
Accuracy: 54.972247918593894
Precision: 37.9803578774167
Recall: 44.02288208867156
F1 Measure: 0.407789940303458
```

```
Enter the filename: project3_dataset2.txt
Enter the number of trees: 3
Accuracy: 62.331174838112865
Precision: 44.924558284852395
Recall: 38.531856301593145
F1 Measure: 0.41483369081269416

Enter the filename: project3_dataset2.txt
Enter the number of trees: 4
Accuracy: 62.331174838112865
Precision: 46.25598086124402
Recall: 38.16412461807199
F1 Measure: 0.4182224146479407

Enter the filename: project3_dataset2.txt
Enter the number of trees: 5
Accuracy: 62.756706753006476
Precision: 46.07463125110184
Recall: 39.73654196680513
F1 Measure: 0.4267151816384724

Enter the filename: project3_dataset2.txt
Enter the number of trees: 6
Accuracy: 65.80943570767808
Precision: 52.33943833943834
Recall: 39.618312973576124
F1 Measure: 0.45098976853723655

Enter the filename: project3_dataset2.txt
Enter the number of trees: 7
Accuracy: 63.38575393154486
Precision: 46.375851573606994
Recall: 48.88178561204877
F1 Measure: 0.475958569028755
```

Number of trees	Accuracy	Precision	Recall	F1 Measure
2	54.972247918593894	37.9803578774167	37.9803578774167	0.407789940303458
3	62.331174838112865	44.924558284852395	38.531856301593145	0.41483369081269416
4	62.331174838112865	46.25598086124402	38.16412461807199	0.4182224146479407
5	62.756706753006476	46.07463125110184	39.73654196680513	0.4267151816384724
6	65.80943570767808	52.33943833943834	39.618312973576124	0.45098976853723655
7	63.38575393154486	46.375851573606994	48.88178561204877	0.475958569028755

- We have run the random forests for the given 2 datasets which produces the results which are shown above
- We have varied the number of trees from 2 to 7
- We can see that the accuracy also tends to increase as the number of trees increases

C) ADVANTAGES:

- Improves accuracy: Incorporate more diversity and reduce variances
- Improve efficiency: Searching among subsets of features is much faster than searching among the complete set
- Suitable for handling both continuous and discrete data
- They are suitable for large size of data

D) DISADVANTAGES:

- Too many classes so misclassification is possible
- It is very complex compared to decision tree because more trees are involved
- Slower and less accurate compared to boosting

5. BOOSTING

Boosting is an ensemble method for improving the model predictions of any given learning algorithm by converting the weak learners to strong learners by making currently misclassified records more important.

AdaBoost is a type of "Ensemble Learning" where multiple learners are employed to build a stronger learning algorithm. AdaBoost works by choosing a decision tree algorithm and iteratively improving it by accounting for the incorrectly classified examples in the training set.

A) ALGORITHM IMPLEMENTATION:

1. The file name and number of learners are read as input from the user.
2. The algorithm is designed for both numerical and categorical input variables. The categorical variables when detected are converted to numerical values.
3. The dataset is split into 10 data chunks for the 10-fold cross validation and the respective train data and test data is determined.
4. Initially we set uniform weights on all the records.
5. For each learner we do the classification on the training dataset and we calculate the error
6. If the error is greater than 50% we reject the classifier
7. We update the weights based on the classification
8. Records that are wrongly classified will have their weights increased
9. Records that are classified correctly will have their weights decreased
10. We implement it on the test data
11. Final prediction is weighted average of all the classifiers

B) RESULT ANALYSIS:

Dataset Used: project3_dataset1.txt

```
Enter the filename: project3_dataset1.txt
Enter the number of learners: 3
Accuracy: 94.01629072681703
Precision: 90.09683995291657
Recall: 93.97845613091879
F1 Measure: 0.9199672200655641

Enter the filename: project3_dataset1.txt
Enter the number of learners: 5
Accuracy: 95.07832080200501
Precision: 93.34612206392391
Recall: 93.54600870197767
F1 Measure: 0.9344595849062763

Enter the filename: project3_dataset1.txt
Enter the number of learners: 7
Accuracy: 96.13095238095238
Precision: 95.91581153952245
Recall: 93.36792383946083
F1 Measure: 0.946247195396014
```

Number of leaners	Accuracy	Precision	Recall	F1 Measure
3	94.01629072681703	90.09683995291657	93.97845613091879	0.9199672200655641
5	95.07832080200501	93.34612206392391	93.54600870197767	0.9344595849062763
7	96.13095238095238	95.91581153952245	93.36792383946083	0.946247195396014

Dataset Used: project3_dataset2.txt

```
Enter the filename: project3_dataset2.txt
Enter the number of learners: 3
Accuracy: 60.59204440333025
Precision: 44.026071967248434
Recall: 38.38425901583796
F1 Measure: 0.41012046176288685

Enter the filename: project3_dataset2.txt
Enter the number of learners: 5
Accuracy: 62.54394079555967
Precision: 45.66919844861021
Recall: 37.29794401504927
F1 Measure: 0.4106124801621394
```

```
Enter the filename: project3_dataset2.txt
Enter the number of learners: 7
Accuracy: 64.96762257169287
Precision: 49.2979242979243
Recall: 43.15540599751126
F1 Measure: 0.4602261337938049
```

Number of leaners	Accuracy	Precision	Recall	F1 Measure
3	60.59204440333025	44.026071967248434	38.38425901583796	0.41012046176288685
5	62.54394079555967	45.66919844861021	37.29794401504927	0.4106124801621394
7	64.96762257169287	49.2979242979243	43.15540599751126	0.4602261337938049

- We have run the boosting algorithm for the given 2 datasets which produces the results which are shown above
- We have varied the number of classifiers – 3,5 and 7.
- From the results we can see that with the increase in number of classifiers the algorithm does a better job because the accuracy increases
- The time taken also increases with increase in number of classifiers

C) ADVANTAGES:

- It uses many classifiers so gives better results compared to individual classifier
- Suitable for handling both continuous and discrete data
- They can handle huge amount of data

D) DISADVANTAGES:

- Boosting is the most complex classification algorithm
- It involves more computations like updating the weights
- It is affected by noise and outliers

CROSS VALIDATION:

Cross-validation is the technique of evaluating the predictive models by partitioning the original dataset into a training set to train the model, and a test set to evaluate it.

Here 10-fold cross validation has been implemented which splits the dataset into 10 partitions and uses 9 subsets for the train data and 1 subset for the test data.

The accuracy, precision, recall and f1 measure are calculated for each fold and their averages are determined at the last as the final performance measures.