Introduction To Data Mining Project 1

Sri Krishna Vamsi Koneru

5881358

koner033@umn.edu

Experiment 1:

1) There are 3333 rows which correspond to 3333 records in the dataset
2) There are 21 features/columns in the dataset:
    1) State: Discrete
    2) Account Length: Discrete
    3) Area Code: Discrete
    4) Phone number: Discrete
    5) International Plan: Binary
    6) Voice mail plan: Binary
    7) Number vmail messages: Discrete
    8) Total day minutes: Continuous
    9) Total day calls: Discrete
    10) Total day charge: Continuous
    11) Total eve minutes: Continuous
    12) Total eve calls: Discrete
    13) Total Eve charge: Continuous
    14) Total night minutes: Continuous
    15) Total night calls: Discrete
    16) Total night charge: Continuous
    17) Total intl minutes: Continuous
    18) Total intl calls: Discrete
    19) Total intl charge: Continuous
    20) Customer service calls: Discrete
    21) Churn: Binary
3) The irrelevant attributes are: Phone number, area code
   Because phone number has no valuable information as it is just used to identify. Area code is also something which might be considered as an irrelevant attribute. Account length cannot be an irrelevant attribute if it is the number of days the accountholder has held an account, else if it is just a random length value like then it can be irrelevant. Area code can also be ignored as we are including state attribute in our dataset and there is no proper trends that can be observed by adding the area code feature.
4) No there are no missing values as all features have 3333 records.
5)

	Total Day minutes	Total day charge	Total eve minutes	Total eve charge	Total night minutes	Total night charge	Total intl minutes	Total intl charge
Average	179.78	30.56	200.98	17.08	200.87	9.04	10.24	2.76
Median	179.40	30.50	201.40	17.12	201.20	9.05	10.30	2.30
Maximum	350.80	59.64	363.70	30.91	395.00	17.77	20.00	5.40
Minimum	0.00	0.00	0.00	0.00	23.20	1.04	0.00	0.00
Standard Deviation	54.47	9.26	50.71	4.31	50.57	2.28	2.79	0.75

6) 1.56 is the average number of customer service calls
7) 51
8) It is skewed as 86% of the records have churn value=False therefore it is skewed towards loyal customers.
9) Max day charge is 59.64 while minimum is 0.00. When sorted in descending order by total day charge, it is observed that the top 10 customers with the higher total day charge have all left the company now while 9 out of the 10 lowest total day charge customers are still loyal to the company.
10) The average number of customer service calls made by the customers who have left the company is 2.23 as opposed to the 1.45 average number of customer service calls made by the customers who are still loyal to the company. The average number is clearly higher for those who have left the company(churn=true).
11) There is not much difference in the total day calls, total night calls, total eve calls, total intl calls between both sets of customers however, among those who have left the averages for total day minutes, total night minutes, total eve minutes and total intl minutes are higher than the customers who are still loyal to the company. Also, all the average charges(day, eve, night, intl) are higher for the customers who have left the company which might be due to the increased minutes. Mainly, the customer service calls for churned out customers is almost 1.5 times the average of those who are still loyal to the company. As a data scientist, to retain the customers my suggestion to the company would be to lower the day charges prices as there is a considerable difference in that feature between churned out customers and the customers who are still loyal to the company. As they usually had higher data minutes, maybe considering to apply relatively lower charges as minutes cross a threshold will help in retaining them.
12) Accuracy:0.84, Precision:0.88, Recall:0.93
13)

P (churn = True | international plan = 'yes') = 0.42
P (churn = False | international plan = 'yes') = 0.57
P (churn= True | international plan = 'no') = 0.11
P (churn = False | international plan = 'no') = 0.88
P(international plan='yes'|churn=True)=0.284
P(international plan='no'|churn=True)=0.716
P(international plan='yes'|churn=False)=0.065

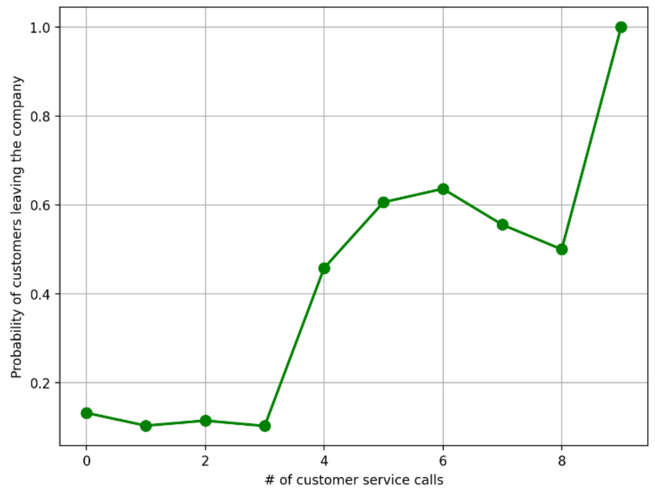
P(international plan='no'|churn=False)=0.935

14) So given that the customer has made 0 customer calls the probability of him leaving the company is 92/697 = 0.131

The probability of customer leaving given no.of calls can be observed in following table:

No.of customer service calls	Probability of customer leaving given those many calls
0	0.131
1	0.103
2	0.11
3	0.1
4	0.45
5	0.6
6	0.63
7	0.55
8	0.5
9	1

The probability of customers leaving company as customer service calls increase



15) In the case of international plan=yes and the customer service calls >3 then the customer will churn model the values are:

Accuracy=0.858

Precision=0.678

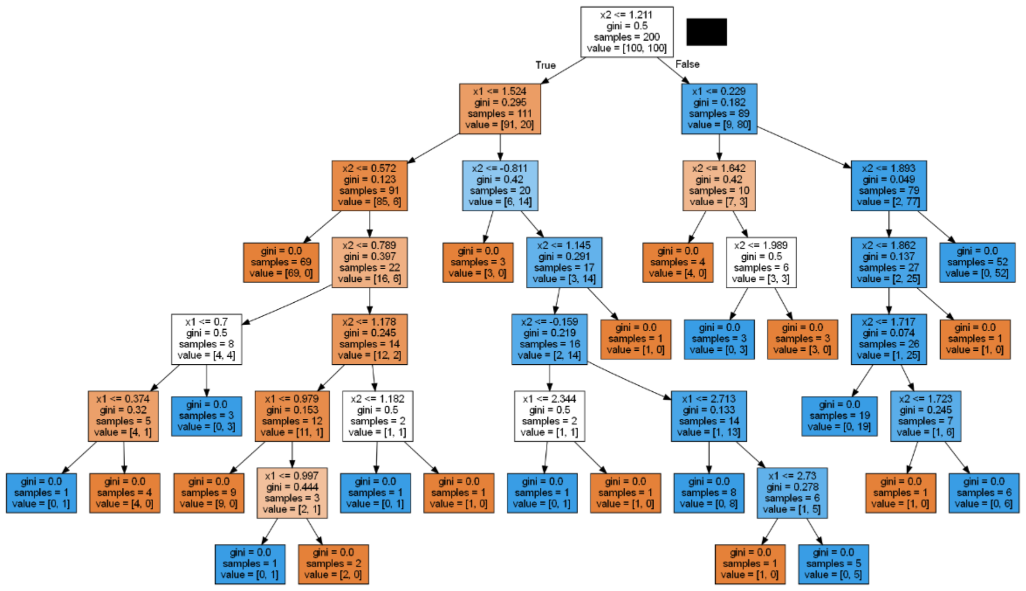
Recall=0.039

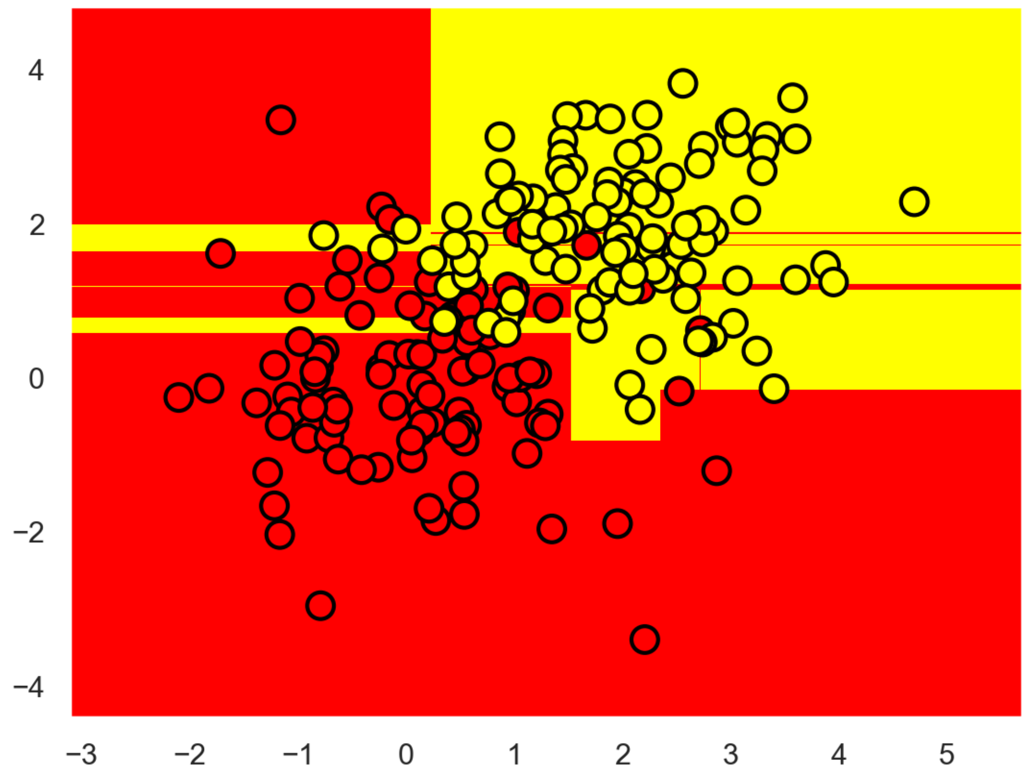
Experiment 2:

1) If we don't specify any criterion, it will consider and calculate gini index by default. If we don't give max depth, then it will keep expanding the tree until it reaches a state in which all leaf nodes have records belonging to only one class and the depth of the tree in that case is 7.

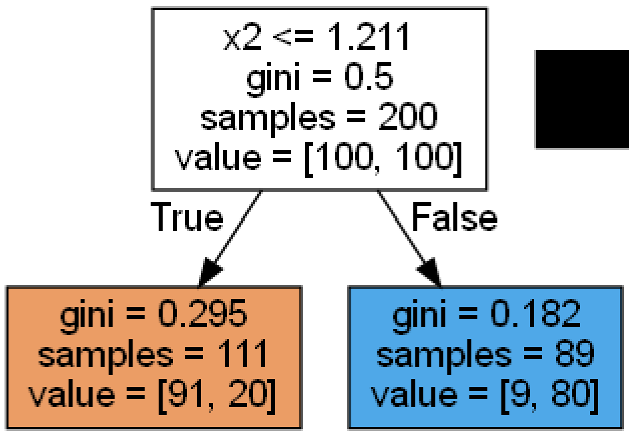
2) The boundary has a lot of rectangles when we overfit as opposed to the only 2 rectangles in the boundary when we underfit(only depth=1)

   When depth is not given / max depth=7,
   D.T is like

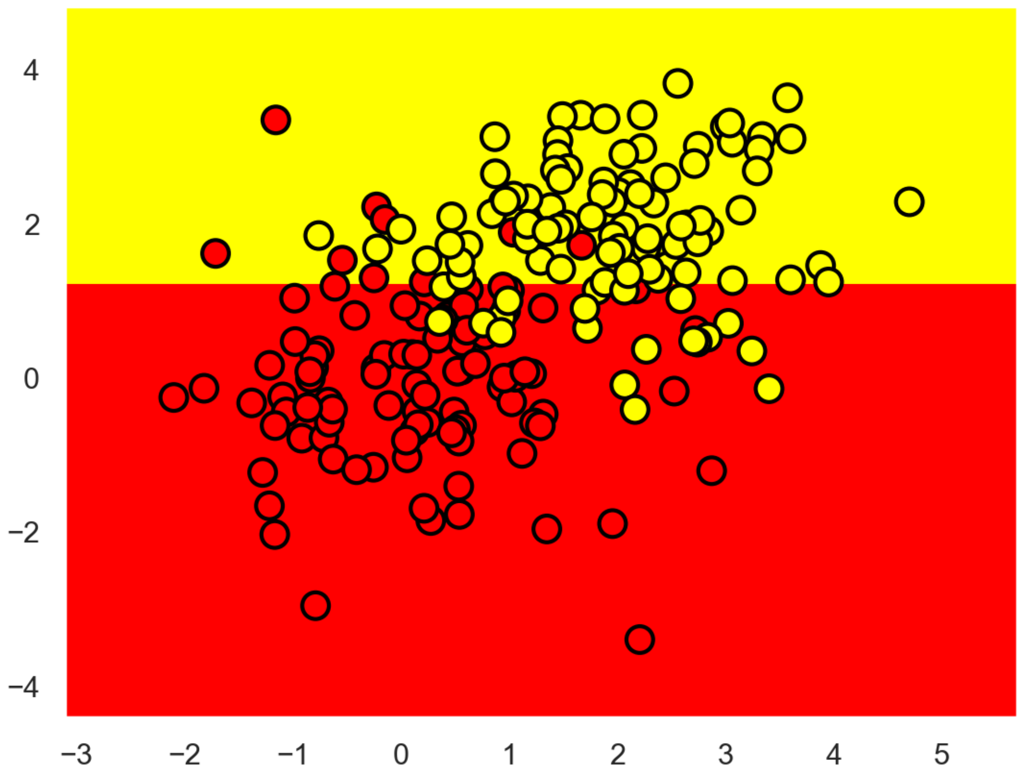

x2 <= 1.211 gini = 0.5 samples = 200 value = [100, 100]
True
False
x1 <= 1.524 gini = 0.295 samples = 111 value = [91, 20]
x1 <= 0.229 gini = 0.182 samples = 89 value = [9, 80]
x2 <= 0.572 gini = 0.123 samples = 91 value = [85, 6]
x2 <= -0.811 gini = 0.42 samples = 20 value = [6, 14]
x2 <= 1.642 gini = 0.42 samples = 10 value = [7, 3]
x2 <= 1.893 gini = 0.049 samples = 79 value = [2, 77]
gini = 0.0 samples = 69 value = [69, 0]
x2 <= 0.789 gini = 0.397 samples = 22 value = [16, 6]
gini = 0.0 samples = 3 value = [3, 0]
x2 <= 1.145 gini = 0.291 samples = 17 value = [3, 14]
gini = 0.0 samples = 4 value = [4, 0]
x2 <= 1.989 gini = 0.5 samples = 6 value = [3, 3]
x2 <= 1.862 gini = 0.137 samples = 27 value = [2, 25]
gini = 0.0 samples = 52 value = [0, 52]
x1 <= 0.7 gini = 0.5 samples = 8 value = [4, 4]
x2 <= 1.178 gini = 0.245 samples = 14 value = [12, 2]
x2 <= -0.159 gini = 0.219 samples = 16 value = [2, 14]
gini = 0.0 samples = 1 value = [1, 0]
gini = 0.0 samples = 3 value = [0, 3]
gini = 0.0 samples = 3 value = [3, 0]
x2 <= 1.717 gini = 0.074 samples = 26 value = [1, 25]
gini = 0.0 samples = 1 value = [1, 0]
x1 <= 0.374 gini = 0.32 samples = 5 value = [4, 1]
gini = 0.0 samples = 3 value = [0, 3]
x1 <= 0.979 gini = 0.153 samples = 12 value = [11, 1]
x2 <= 1.182 gini = 0.5 samples = 2 value = [1, 1]
x1 <= 2.344 gini = 0.5 samples = 2 value = [1, 1]
x1 <= 2.713 gini = 0.133 samples = 14 value = [1, 13]
gini = 0.0 samples = 19 value = [0, 19]
x2 <= 1.723 gini = 0.245 samples = 7 value = [1, 6]
gini = 0.0 samples = 1 value = [0, 1]
gini = 0.0 samples = 4 value = [4, 0]
gini = 0.0 samples = 9 value = [9, 0]
x1 <= 0.997 gini = 0.444 samples = 3 value = [2, 1]
gini = 0.0 samples = 1 value = [0, 1]
gini = 0.0 samples = 1 value = [1, 0]
gini = 0.0 samples = 1 value = [0, 1]
gini = 0.0 samples = 1 value = [1, 0]
gini = 0.0 samples = 8 value = [0, 8]
x1 <= 2.73 gini = 0.278 samples = 6 value = [1, 5]
gini = 0.0 samples = 1 value = [1, 0]
gini = 0.0 samples = 6 value = [0, 6]
gini = 0.0 samples = 1 value = [0, 1]
gini = 0.0 samples = 2 value = [2, 0]
gini = 0.0 samples = 1 value = [1, 0]
gini = 0.0 samples = 5 value = [0, 5]
[redacted]



When depth=1,



[redacted]



3) The 5 values used by the Decision Tree for age are: 43.5,19.0,22.5,30.0,32.0
   The significance is that at each point of time when using that particular age to split the records, one of the two childs was a leaf node where all records were belonging to only one class.

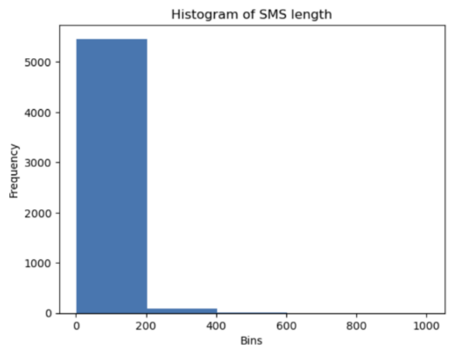
4) sklearn.preprocessing.StandardScaler is used to scale the features to a unit variance. of the dataset so that k-NN can function properly
   array([[-1., -1.],
   [-1., -1.],
   [1., 1.],
   [1., 1.]]) is the scaler transformation
5) In the section, the decision tree is trained on training data set and then tested using the test dataset. In the experiment we considered two decision trees where we limited depth to 2 for one and let it go till 8 for the other one. Then we trained each one separately on both the small and large datasets and we can observe the different accuracies. From the comparisions between the training and test accuracies, it is clear that in the case of training Large D.T on small dataset underfitting is happening(can be observed by difference in the test accuracy in both large D.Ts) and when training small D.T with big dataset then overfitting is happening(can be noticed due to the difference in training and test accuracies)
6) In this experiment, our goal is to compare and contrast the performance of decision trees when trained and tested using balanced and imbalanced datasets and compare the various measures in all 4 scenarios.
   In all 4 cases, the accuracy on training dataset is 1 and precision, recall , f1 score on the dataset used to train the model is 1.0,1.0,1.0
   However there are differences in the values for the test dataset as follows:

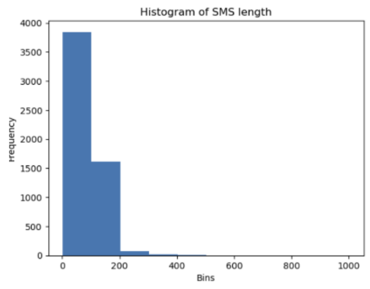
When trained on balanced dataset and tested on balanced data set the experiment's accuracy,f1 score,precision,recall are: 0.79,0.789,0.8,0.77
When trained on balanced dataset and tested on imbalanced dataset the experiment's accuracy,f1 score,precision,recall are:0.79,0.05,0.02,1.0
When trained on imbalanced dataset and tested on balanced dataset the experiment's accuracy, f1 score, precision, recall are:0.57,0.27,1.0,0.157
When trained on imbalanced dataset and tested on imbalanced dataset the experiment's accuracy, f1 score, precision, recall are:0.99,0.0,0.0,0.0
It is clear that the decision tree is performing better when trained on balanced dataset than when trained on imbalanced dataset. This can be observed by clearly observing the accuracies when tested on a imbalanced dataset despite being trained on balanced dataset the accuracy is same as when tested on balanced dataset. However, when trained on imbalanced dataset accuracy is very high when tested on imbalanced dataset but it is very low when tested on balanced dataset. This is due to the fact that the imbalanced dataset does not have a proper representation of the classes and therefore since it has instances of one class mainly the decision tree seems to be biased towards that class.

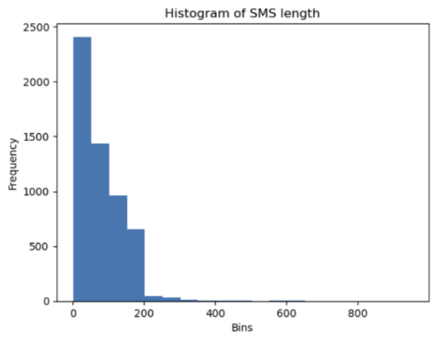
7) After adding the irrelevant attributes, the accuracy of the D.T decreases as opposed to the accuracy on the test set before adding those attributes. However, we know that Decision tree should be able to handle irrelevant attributes unless they are interacting. Here, since the accuracy is decreasing on the test dataset it means that the D.T was not able to handle them properly. Also there is a chance for another scenario, since training accuracy is 100% while test is decreasing there is a chance that due to increase in the number of attributes overfitting has occurred and since depth increased a increase in model complexity might have also happened which resulted in decrease in the test dataset accuracy.
to the fact that the irrelevant attributes that added were interacting attributes.
8) The accuracy is 94% when max depth=5 but when we leave at default the accuracy decreases to 92%. It is a 2% fall. This is because the complexity of the model increases as the depth increases and overfitting happens which results in a decrease in the accuracy as it classifies some more of the instances incorrectly.
9) Initially , when max_depth is [1-10] and features is [4-18], in the code the cross validation is 5 initially and therefore we get 5 folds for each of 126 candidates and thus 630 fits. However, when we change cross validation factor to 10 then there are 10 folds for all of the 126 candidates and thus we get 1260. When we ignore the cross validation and assume it doesn't happen, then we will have 126 Decision trees.
10) The best choice of k in k-NN when we use cross validation of 10 folds is 9. We get 10 fits for all the 9 possible candidates and total we get 90 fits. From them, the best parameter possible is having value of k as 9.
11) For MNIST dataset, the accuracy of the Decision Tree when max depth=5 is 0.666 and the accuracy for for K nearest neighbor when K=10 is 0.976. With 5 fold cross validation the best parameters for the Decision Tree are max depth 10 and max features 50 while the decision tree accuracy now is 0.84 which is more than the 66% initially.

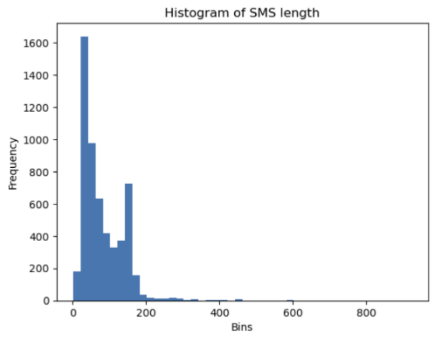
Experiment 3:

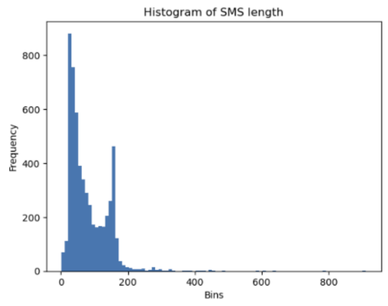
1. There are 5572 records. The label class distribution is skewed towards the label “ham”. The distribution is ham-4825 and spam-747.
2. There are 5169 unique messages in the dataset.The message that occurs most frequently is “Sorry, I’ll call later” and it occurs a total of 30 times.
3. The maximum length of the SMS is 910 and the minimum length is 2. As the bin size increases from 5 to 50 to 200 the number of bins decrease and the graph is more clear(pointwise the histogram is clearer when bin size is lower and as bin size increases it becomes smoother)
   The histograms for bin sizes 200,100,50,20,10,5 are as follows:

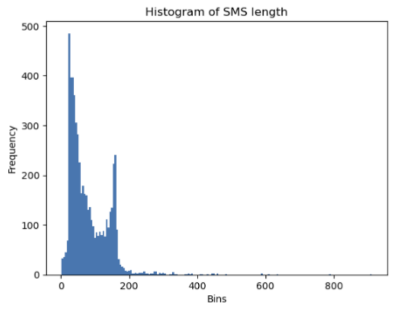








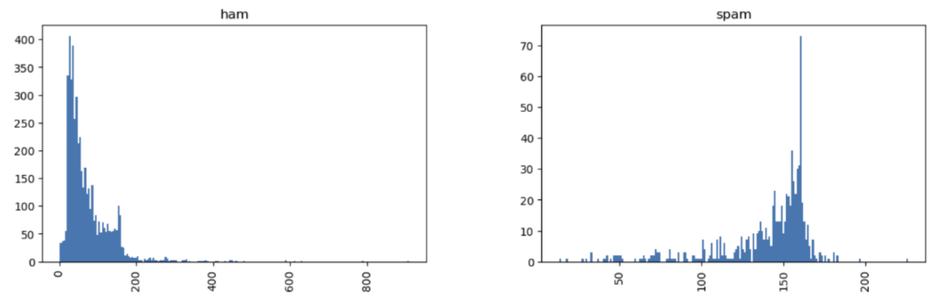




4. As bin size is increasing the graph sharpens and the points are clear as opposed to the above scenario. So, the histograms for the bin sizes 200,100,50,20,10,5 are:

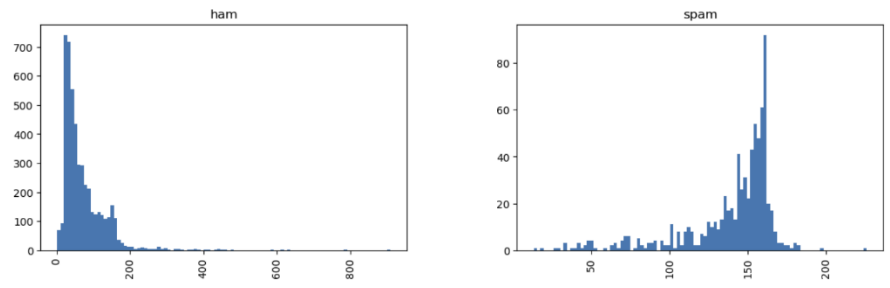
```
df_sms.hist(column='length', by='label', bins=200,figsize=(14,4))
```

```
array([<Axes: title={'center': 'ham'}>, <Axes: title={'center': 'spam'}>],
      dtype=object)
```



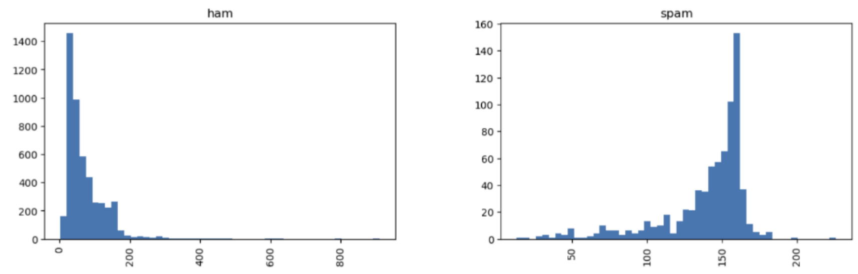
```
df_sms.hist(column='length', by='label', bins=100,figsize=(14,4))
```

```
array([<Axes: title={'center': 'ham'}>, <Axes: title={'center': 'spam'}>],
      dtype=object)
```



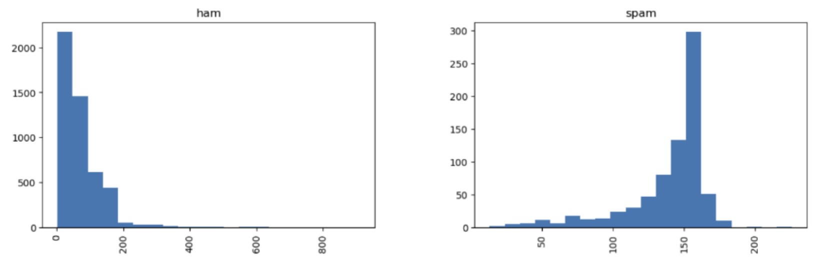
```
df_sms.hist(column='length', by='label', bins=50,figsize=(14,4))
```

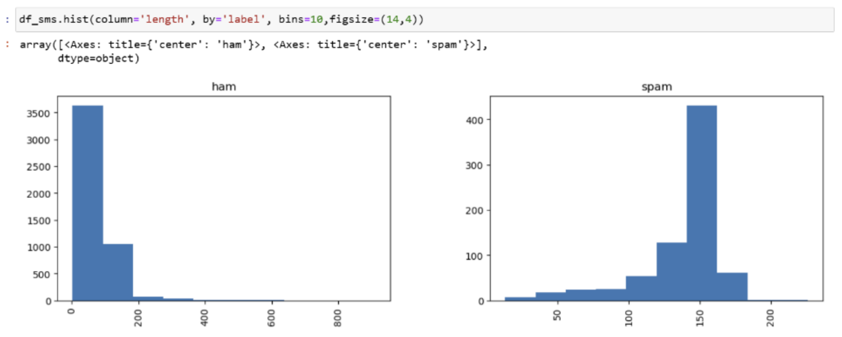
```
array([<Axes: title={'center': 'ham'}>, <Axes: title={'center': 'spam'}>],
      dtype=object)
```



```
df_sms.hist(column='length', by='label', bins=20,figsize=(14,4))
```

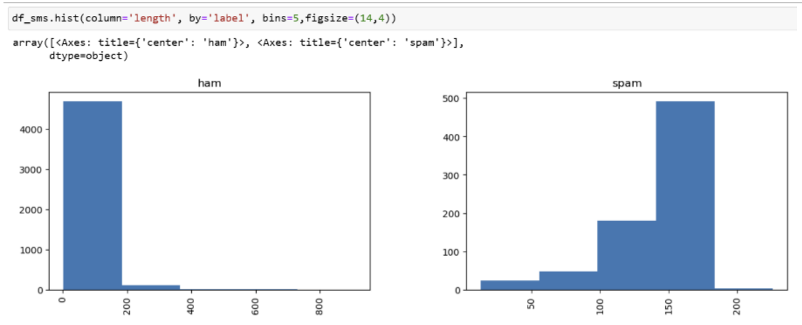
```
array([<Axes: title={'center': 'ham'}>, <Axes: title={'center': 'spam'}>],
      dtype=object)
```





```
df_sms.hist(column='length', by='label', bins=10,figsize=(14,4))
```

```
array([<Axes: title={'center': 'ham'}>, <Axes: title={'center': 'spam'}>],
      dtype=object)
```



```
df_sms.hist(column='length', by='label', bins=5,figsize=(14,4))
```

```
array([<Axes: title={'center': 'ham'}>, <Axes: title={'center': 'spam'}>],
      dtype=object)
```

5. We do that as a process of cleaning the text. It is important so that just because of case same word is not considered as two different words in the vocabulary. It is essential to decrease the vocabulary. Yes even if we convert all to upper case the approach should still work as it fulfills the original goal of ignoring the case.
6. CountVectorizer achieves the purpose of cleaning the data that converts all to lowercase and also token parameter to ensure punctuations are ignored .The stopwords parameter when set to English will remove all the words from our document set that match a list of English stop words which are defined in scikit-learn. 5 examples of the stop words are “ already,and,cannot,describe,each”
7. We first do data cleaning and pre processing using the count vectorizer. After this,we get all the feature names and then transform the occurrences into a to array and then transforming this to form a frequency matrix in the dataset.We first split it into training and test and then generate the document matrix by fitting the training dataset. Then, we transform the test data set and return the matrix
8. The document term matrix for the given documents using bag of words approach in a tabular representation is:

	Are	Call	From	Hi	Home	How	Money	Now	Or	Tomorrow	Win	you
0	1	0	0	1	0	1	0	0	0	0	0	1
1	0	1	1	0	1	0	1	1	0	0	2	0
2	0	1	0	1	0	0	0	1	1	1	0	1

9. 7777 features are created for the given data set. To reduce the features, we can use the stopwords=English parameter in countvectorizer, this can act in something similar to like removing irrelevant attributes from our dataset. While the pros include removal of usually non important words the con is that sometimes even the words we want might be ignored as part of this.

10. We use Multinomial Naïve Bayes for our dataset as multinomial is suitable for classification with discrete features(wordcount in our case) While Gaussian Naïve Bayes is better when dealing with continuous features.
    After applying the algorithm, for the spam class:
    Accuracy : 0.984
    Precision score: 0.942
    Recall score: 0.935
    F1 score: 0.939

Experiment 4:

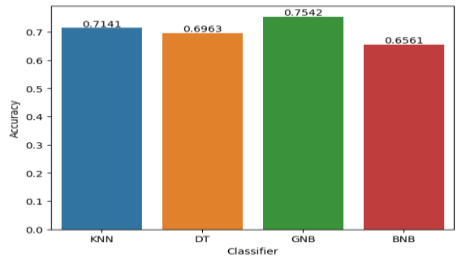
Analysis on Diabetes Data:

The diabetes data set has 9 features and 768 records. The features are Pregnancies, Glucose,BloodPressure, Skinthickness,Insulin,BMI, DiabetesPedigreeFunction,Age,Outcome. The dataset looks like it can be used to train a model to classify if a person can be at risk of having diabetes or not based on the other column values. The dataset however seems to not be balanced because there are 500 records with outcome zero as opposed to 268 with outcomes one. However it is not heavily imbalanced. Initially, histograms are used to observe the number of records for each column values. This provides an overview on the number of records for each column value. Later, we group them by outcomes(whether the record is of diabetes positive or negative) and then based on that we observe the trends between the features when we see the ones with outcome 0 and 1.

Also, there are no missing values as observed by the isnull().sum(). Then, we check for outliers by seeing the number of records that have unusual values(like 0) and then we see the number of recrords that have those values If the number of records having such values for the particular attribute is not too high then it means that theses are outliers and those records should be excluded. After doing this process for all the features , we observe the number of records now has been reduced to 724.

Now, we take all the 8 features as x value and the outcome(which is the class on which the model needs to classify) as y. We take four algorithms to train our various models. KNN,Decision Tree, Gaussian Naïve Bayes,Bernoulli Bayes. Before we being training, we first have to split our dataset into training and test dataset. Since the outcomes are not equally represented(there are almost double the 0s than 1s) so we use stratified split to make sure the dataset is equally represented and not skewed.

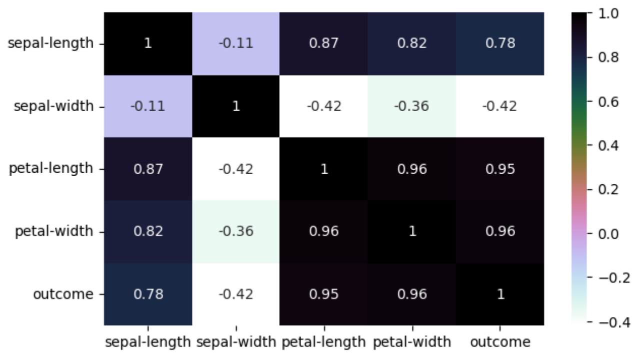
We predict the accuracies using all four algorithms, KNN and DT have same accuracy of 72.9% while Gaussian Naïve Bayes has an accuracy of 73.4% and Bernoulli has 65.7%. Now,we try k fold cross validation and we use stratification again to ensure each split is well represented as the whole dataset. We use 10 fold cross validation and again compare the accuracies of the various classifiers. Both KNN and DT have decreased to 71.4% and 69.6% respectively while Bernoulli stayed relatively unaffected with 65.6% however, Gaussian Naïve Bayes accuracy increased to 75.4%. This tells us that for this dataset, after cross validation Gaussian Naïve Bayes might be the better model to go with.



Analysis of Iris Dataset:

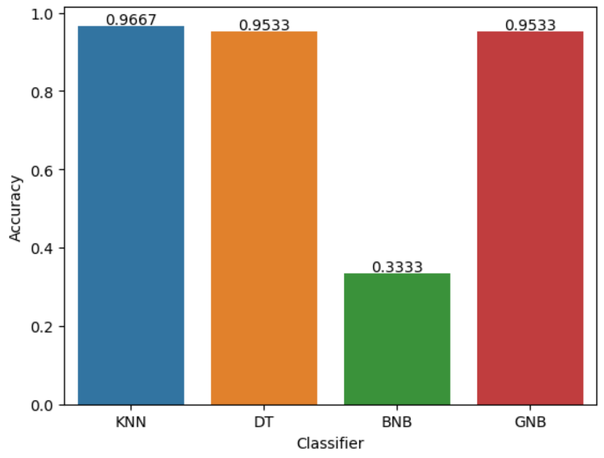
The dataset has 5 columns, sepal-length sepal-width petal-length petal-width and outcome. The dataset has 150 records. We calculate the numerical measures by using describe() for the columns other than outcome. Also all the values are non-null as observed from the .info() as all features have 150 non null values. Also the dataset seems to be balanced as the 3 outcome values iris-setosa, iris-versicolor, iris-virginica each have 50 records meaning they are equally represented.

We plot the histogram for all the 4 features other than outcome to get an overview of how the records values are. Then using pairplot we plot the 4 features across each other and observe the distribution based on the outcome values. We can observe the correlation here. We also plot the correlation heat map.



Next we clean the data and assign numerical values to the categorical values of the outcome attribute and then we select the other 4 columns as features and outcome as response in the dataset. We use the 4 classifiers, KNN(k nearest neighbors) D.T(Decision tree) BNB(Bernoulli naïve bayes) GNB(gaussian naïve bayes).

We get the following accuracies after doing 10 fold stratified cross validation, knn-96% D.T-95.3% Bernoulli-33.3% Gaussian-95.3%. Bernoulli naïve bayes accuracy is very low because it is not suitable for the type of data that is present in the dataset. It works better with binary as opposed to gaussian naïve bayes.

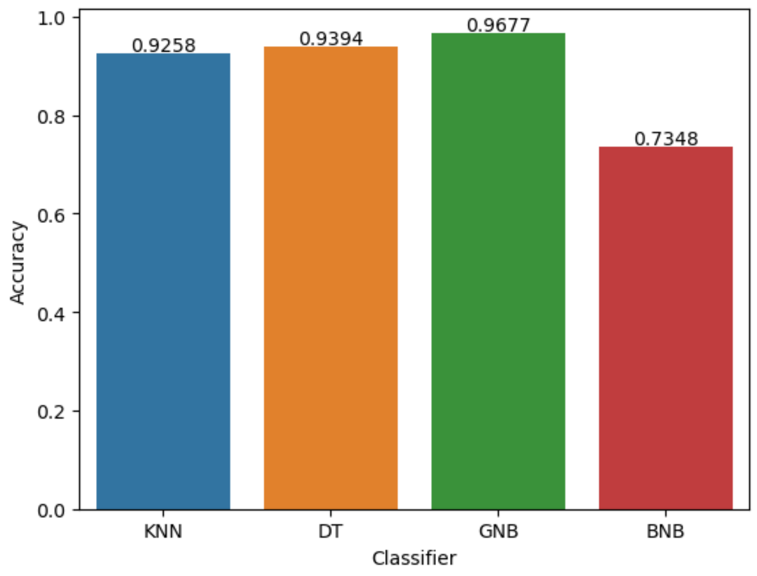


Analysis for Thyroid dataset:

The dataset contains 215 records and 6 columns that are T3_resin, Serum_thyroxin, Serum_triiodothyronine, Basal_TSH, Abs_diff_TSH, Outcome. The outcome attribute has 3 possible values 1,2,3. In the dataset 150 records have outcome 1 while 35 have outcome 2 and remaining 30 have outcome 30. Histograms are plotted to observe the number of records having particular values for all the attributes. Also, as clear by the isnull().sum() and isna().sum() there are no missing values in this dataset as well. Again we split the attributes into the 5 features and finally the outcome which is the response as this is what the model needs to classify a record as.

We again import the same 4 classifiers, K Nearest Neighbors, Decision Tree, Gaussian Naïve Bayes, Bernoulli Naïve Bayes. We do stratified cross validation fold split on the dataset with 10 splits. How do we get the final accuracy?, we get a mean of all the accuracies of the various models{obviously we consider models of only one type of classifier at a time} over the cross validated folds. We get the following accuracies, KNN-92.5%, DT-93.9%,GNB-96.7%,BNB-73.4%.We plot the final bargraph depicting the accuracies of the 4 classifier models.

Since we have used stratified cross fold splits, to explain more about it, lets assume we take k=10, then how the model works is, it is trained on one of those 10 and then tested on the other 9. This process is repeated where the model is trained on all of the other 9a nd tested on the remaining at each time and then finally to get the accuracy the mean of all of the accuracies of such models are considered.


Accuracy
KNN
DT
GNB
BNB
Classifier