

## Problem 1 [20 points]

The files for this problem is under Experiment 1 folder. Datasets to be used for experimentation: telecom churn.csv. Jupyter notebook: Exploratory data analysis.ipynb. In this experiment, we will do exploratory data analysis to get a better sense of data. The dataset contains record of telecom customer along with the label "churn". Churn = "true" signifies that the customer has left the company and churn = "false" signifies that the customer is still loyal to the company.

### 1. How many records are there in the dataset?

→ 3333 datasets.

### 2. How many features are there? Name each feature and assign it as binary, discrete, or continuous.

→ 20 features and 1 outcome(class)

'state' – discrete, 'account length' - discrete, 'area code' - discrete, 'phone number' - discrete, 'international plan' - binary, 'voice mail plan' - binary, 'number vmail messages' - discrete, 'total day minutes' - continuous, 'total day calls' - discrete, 'total day charge' - continuous, 'total eve minutes' - continuous, 'total eve calls' - discrete, 'total eve charge' - continuous, 'total night minutes' - continuous, 'total night calls' - discrete, 'total night charge' - continuous, 'total intl minutes' - continuous, 'total intl calls' - discrete, 'total intl charge' - continuous, 'customer service calls' - discrete.

### 3. As a data scientist, your job is to build a model that identifies customers intending to leave your company. To do that, we prepare our data for the machine learning model. We can have the most advanced algorithm, but if our training data is terrible, our result will be poor. According to your intuition, which features are irrelevant. Briefly explain your reasoning.

→ state, account length, area code, phone number.

I guess that those above features are irrelevant to build a model because those could be classified into categorical features that do not have a logical order or meaning just like a unique ID for each customer has. If there is a specific state where most people use this company plans, then 'state' feature could be a important feature when we build a model, however, it might not be happened.

### 4. Are there any missing values in the data?

→ There are none missing values in the data because 3333 non-null means each column contains 3333 observations.

5. For the continuous features, what is the average, median, maximum, minimum, and standard deviation values?

```
df.describe(include=['float64'])
```

	total day minutes	total day charge	total eve minutes	total eve charge	total night minutes	total night charge	total intl minutes	total intl charge
count	3333.00	3333.00	3333.00	3333.00	3333.00	3333.00	3333.00	3333.00
→ mean	179.78	30.56	200.98	17.08	200.87	9.04	10.24	2.76
→ std	54.47	9.26	50.71	4.31	50.57	2.28	2.79	0.75
→ min	0.00	0.00	0.00	0.00	23.20	1.04	0.00	0.00
25%	143.70	24.43	166.60	14.16	167.00	7.52	8.50	2.30
→ 50%	179.40	30.50	201.40	17.12	201.20	9.05	10.30	2.78
75%	216.40	36.79	235.30	20.00	235.30	10.59	12.10	3.27
→ max	350.80	59.64	363.70	30.91	395.00	17.77	20.00	5.40

→ Each 8 of continues features shows average(mean), median(50%), maximum(max), minimum(min), and standard deviation(std).

6. What is the average number of customer service calls made by a customer to the company?

→ 1.56 calls

7. In our dataset, data comes from how many states?

→ 51 unique states.

8. What's the distribution of the "Churn" feature. Is the feature skewed?

→ Total 483 churned users and 2850 non-churned users.

On normalize, 14% churned and 86% non-churned users. Thus, the feature skewed to non-churned users.

9. What's the highest and lowest "total day charge" encountered by the customer? If we sort the dataset in ascending and descending order by "total day charge", what observation can you make regarding the connection between "total day charge" and "churn" rate?

→ Highest total day charge: 59.64, lowest: 0.0

→ If we sort the dataset in descending by "total day charge", most of them are churned users.

→ If we sort the dataset in ascending by "total day charge", most of them are non-churned users.

**10. What's the average number of customer service calls made by the user who has churned out of the company? Compare and contrast it with the average number of customer service calls made by the user who is still with the company.**

➔ The average of customer service calls made by churned users 2.23 and non-churned users 1.45 calls.

**11. Compare and contrast the average values of numerical features for churned and non-churned users? As a data scientist, what strategy will you recommend to the company to retain more customers?**

➔ Except account categorical features, the average of the number of vmail messages, customer service calls, and total minutes and charges between churned and non-churned users show significant gaps. Usually, it shows that more calls, more charged. I would recommend the company "provide free voice mail message service and be nicer on the phone. Also, the more they call, the cheaper the fare service."

**12. Assume you have devised a model which states that if international plan" = 'no', then the customer will not churn (i.e., "churn" = False). Report accuracy, precision and recall concerning "churned" class.**

➔ Accuracy:  $(2664+137)/3333 = 84\%$ , Precision:  $137/(137+346) = 28.4\%$ , Recall:  $137/(137+186) = 42.4\%$

**13. Calculate  $P(\text{churn} = \text{True} \mid \text{international plan} = \text{'yes'})$ ,  $P(\text{churn} = \text{False} \mid \text{international plan} = \text{'yes'})$ ,  $P(\text{churn} = \text{True} \mid \text{international plan} = \text{'no'})$ ,  $P(\text{churn} = \text{False} \mid \text{international plan} = \text{'no'})$ . Given that the customer has churned, what are the probabilities that the customer has opted/not-opted for the international plan? Similarly, given that the customer has not churned, what are the probabilities that the customer has opted/not-opted for the international plan?**

$P(\text{churn} = \text{True} \mid \text{international plan} = \text{'yes'}) \rightarrow 137/323 = 0.424$

$P(\text{churn} = \text{False} \mid \text{international plan} = \text{'yes'}) \rightarrow 186/323 = 0.576$

$P(\text{churn} = \text{True} \mid \text{international plan} = \text{'no'}) \rightarrow 346/3010 = 0.115$

$P(\text{churn} = \text{False} \mid \text{international plan} = \text{'no'}) \rightarrow 2664/3010 = 0.885$

$P(\text{international plan} = \text{'yes'} \mid \text{churn} = \text{True}) \rightarrow 137/483 = 0.284$

$P(\text{international plan} = \text{'yes'} \mid \text{churn} = \text{False}) \rightarrow 346/483 = 0.716$

$P(\text{international plan} = \text{'no'} \mid \text{churn} = \text{True}) \rightarrow 186/2850 = 0.065$

$P(\text{international plan} = \text{'no'} \mid \text{churn} = \text{False}) \rightarrow 2664/2850 = 0.935$

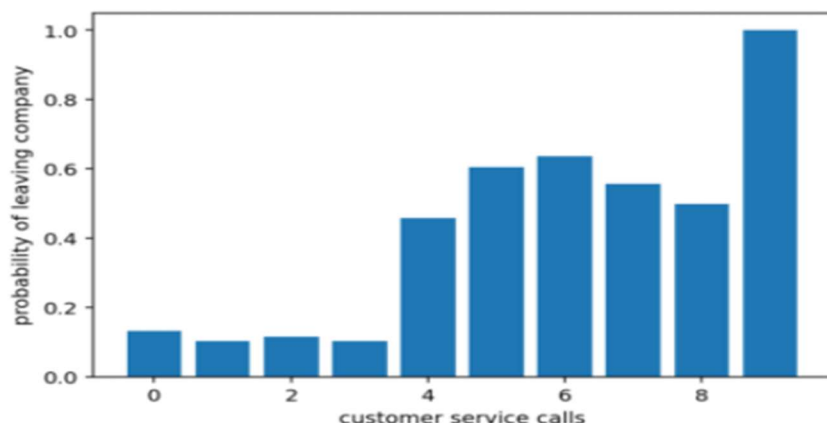
14. Calculate the probability of customers leaving the company, given that he has not made any customer service call. Compare and contrast it with the customer making 1,2,3,4,5,6,7,8,9 customer service calls. Plot the probability of customers leaving the company as customer service calls increase.

```
p = [round(92/697, 4), round(122/1181, 4), round(87/759, 4),
      round(44/429, 4), round(76/166, 4), round(40/66, 4),
      round(14/22, 4), round(5/9, 4), 1/2, 2/2]
y = [0,1,2,3,4,5,6,7,8,9]

plt.xlabel('customer service calls')
plt.ylabel('probability of leaving company')

plt.bar(y, p)
```

<BarContainer object of 10 artists>



The probability of customers leaving the company, given that he has not made any customer service call =  $92 / 697 = 0.132$

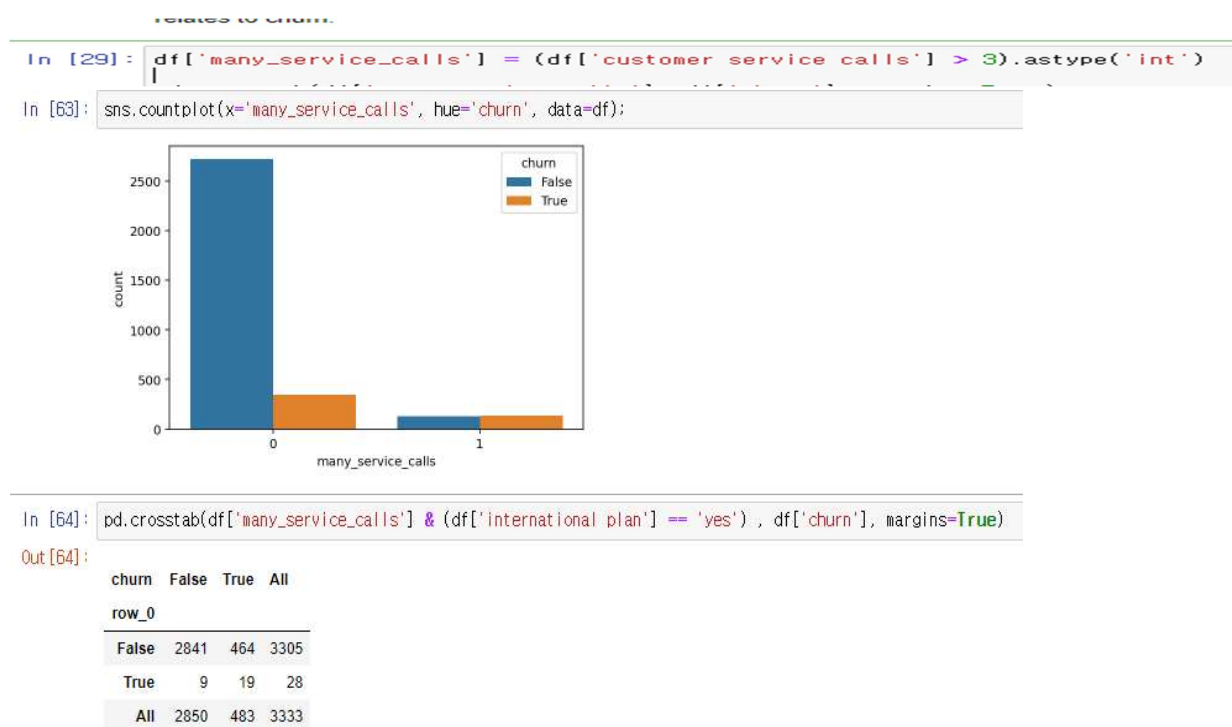
And the probability of customers leaving the company as customer service calls increase

1= 0.1, 2= 0.114, 3= 0.1, 4= 0.458, 5= 0.606, 6= 0.636, 7= 0.555, 8= 0.5, 9= 1

The probability of customers leaving the company increase as customer service calls increase 1-2, 3-6, 8-9 calls.

It can be seen that usually people who makes more customer service calls have a more probability of leaving company.

15. Assume you have devised a model which states that if "international plan" = "yes" and the number of calls to the service center is greater than 3, then the customer will churn (i.e., "churn" = True). Report accuracy, precision and recall concerning "churned" class.



Accuracy:  $(2841+19)/3333 = 0.858$

Precision:  $19/(19+464) = 0.039$

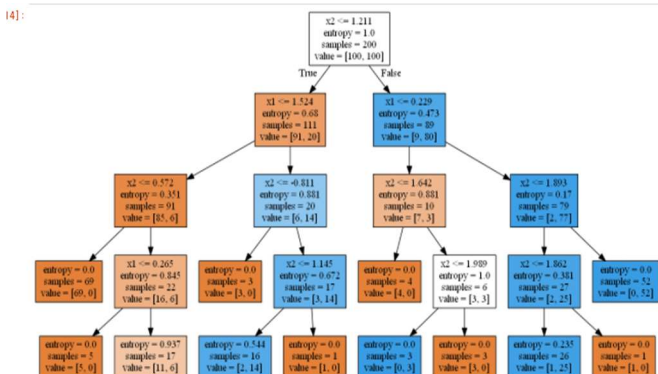
Recall:  $19/(19+9) = 0.679$

It can be seen that usually people who makes 3 or more customer service calls have a more probability of leaving company.

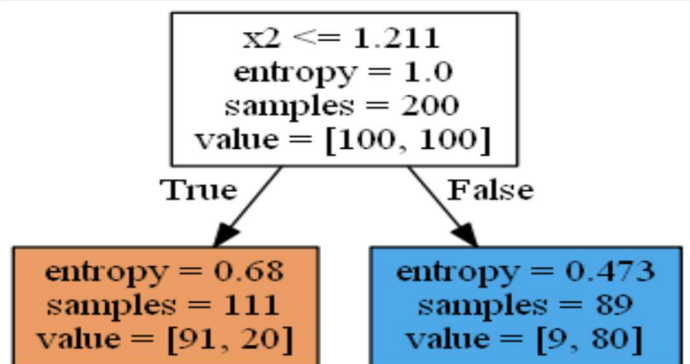
## Problem 2 [20 points]

The files for this problem are under Experiment 2 folder. Datasets to be used for experimentation: telecom churn.csv. Jupyter notebook: Decision Trees and kNN.ipynb. In this experiment we will apply and visualize decision trees, kNN, finetune parameters and learn about k-fold cross validation etc. To visualize decision tree, we need additional packages to be installed i.e. Graphviz and pydotplus. Answer the following questions:

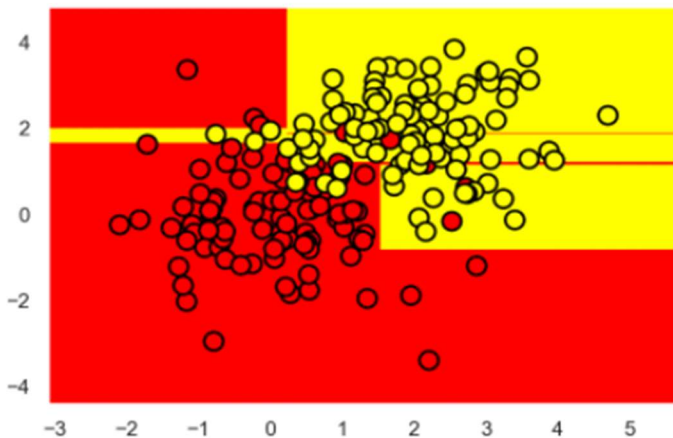
1. For the synthetic dataset, we separate two classes by training a decision tree. What does the boundary look like when we overfit (max depth  $\geq 4$ ) and underfit (max depth = 1) the decision tree on data. For both cases, paste the decision tree and the decision boundary from Jupyter notebook output.



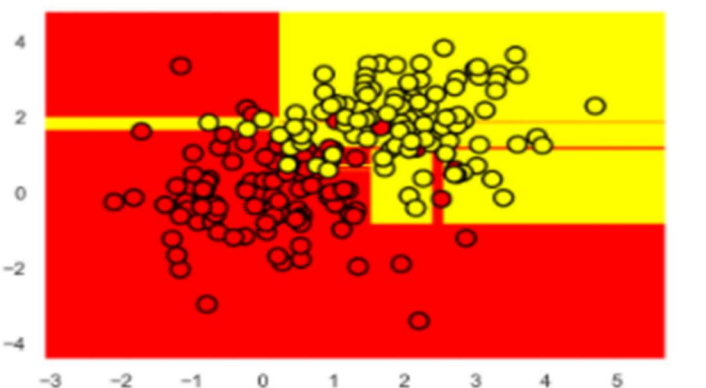
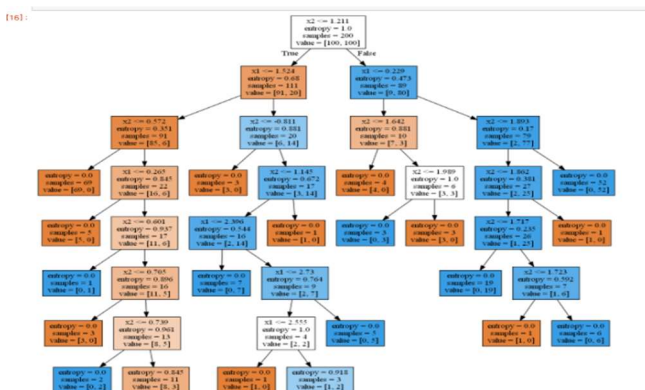
→ Max depth = 4



max depth = 1



→ Max depth = 7



**2. Decision tree classifier `sklearn.tree.DecisionTreeClassifier` has parameter "max depth" which defines the maximum depth of the tree and "criterion" which measure the quality of the split. What happens if we don't specify any value for both parameters?**

- ➔ what does increase the maximum depth mean, what does change the minimum sample leaves nodes in your model. If we don't specify any value for both parameters, don't limits the depth of the tree it will expand all the node till the end, this could be overfitting and criterion will set as "gini" by default.

**3. For Bank Dataset, what are the 5 different age values that the decision tree used to construct the split the tree? What is the significance of these 5 values?**

- ➔ In decision tree sort in ascending order is the simplest heuristics for handling numeric features so the significance of 5 values is that a decision tree looks for the best split(information gain) by checking binary attributes. The best split point is the switching point from 1 to 0 or 0 to 1.
- ➔ 5 different age values are 43.5, 19, 22.5, 30, and 32 years.

**4. Given a dataset `d`, with `n` sample and `m` continuous features, what does Standard Scalers `sklearn.preprocessing.StandardScaler` do? Given dataset `d = [[0, 0], [0, 0], [1, 1], [1, 1]]`, write down its scaler transformation.**

- ➔ It standardizes features by removing the mean and then scaling to unit variance to use in KNN classifier. Shortly, it transforms the data such that its distribution will have a mean value 0 and standard deviation of 1.

`array([[ -1., -1.], [ -1., -1.], [ 1., 1.], [ 1., 1.]])`

**5. In section Underfitting and Overfitting (Jupyter notebook), we have classified two sets of data (smaller dataset and big dataset) using two different decision trees to demonstrate underfitting and overfitting. Briefly describe the experiments done and what do you learn from this experiment?**

→ When we get the train accuracy and test accuracy on small data, small decision tree will be fit because the gap between train accuracy and test accuracy is relatively low.

When we get the train accuracy and test accuracy on big data, big decision tree will be fit because the gap between train accuracy and test accuracy is relatively low and the train and test accuracies high.

**Small decision tree has 2 Max-depth and big decision tree has 8 Max-depth.**

**Train Accuracy for small decision tree trained on small data is 0.901, while test accuracy is 0.884**

$0.901 - 0.884 = 0.017 = 1.7\%$  (underfitting)

**Train Accuracy for Large decision tree trained on small data is 0.974, while test accuracy is 0.862**

$0.974 - 0.862 = 0.112 = 11.2\%$  (overfitting)

**Train Accuracy for small decision tree trained on Big data is 0.888, while test accuracy is 0.881**

$0.888 - 0.881 = 0.007 = 0.7\%$  (underfitting)

**Train Accuracy for Large decision tree trained on Big data is 0.957, while test accuracy is 0.944**

$0.957 - 0.944 = 0.013 = 1.3\%$  (overfitting)

Having a overfitting problem is better than underfitting since the train accuracy is way higher than small decision tree.

If a model has a great accuracy on training data while it doesn't on unseen data, this could cause overfitting. To prevent the overfitting we minimize the size of the data, this could cause underfitting. We need to find the best size of the data set which is not suffer from under or over-fitting.



6. In section Imbalance Class (Jupyter notebook), we have trained a couple of classifiers on the balanced and unbalanced dataset and evaluated its accuracy on balanced and unbalanced dataset. Furthermore, we have printed out the confusion matrix. Briefly describe the experiments done, and what do you learn from these experiments? Also, write down precision, recall, and f1 score for the experiments.

- ➔ Accuracy is a poor indicator of the Imbalanced case. It can be seen that some Imbalanced matrixes get greater precision than balanced.
- ➔ In the case of imbalanced class problem, accuracy does not evaluate how well the performance of the algorithm on the smaller class. Instead, F-measure is a good indicator of the classifier's performance since it takes into account both precision and recall. However, there some Imbalanced accuracy get still high, using TPR/FRP is the best choice here.

[[350 0] **Balanced**

[ 0 350]] Precision = 1, Recall = 1, f1 score = 1

[[688 179] **Imbalanced**

[ 0 5]] Precision = 1, Recall = 0.79, f1 score = 0.88

[[108 25] **Balanced**

[ 30 103]] Precision = 0.78, Recall = 0.81, f1 score = 0.8

[[1983 0] **Imbalanced**

[ 0 10]] Precision = 1, Recall = 1, f1 score = 1

[[133 0] **Balanced**

[112 21]] Precision = 0.54, Recall = 1, f1 score = 0.7

[[864 3] **Imbalanced**

[ 5 0]] Precision = 0.99, Recall = 0.997, f1 score = 0.99

**7. In section Adding irrelevant attributes (Jupyter notebook), we have added an irrelevant attribute to the dataset and have trained a decision tree classifier on it. Based on the test set results, what do you think has happened, and can you collaborate it with the class material? Briefly describe the experiment done, and the intuition developed from these experiments.**

- ➔ According to the accuracy with addition of irrelevant attributes of decision tree, even though the accuracy of decision tree on test get lower, it still performs well as long as there not too many noise attributes. If there are too many noise attributes, they are likely to be selected at many internal nodes and thus make the model too complex tree.

Without addition of irrelevant attributes ->

Accuracy on test dataset 0.92 and Accuracy on training dataset 1.0

With addition of irrelevant attributes ->

Accuracy on test dataset 0.884 and Accuracy on training dataset 1.0

**8. For the customer churn prediction task, we show that the accuracy of the decision tree is 94% when max depth is set to 6. What happens to accuracy when we leave the value of max depth to its default value? Explain the rise/fall of accuracy.**

- ➔ When I leave the value of max depth to its default value, the accuracy falls to 0.91 since it will expand till the end because max\_depth 6 is the best parameter on the decision tree.

**9. How many decision trees do we have to construct if we have to search the two-parameter space, max depth [1-10] and max features [4-18]? If we consider 10-fold cross-validation with the above scenario, how many decision trees do we construct in total?**

- ➔ Depth 10 x features 15 = 150 and fitting 10-fold cross validation 150 candidates so  $10 \times 150 = 1500$  trees.

**10. For the customer churn prediction task, what is the best choice of k [1-10] in the k-nearest neighbor algorithm in the 10-fold cross-validation scenario?**

- ➔ 9 neighbors get the accuracy of the KNN 88.7.% as best.

**11. For MNIST dataset, what was the accuracy of the decision tree [max depth = 5] and K-nearest neighbor [K = 10]? What are the best parameters and accuracy for holdout dataset for decision trees when we used GridSearchCV with 5-fold cross-validation?**

- ➔ The accuracy of the decision tree(max depth=5) = 0.66. and knn(k=10) = 0.976.
- ➔ The best parameters for decision trees when we used GridSearchCV with 5-fold cross-validation are max\_depth = 10, max\_features = 50 and accuracy = 0.843.

## Problem 3 [20 points]

The files for this problem are under Experiment 3 folder. Datasets to be used for experimentation: spam.csv. Jupyter notebook: Naive Bayes Spam.ipynb. The dataset contains 5,574 messages tagged according to ham (legitimate) or spam. In this experiment we will learn about text features, how to convert them in matrix form and Naive Bayes algorithm.

1. How many records are there? What's the distribution of the "label" class? Is it skewed?

→ 5572 records included 4825 ham, 747 spam SMS and the record is skewed to ham.

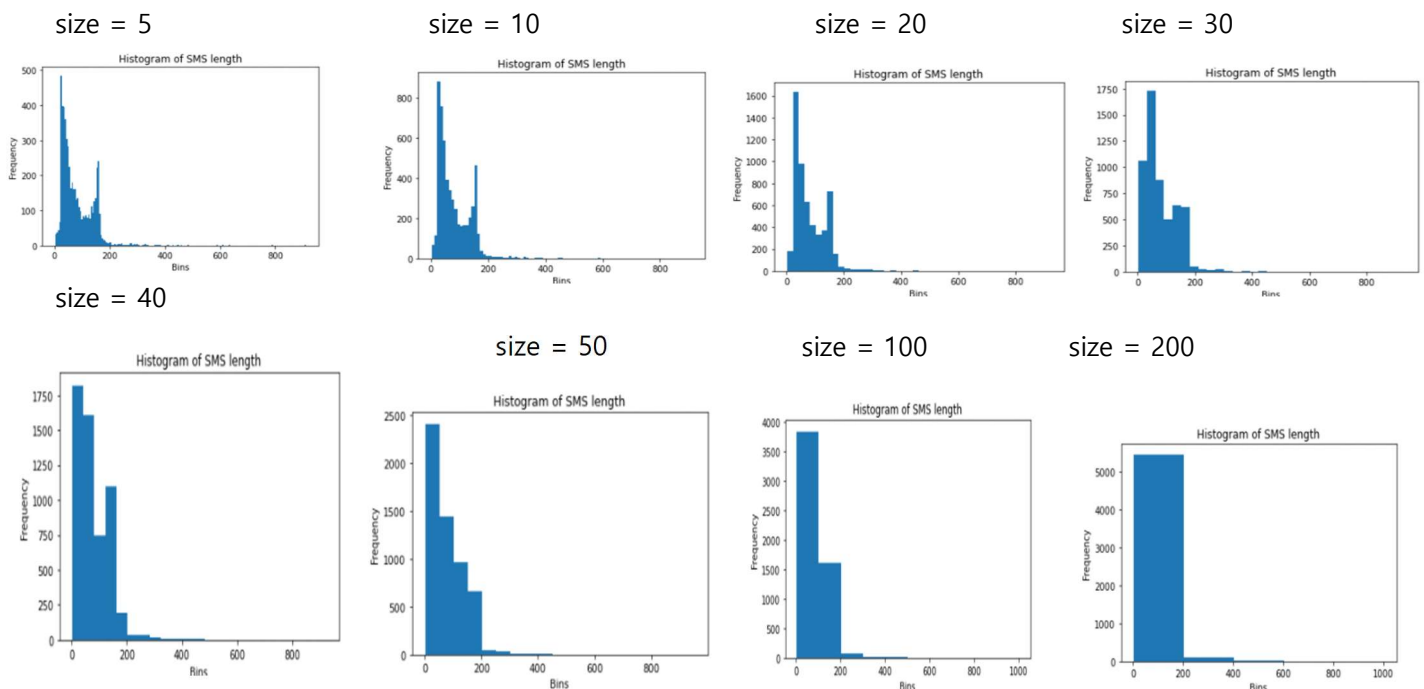
2. How many unique SMS is there in the dataset? What is the SMS that occurred most frequently and what is its frequency?

→ 5169 unique sms, "Sorry, I'll call later", and 30 frequency.

3. What is the maximum and minimum length of SMS present in the dataset? Plot the histogram of the length of SMS with bin size 5,10,20,30,40,50,100,200. What do you hypothesize with the plots?

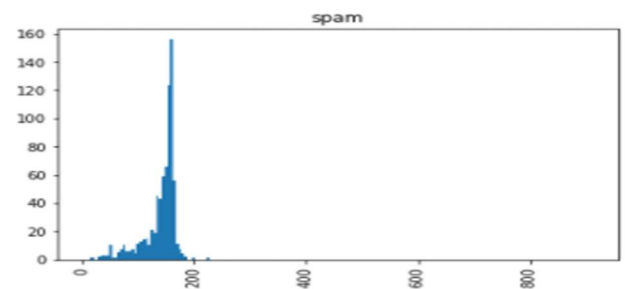
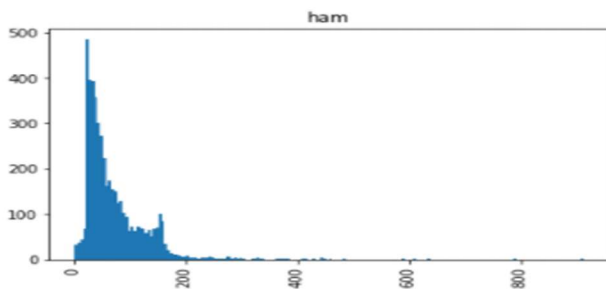
→ The maximum length of SMS is 910 and the minimum is 2.

→ The bin size gets bigger the size, the more monotonous its histogram becomes. If the bin size is small or big, it will not cover all the information they need to show. Most of the length of SMS is short only few of them shows long when the length is over 200.

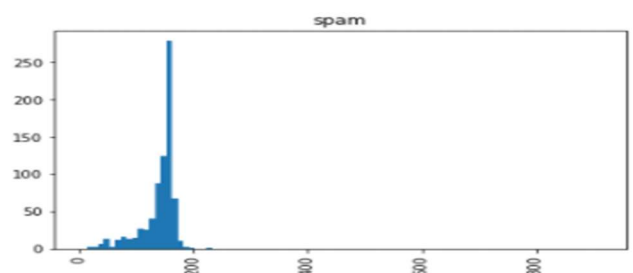
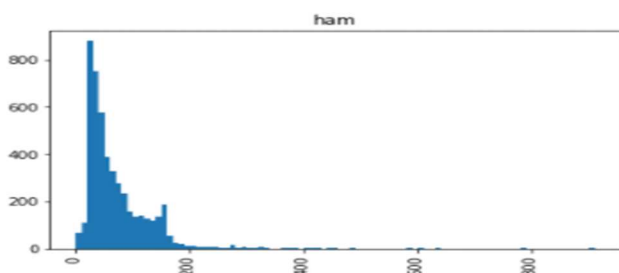


4. Plot the histogram of the length of SMS for each label separately with bin size 5,10,20,50 i.e. histogram of the length of all ham SMS and histogram of the length of all spam SMS. What can you perceive after examining the plots?

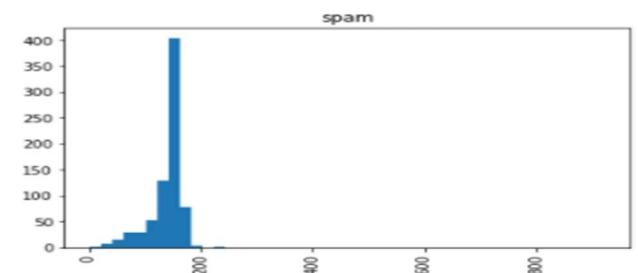
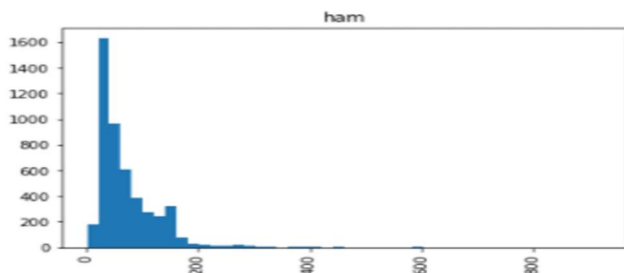
Size = 5



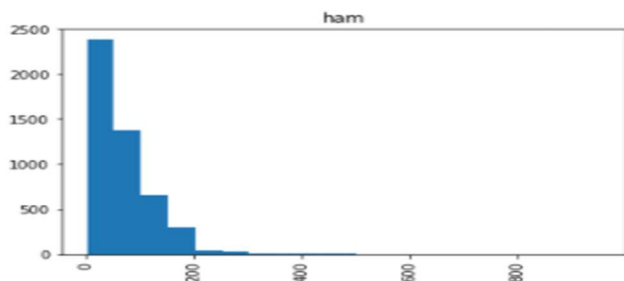
Size = 10



Size = 20



Size = 50



➔ The bin size gets bigger the size, the more monotonous its histogram becomes. It seems like enlarge the histogram and when it gets bigger, it is hard to expect the small values there. As the histograms increase the width of bin size, the frequency becomes clearer. Ham SMS are usually shorter than Spam SMS. Using a proper which is not small or big is the best choice to deliver the all information that we need to show.

**5. In the Bag of words approach, we convert all strings into lower cases. Why did we do that, and why is it important? Can we convert all strings into the upper case and still fulfill our original goal?**

- Since upper and lower cases are same in the count function, we do not need to produce for a new row for upper and lower cases. Using sklearn's count vectorizer function requires tokenize with lowercase. Since those upper and lower cases shows a same frequency because those have same meaning in a dataset, we need to convert all strings into lower cases or upper cases. If the countvectorizer function counts upper cases, then converting into the upper case will be fine too. Just we need to set up the either lower case count or upper case count.

**6. What does CountVectorizer achieve? What will happen if we set stop words = "english". Give five examples of stop-words in English.**

- CountVectorizer() method from scratch that entailed cleaning our data first. This cleaning involved converting all of the data to lower case and removing all punctuation marks.
- If we set stop words = "english", this removes all words from the document set that match a list of english stop words which is defined in scikit-learn.

Example: of, co, other etc

**7. Given a dataset, how do we generate a document-term matrix? Do we first generate document-term matrix and then separate matrix into train/test or first separate the data into train/test and then generate document-term matrix based on train dataset and afterwards generate matrix for test set? Explain your reasoning.**

- We first separate matrix into train/test and generate document-term matrix and then generate document-term matrix. Otherwise, test set could have seen data which does not give a great prediction. The purpose of splitting train and test set is that test set is not used to train the model. Instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values.

**8. Using bag of words approach, convert documents = ['Hi, how are you?', 'Win money, win from home. Call now.', 'Hi., Call you now or tomorrow?'] to its document-term matrix.**

	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. How many features are created while making document-term matrix for SMS dataset? Can you think of a method to reduce the number of features? List the pros and cons of the method.**

→ 7777 features.

→ A method to reduce the number of features is do not count the not-important words like "hi", "bye" the greeting words and repeated words such as "Good morning", "call you later", etc. There are lots of way to reduce the features, however, if we reduce those greeting and repeated words, this could loss the important information if the repeated word is not spam and while tokening the words and it still could classify to Spam SMS.

**10. For our input dataset, which Naive Bayes model should we use, Gaussian Naive Bayes or Multinomial Naive Bayes? Explain your reasoning? Report accuracy, precision, recall and F1 score for the spam class after applying Naive Bayes algorithm.**

→ Multinomial Naive Bayes works better performance for classification with discrete features like our case: word counts for text classification because It takes in integer word counts as its input.

On the other hand, Gaussian Naive Bayes works better for classification with continuous data as it assumes that the input data has a Gaussian(normal) distribution.

Accuracy score: 0.985

Precision score: 0.94

Recall score: 0.935

F1 score: 0.94

## 4 Problem [20 points]

The files for this problem are under Experiment 4 folder. In this assignment, we provide three real-world datasets for classification, i.e., Iris dataset (<https://archive.ics.uci.edu/ml/datasets/Iris>), Thyroid dataset (<https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease>), and Diabetes dataset (<https://www.kaggle.com/uciml/pima-indians-diabetes-database>). Also, we give three jupyter notebooks, one for each dataset, in which we have applied k-nearest neighbor, decision-tree, and Naive Bayes Algorithm without any parameter tuning.

Write a short report for each dataset; you can be as innovative as you want, giving your analysis about the dataset, your observations, and comments. It should be no more than half a page to a page in length. It can include a description of the dataset, the number of observations, missing value or not, testing strategy you deployed, classification accuracy of algorithms, etc.

### Iris dataset-

There are 150 records(datasets=row) which have 4 features(columns): sepal-length, sepal-width, petal-length, petal-width and 1 outcome(class) in the dataset.

The 3 outcomes(classes) are 50 of Iris-setosa, 50 of Iris-versicolor, and 50 of Iris-virginica, total 150 non-skewed records show.

There are none missing values in the data because 150 non-null records contain 150 observations.

The 'sepal-length' has the maximum value of 7.9, the highest mean 5.84, and highest median 5.8, while 'sepal-width' has the lowest std 0.43.

The 'petal-length' has the highest std 1.76 while 'petal-width' has the lowest min 0.1 and max 2.5 and have the lowest mean 1.2.

I can expect that if one has smaller value of sepal-length, petal-width, petal-length, and sepal-length, then it would be Iris-setosa. If one has larger petal-width, sepal-length, petal-length, then it would be Iris-Virginica. If one has medium length of petal and sepal, then it would be Iris-versicolor.

When we use train and test split, classification accuracy could be gathered by K-Near-Neighbor, Decision-Tree, Bernoulli-Naive-Bayes, Gaussian-Naive-Bays.

When we use k-fold cross validation, KNN and GNB shows the highest accuracy of 96%, DT has 95%, and BNB has the lowest 33.3%.

## Thyroid dataset

There are 215 records(datasets=row) which have 5 features(columns): 'T3\_resin', 'Serum\_thyroxin', 'Serum\_triiodothyronine', 'Basal\_TSH', 'Abs\_diff\_TSH' and 1 outcome(class) in the dataset.

The 3 outcomes(classes) are 150 of 1, 35 of 2, and 30 of 3, total 215 records is skewed to class of 1.

There are none missing values in the data because 215 non-null records contain 215 observations.

The 'T3\_resin' has the maximum value of 144, the highest mean 109.6, and highest median 110 while 'Serum\_triiodothyronine' has the lowest mean 2.05, std 1.42, and median 1.7 and lowest value of 10.

The 'Abs\_diff\_TSH' has the highest std 8.07, but has the lowest value of -0.7.

I can expect that if one has lower value of Basal\_TSH, Abs\_diff\_TSH, Serum\_triiodothyronine, then the one would be in class 1. If one has larger T3\_resin and Serum-thyroxin, then the one would be in class 3.

When we use train and test split, classification accuracy could be gathered by K-Near-Neighbor, Decision-Tree, Bernoulli-Naive-Bayes, Gaussian- Naive-Bays.

When we use k-fold cross validation, GNB shows the highest accuracy 96.77%, DT has 94.4%, KNN has 92.9% and BNB has the lowest accuracy which is 73.5%.

## Diabetes dataset

There are 768 records(datasets=row) which have 8 features(columns): 'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', and 1 outcome(class) in the dataset.

The 2 outcomes(classes) are 500 of 0 and 268 of 1, total 768 non-skewed records.

There are none missing values in the data because 768 non-null records contain 768 observations.

The 'Glucose' has the highest mean 120.9, and highest median 117 while 'DiabetesPedigreeFunction' has the lowest value of 2.42, std 0.33, and median 0.372 and lowest value of 0.078.

The 'Insulin' has the highest value of 846 and highest std of 115.2.

I can expect that if one has higher value of Glucose, BloodPressure, BMI, then the one would be class of 1. If the one has lower value of SkinThickness, Insulin, DiabetesPedigreeFunction, and age, then the one would be class of 0.

When we use train and test split, classification accuracy could be gathered by K-Near-Neighbor, Decision-Tree, Bernoulli-Naive-Bayes, Gaussian-Naive-Bays.

When we use k-fold cross validation, GNB shows the highest accuracy of 75.6%, KNN has 70.2%, DT has 70.2%, and BNB has the lowest accuracy which is 65.6%.






## Problem 5 [20 points]

Proceed to the following link <https://www.kaggle.com/c/titanic/overview> and follow the description meticulously. Then check out Alexis Cook's Titanic tutorial

<https://www.kaggle.com/alexisbcook/titanic-tutorial> and get ready to make your submission.




1. Submit example gender submission.csv file that predicts that all female passengers survived, and all-male passengers died.

Your most recent submission				
Name	Submitted	Wait time	Execution time	Score
submission1.csv	just now	1 seconds	1 seconds	0.76555
Complete				
<a href="#">Jump to your position on the leaderboard</a>				

7022	Kim Hyeon Soo		0.77751	50	2h
7023	Hyunwoo Teddy Kim		0.77751	10	now
<b>Your Best Entry</b> ↑					
Your submission scored 0.76555, which is not an improvement of your best score. Keep trying!					
7024	ttz kky		0.77751	2	36m

2. Submit the gender submission.csv file that predicts that all passengers survived.

Your most recent submission				
Name	Submitted	Wait time	Execution time	Score
submission2.csv	a minute ago	1 seconds	0 seconds	0.37799
Complete				
<a href="#">Jump to your position on the leaderboard</a>				

7022	Kim Hyeon Soo		0.77751	50	2h
7023	Hyunwoo Teddy Kim		0.77751	11	~10s
<b>Your Best Entry</b> ↑					
Your submission scored 0.37799, which is not an improvement of your best score. Keep trying!					
7024	ttz kky		0.77751	2	37m

### 3. Submit the gender submission.csv file that predicts that all passengers died.

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submission3.csv	just now	4 seconds	1 seconds	0.62200

Complete

[Jump to your position on the leaderboard](#)

7022	Kim Hyeon Soo		0.77751	50	2h
7023	Hyunwoo Teddy Kim		0.77751	12	now
<b>Your Best Entry</b>					
Your submission scored 0.76555, which is not an improvement of your best score. Keep trying!					
7024	ttz kky		0.77751	2	39m

### 4. Submit the model output of the random forest model as detailed in the tutorial. Try to play with the number of decision trees (we constructed 100) and see if accuracy improves.

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
summission4.csv	just now	1 seconds	0 seconds	0.76555

Complete

[Jump to your position on the leaderboard](#)

7022	Kim Hyeon Soo		0.77751	50	2h
7023	Hyunwoo Teddy Kim		0.77751	13	~10s
<b>Your Best Entry</b>					
Your submission scored 0.76555, which is not an improvement of your best score. Keep trying!					
7024	ttz kky		0.77751	2	39m

5. Copy and Edit the kernel in <https://www.kaggle.com/ldfreeman3/a-data-science-framework-to-achieve-99-accuracy>. Submit the model output from the kernel, write a short (half page to a page) report on what the kernel does and include the position on the leader-board, screenshot as detailed above.

Based on the 'Titanic' project, we can learn both the basics and the deep version of data mining at the same time, and it will be a really good reference, especially for those who are new to data mining. Starting with a framework, there are each step and you can learn these things from that step: how to define the problem, gather the data, prepare data for consumption, perform exploratory analysis, model data, tune model with hyper-parameters, validate and implement, and how to optimize and strategize.

This kernel uses all different implementation of a classification method such as decision tree, random forest, K-nearest neighbors, XGB, etc. It also shows how to implement and plot with all different classification.

There is a step that how to create my own model according to the question like Were you on the Titanic? Are you male or female, Are you in class 1,2, or3? This allows us to define and classify the problem and model step by step. Before proceeds to the cross validation, it tests first through simple examples such as coin flip and start the actual game. Everything was explained procedurally and easily, so I could follow it and couldn't believe the results which is the 0.77751 Score. Everything from Experiment1 to 4 in our project is aggregated, in problem5, everything from submission1 to 4 is included, and this shows the basics of data mining.

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submission5.csv	a few seconds ago	1 seconds	0 seconds	0.77751

Complete

[Jump to your position on the leaderboard](#)

7018	zhaoludesu		0.77751	5	1h
7019	Kim Hyeon Soo		0.77751	50	1h
7020	Hyunwoo Teddy Kim		0.77751	7	1m

Your Best Entry ↑

Your submission scored 0.77751, which is an improvement of your previous score of 0.76555. Great job!

Tweet this!

7021	Çağlar Sevinç		0.77511	3	3mo
------	---------------	--	---------	---	-----