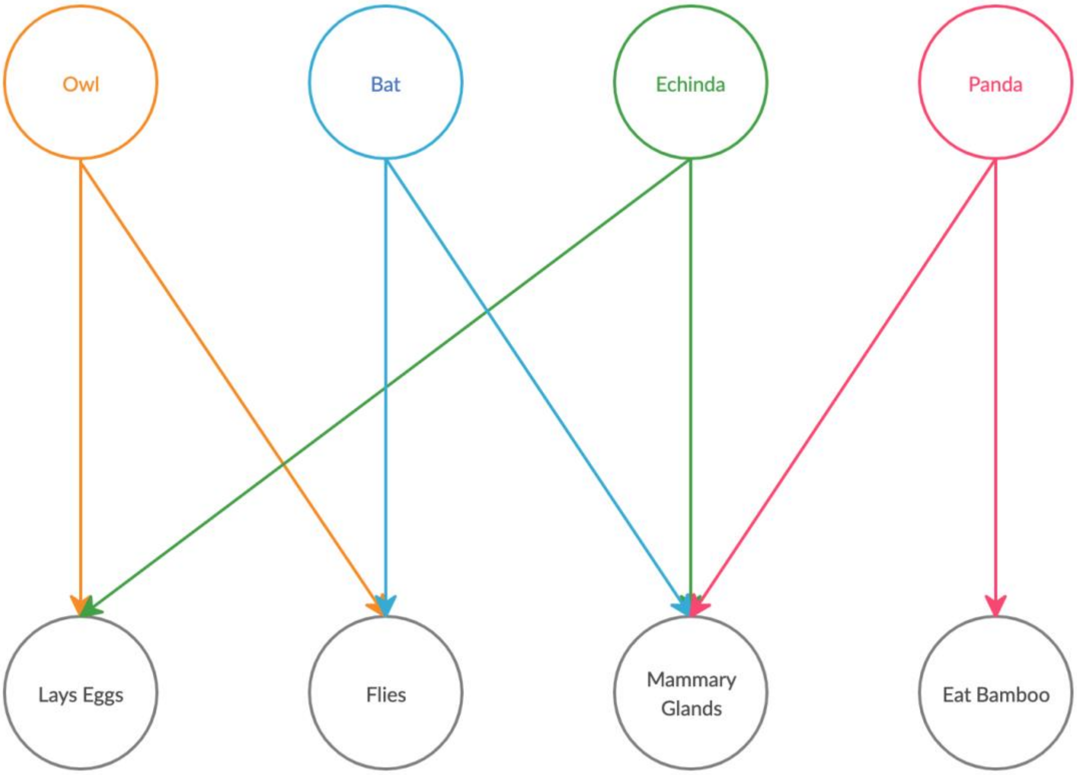
CSC520 – Artificial Intelligence

Assignment 5

Question 1- Using the knowledge base, show the corresponding Bayes Net. (Caution: the complexity of the rest of the assignment depends on the directionality of the edges in the net, so think carefully: does the animal classification cause the physical attributes or do the physical attributes cause the classification?)

Answer –



Question 1.2 - Using the ideas of conditional independence discussed in class and in Ch. 14 of R&N, identify which variables in this network are conditionally independent of which others given what conditions. For example, is being an echidna conditionally independent of being a lizard given that it lays eggs? Is flying conditionally independent of having mammary glands given it being a bat (etc. etc.)? Where appropriate, it's sufficient to make shorthand statements such as "X is conditionally independent of the rest of the network given Y", to save having to enumerate all the nodes in the rest of the network.

P(Owl = T)	P(Bat = T)	P(Echidna = T)	P(Panda = T)
0.05	0.03	0.01	0.02

Panda	Eats_bamboo	P(Eats_bamboo | Panda)
T	T	0.7
F	T	0.03

Owl	Echidna	Lays_eggs	P(Lays_eggs | Owl, Echidna)
T	T	T	0.7
T	F	T	0.57
F	T	T	0.03
F	F	T	0.04

Owl	Bat	Flies	P(Flies | Owl, Bat)
T	T	T	0.9
T	F	T	0.95
F	T	T	0.86
F	F	T	0.03

Echidna	Bat	Panda	Mammary_glands	P(Mammary_glands | Echidna, Bat, Panda)
T	T	T	T	0.9
T	T	F	T	0.75
T	F	T	T	0.4
T	F	F	T	0.02
F	T	T	T	0.5
F	T	F	T	0.08
F	F	T	T	0.5
F	F	F	T	0.01

Answer – For variables to be conditionally independent in Bayesian network they have to be in following conditions:

1. X is conditionally independent of it nondescendents given its parents.
2. Markov Blanket : X is conditionally independent of all network variables given its parents, children and children's parents.
3. Children are independent of each other given their parents.

Conditional Independent variables:

1. Owl is conditional independent of all nodes given Lays Eggs, Flies, Bat
2. Bat is conditional independent of all nodes given Flies, Mammary Glands, Echinda, Owl
3. Echinda is conditional independent of Flies and Eat Bamboo given Mammary Glands, Lays Eggs, Owl, Bat, Panda
4. Panda is conditional independent of Owl, Lays Eggs and Flies given Eat Bamboo, Mammary Glands, Echinda, Bat
5. Flies are conditional independent of Mammary Glands given Bat
6. Flies are conditional independent of Lays Eggs given Owl
7. Mammary Glands are conditional independent of Lays Eggs given Echinda
8. Mammary Glands are conditional independent of Eat Bamboo given Panda
9. Lays Eggs is conditional independent of Eat Bamboo, given Owl and Echinda
10. Flies is conditional independent of Eat Bamboo given Bat

Question 1.3 - Use the prior and conditional probabilities provided to construct probability tables for each node of this network. Then use exact inference to answer the following questions. What is the probability that

1. The animal can fly.
2. The animal is an echidna.
3. The patient is a bat, given that it eats bamboo.

Answer – Below are my full probability tables:

	P(Owl)	P(Bat)	P(Echidna)	P(Panda)
T	0.05	0.03	0.01	0.02
F	0.95	0.97	0.99	0.98

Panda	Eats_bamboo	P(Eats_bamboo | Panda)
T	T	0.7
F	T	0.03
F	F	0.97
T	F	0.3

Owl	Echidna	Lays_eggs	P(Lays_eggs | Owl, Echidna)
T	T	T	0.7
T	F	T	0.57
F	T	T	0.03
F	F	T	0.04
T	T	F	0.3
T	F	F	0.43
F	T	F	0.97
F	F	F	0.96

Owl	Bat	Flies	P(Flies | Owl, Bat)
T	T	T	0.9
T	F	T	0.95
F	T	T	0.86
F	F	T	0.03
T	T	F	0.1
T	F	F	0.05
F	T	F	0.14
F	F	F	0.95

Echidna	Bat	Panda	Mammary_glands	P(Mammary_glands | Echidna, Bat, Panda)
T	T	T	T	0.9
T	T	F	T	0.75
T	F	T	T	0.4
T	F	F	T	0.02
F	T	T	T	0.5
F	T	F	T	0.08
F	F	T	T	0.5
F	F	F	T	0.01
T	T	T	F	0.1
T	T	F	F	0.25
T	F	T	F	0.6
T	F	F	F	0.98
F	T	T	F	0.5
F	T	F	F	0.92
F	F	T	F	0.5
F	F	F	F	0.99

Abbreviations used:

O – Owl, B - Bat, E - Echinda, P – Panda, LE – Lays Eggs, F – Fly, MG – Mammary Glands, EB – Eat Bamboo

1. **The Animal can fly:**

$P(F) = \Sigma_{O,B,E,P,LE,MG,EB} P(O, B, E, P, LE, F, MG, EB)$

From Bayesian Network in 1st part we can write above equation as:

$= \Sigma_{O,B,E,P,LE,MG} P(O) P(B) P(E) P(P) P(LE|O,E) P(F|O,B) P(MG|E,B,P) \Sigma_{EB} P(EB|P)$

From Probability table we can see that $\Sigma_{EB} P(EB|P) = 1$

$= \Sigma_{O,B,E,P,LE} P(O) P(B) P(E) P(P) P(LE|O,E) P(F|O,B) \Sigma_{MG} P(MG|E,B,P)$

From Probability table we can see that $\Sigma_{MG} P(MG|E,B,P) = 1$

$= \Sigma_{O,B,E,P} P(O) P(B) P(E) P(P) P(F|O,B) \Sigma_{LE} P(LE|O,E)$

From Probability table we can see that $\Sigma_{LE} P(LE|O,E) = 1$

$= \Sigma_{O,B} P(O) P(B) P(F|O,B) \Sigma_{E} P(E) \Sigma_{P} P(P)$

From Probability table we can see that $\Sigma_{E} P(E) = 1$ $\Sigma_{P} P(P) = 1$

$= \Sigma_{O,B} P(O) P(B) P(F|O,B)$

Now we have two nodes O, B and we have to try combinations of O and B to find value of Fly. We will take O False one, then B False then Both False and then none False

$= P(O) P(B) P(F|O,B) + P(\neg O) P(B) P(F|\neg O,B) +$

$P(O)\ P(\neg B)\ P(F|O, \neg B) + P(\neg O)\ P(\neg B)\ P(F|\neg O, \neg B)$

$= 0.05 * 0.03 * 0.9 + 0.95 * 0.03 * 0.86 + 0.05 * 0.97 * 0.95 + 0.95 * 0.97 * 0.03$

$= 0.00135 + 0.02451 + 0.046075 + 0.027645$

P(F) = 0.09958

Probability that Animal can Fly is 0.09958

2. **The Animal is an echidna:**

$P(Echidna) = \Sigma_{O,B,P,LE,F,MG,EB}\ P(O, B, E, P, LE, F, MG, EB)$

From Bayesian Network in 1st part we can write above equation as:

$= \Sigma_{O,B,P,LE,F,MG}\ P(O)\ P(B)\ P(E)\ P(P)\ P(LE|O,E)\ P(F|O,B)\ P(MG|E,B,P)\ \Sigma_{EB}\ P(EB|P)$

From Probability table, $\Sigma_{EB}\ P(EB|P) = 1$

$= \Sigma_{O,B,P,LE,F}\ P(O)\ P(B)\ P(E)\ P(P)\ P(LE|O,E)\ P(F|O,B)\ \Sigma_{MG}\ P(MG|E,B,P)$

From Probability table we can see that $\Sigma_{MG}\ P(MG|E,B,P) = 1$

$= \Sigma_{O,B,P,LE}\ P(O)\ P(B)\ P(E)\ P(P)\ P(LE|O,E)\ \Sigma_{F}\ P(F|O,B)$

From Probability table, $\Sigma_{F}\ P(F|O,B) = 1$

$= \Sigma_{O,B,P}\ P(O)\ P(B)\ P(E)\ P(P)\ \Sigma_{LE}\ P(LE|O,E)$

From Probability table we can see that $\Sigma_{LE}\ P(LE|O,E) = 1$

$= \Sigma_{O}\ P(O)\ \Sigma_{B}\ P(B)\ P(E)\ \Sigma_{P}\ P(P)$

From Probability table we can see that $\Sigma_{O}\ P(O) = 1\ \Sigma_{B}\ P(B) = 1\ \Sigma_{P}\ P(P) = 1$

$= P(E)$

From Probability table we can see P(E) = 0.01

From Bayesian Network also we can see that, we do not have any parent of Echinda. So, Its probability will be as it is given in the table. **P(Echinda) = 0.01**

3. **The patient is a bat, given that it eats bamboo:**

Using conditional Probability formula P(X|Y) = P(X,Y) | P(Y), we can write as:

P(Bat | Eat Bamboo) $= \frac{P(Bat,Eat\ Bamboo)}{P(Eat\ Bamboo)}$

$$\frac{P(B,EB)}{P(EB)} = \frac{\Sigma_{O,E,P,LE,F,MG}\, P(O,B,E,P,LE,MG,F,EB)}{P(EB)}$$

$$= \frac{\Sigma_{O,E,P,LE,F,MG}\, P(O)\ P(B)\ P(E)\ P(P)\ P(LE|O,E)\ P(F|O,B)\ P(EB|P)\ P(MG|E,B,P)}{P(EB)}$$

$$= \frac{\Sigma_{O,E,P,LE,F}\, P(O)\ P(B)\ P(E)\ P(P)\ P(LE|O,E)\ P(F|O,B) P(EB|P)\ \Sigma_{MG} P(MG|E,B,P)}{P(EB)}$$

From Probability table we can see that Σ_{MG} P(MG|E,B,P) = 1

$$= \frac{\Sigma_{O,E,P,LE,F}\, P(O)\ P(B)\ P(E)\ P(P)\ P(LE|O,E)\ P(F|O,B) P(EB|P)}{P(EB)}$$

$$= \frac{\Sigma_{O,E,P,LE}\, P(O)\ P(B)\ P(E)\ P(EB|P)\ P(P)\ P(LE|O,E)\ \Sigma_F P(F|O,B)}{P(EB)}$$

From Probability table, Σ_F P(F|O,B) = 1

$$= \frac{\Sigma_{O,E,P,LE}\, P(O)\ P(B)\ P(E)\ P(EB|P)\ P(P)\ P(LE|O,E)}{P(EB)}$$

$$= \frac{\Sigma_{O,E,P}\, P(O)\ P(B)\ P(E)\ P(EB|P)\ P(P)\ \Sigma_{LE} P(LE|O,E)}{P(EB)}$$

From Probability table we can see that Σ_{LE} P(LE|O,E) = 1

$$= \frac{\Sigma_{O,E,P}\, P(O)\ P(B)\ P(P)\ P(EB|P)\ P(E)}{P(EB)}$$

$$= \frac{\Sigma_{O,P}\, P(O)\ P(B)\ P(EB|P)\ P(P)\ \Sigma_E P(E)}{P(EB)}$$

From Probability table we can see that Σ_E P(E) = 1

$$= \frac{\Sigma_{O,P}\, P(O)\ P(B)\ P(EB|P)\ P(P)}{P(EB)}$$

$$= \frac{\Sigma_P\, P(B)\ P(EB|P)\ P(P)\ \Sigma_O P(O)}{P(EB)}$$

From Probability table we can see that Σ_o P(O) = 1

$$= \frac{\Sigma_P\, P(EB|P)\ P(P)\ P(B)}{P(EB)}$$

We know that Σ_P P(EB|P) P(P) = P(EB)

$$= \frac{P(EB)\ P(B)}{P(EB)} = P(B)$$

P(Bat | Eat Bamboo) = P(B) = 0.03

By looking at Bayesian Network also we can say that P(Bat | Eat Bamboo) = P(Bat) because Bat is conditionally independent of Eat Bamboo

Question 2: A study by McAfee shows that over 97% of consumers were unable to correctly identify phishing scams when presented with them. As digital communication has evolved so has the sophistication of hackers and scammers. In addition to phony phone calls and email phishing scams, text scams have become more common with the proliferation of smartphones and the emergence of text messages as a part of everyday communication. However, many people could not correctly figure out whether a text is a spam text or a ham text. According to Wikipedia, spam: “the use of electronic messaging systems to send unsolicited bulk messages, especially advertising, indiscriminately.”; ham: “E-mail that is generally desired and isn't considered spam.”. So identify when a text message is spam or ham will benefit people in their daily life.

Answer :

Approach: My approach for this question is to tokenize the words. Tokenize mean separating all the words with commas and identifying how many times that word has been repeated or reused and in what kind of messages. Then I will divide the data in training and testing with 80-20 ratio to train and test the model.

***Statistics of the data*:** Data in the form of a set having two columns first column as class of the message like ham or spam and second column will have actual message. Class and Message of the data are tab separated.

Program : In my Program, I am reading the dataset from SMSSpamCollection folder downloaded from the site. To use this unstructured raw data while reading I will convert into dataframe structure of python using pandas and use tab separator. My dataframe object name is UnprocessedData having two columns – Class and Message. This data is still unprocessed but just in structure form. I achieved it by using:

```
EmailData = pd.read_csv(‘SMSSpamCollection', sep='\t', names=["class", "message"])


EmailData.drop_duplicates(inplace=True)
```

To get more insights into the data I printed the shape of the EmailData and found total number of rows in dataset are 5572. Then I removed the duplicate rows and I got 5169 unique rows.

***Processing the raw data*:** Now I have data in dataframe object and we can easily access the data. If we correctly see how our model just cares about the words used in ham or spam, number of times its used and many other things. There are other things also in dataset which model does not really care about like punctation marks in text. It is good to have all the words in lowercase also. Then according to my approach I will tokenize all the words in the messages during this process. This all was achieved in my program using:

```
def process_data(text):

    nosymbol = [c for c in text if c not in string.punctuation]

    nosymbol = ".join(nosymbol)

    tokens = [w for w in nosymbol.split()]

    return tokens
```

***Feature Extraction*:** We now have data, we have a processing method to preprocess the data before using. We now need a method which will actually use that preprocess method on the data and give features of the data. By features we mean unique words, number of the time it is coming in the message. This will kind of create a token matrix for us which gives us details of all the messages in the data that this message have this word and it came this number of times. For this we will use inbuilt method CountVectorizer from python library sklearn.feature_extraction.text.

Why CountVectorize? - I chose to use CountVectorize instead of DictVectorizer, FeatureHasher and other methods because it is most relevant to type of data we have. CountVectorize also helps us to achieve occurrence count, it removes words like “a” and “I” itself, we do not separately have to use stop words for these kinds of words. It also helps in converting words into lowercase automatically. It provides best structure that can help us to visualize things, use clustering. In my report I used it as below:

```
messages = CountVectorizer(analyzer=process_data).fit_transform(EmailData['message'])
```

Divide the data in train and test **-** After preprocessing of the data, the last thing before creating the model is to divide the data in train and test the data. I will divide the data in 80% training and 20% testing. Train data means the data which our model will see and learn from it. Test data is for what our model will predict that email is ham or spam. Test data will be unseen to the model so we can if model is actually trained and is not overfitting or underfitting. I achieved this division of data by using train_test_split method from sklearn.model_selection library. Below is the corresponding code:

```
x_train, x_test, y_train, y_test = train_test_split(messages, EmailData['class'], test_size=0.20,
random_state=0)
```

x_train, x_test will have actual message and y_train and y_test will have class label of spam or ham. X_train and y_train data I will use to train my model and tell it that this email will have this corresponding class.

Creating the Model – Now this model will train itself on the data and learn to identify the ham or spam email. There are various models available in Naïve Bayes classifiers, Support Vector Machines and others. Since this is classification exampled and supervised learning, I used SVC classifier from Support Vector Machines library of python.

Why SVC? – One of the reasons for using SVM as a classifier is that I have converted my data in vector format using CountVectorize feature extraction. Although it is suitable for other models also but it will best work with SVM. They create best decision boundary when classifying a vector. SVM takes interactions between different features into consideration but Naïve Bayes classifier mostly treats feature as independent. Additionally I tried training the model with SVC and MultinomialNB and SVC gave me better accuracy rates. I created and trained the model like below. Fit method is used to train the model.

```
classifier = SVC()
classifier.fit(x_train, y_train)
```

Evaluate Model – Now we have trained the model we can test or evaluate the model. By evaluating the model means we will check what model predicts for the email it has never seen. Will it be able to correctly identify that the email is a spam or a ham. For the classifier we have predict method, in which we will pass the x_test data from the data divide step. I did that like:

```
pred = classifier.predict(x_test)
```

Now whatever the output my classifier have predicted for the unseen images we have those values in the pred variable. To find the values are correct or not we can compare it with the correct answers in y_test. Instead of doing that manually we can use many inbuilt and better features provided by python like classification_report, confusion_matrix, accuracy_score from sklearn.metrics. I achieved that with below code:

```
print("Classification Report: ")
print(classification_report(y_test, pred))



print("Confusion Matrix:")
print(confusion_matrix(y_test, pred))



print("Accuracy:")
print(accuracy_score(y_test, pred))
```

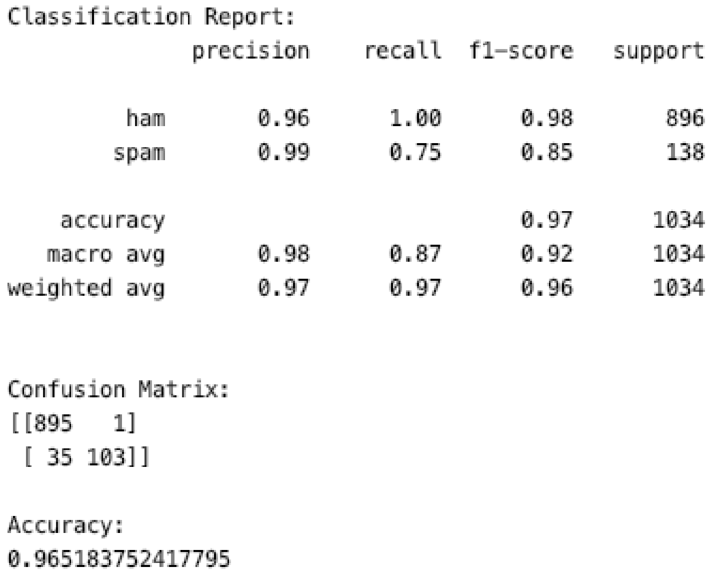
Classification report: It gives idea about various classification metrics of the classifier like precision, recall, F1- Score, Support for all the different classes in the data.

Precision : It is the ability of a classifier not to label an instance positive that is actually negative
Recall: How many all positive were found
F1 Score : It is mean of precision and recall
Support: It is the number of actual occurrences of the class in the specified dataset to see distribution of the data for different classes.

Confusion Matrix: It is a way of tabulating the number of misclassifications. This means that number of predicted classes which were in a wrong classification based on true classes.

Accuracy: It gives correctly identified classes in complete data by the model.

Below is my output from the SVC model I trained:



```
Classification Report:
              precision    recall  f1-score   support

         ham       0.96      1.00      0.98       896
        spam       0.99      0.75      0.85       138

    accuracy                           0.97      1034
   macro avg       0.98      0.87      0.92      1034
weighted avg       0.97      0.97      0.96      1034


Confusion Matrix:
[[895   1]
 [ 35 103]]

Accuracy:
0.965183752417795
```

Extra Credit(up to 20 points): If you meet more problems when you develop, please also put them in the report and explain why and how you solve it.

Answer: Machine learning is a new concept to me and all the things were quite a challenge for me to understand. Although I was familiar with Python but I was not familiar with libraries like Pandas, Sklearn, type of models available, training the model etc. Because of that I did find many problems to write the program but below are few of the problems which consumed most of my time.

Problem 1 – ***Reading the Data****:* After looking and understanding the dataset I was not sure how to read it in my program, what kind of data format does machine learning algorithm needs. I had created my custom array data structure and was using simply reading the raw data from file SMS dataset file. I was not able to understand the tab separator fact and that I have to create two different columns for class and actual message. I was treating them as one complete thing. Then again I created a custom 2D data structure and was reading data in two different columns, but I was not able to separate class from the actual message given in the dataset.

Solution 1 – Then I observed the data carefully and read from actual SMS Dataset website, I came across the fact that data separated by tab separator which I was not able to figure out myself. After that I was going through machine learning libraries of Python and I found Pandas library which I can use to create a Data Frame, it will read the dataset file also. Then I can use this dataframe object in rest of the program. All this time I was creating my own 2D data structure and struggling to read the data, this all was one with just one statement later on.

Problem 2 – ***Parsing the Data:*** I have never worked with a text dataset before and I did not know that we have to parse it, how to parse the data. I was not aware of the fact that we need to remove symbols, there are some stop words like “a”, “i” which we need to remove before giving the actual words to the model. After I read the data in DataFrame object I was directly trying to use that to fit the model and it was not working. T

Solution 2 – Then I first had to understand that what does natural language processing involves, what are things we need to remove, how data structure is going to be. I came across many python libraries again like Natual Language Tool Kit, Sklearn etc. Then I understood that we have to remove all the symbols like comma or other punctuation marks etc. I did that by using String library of Python. All this time I was struggling to understand the data structure format which we need to train the model.

Problem 3 – ***Feature Extraction:*** Feature Extraction was another big hurdle in my program, mostly because if someone has not worked with text data they will not be able to figure out what is feature extraction, what tokenization is why do we need it. It was completely new concept for me as well. This is again because I was trying to understand the data structure which we need to put into the model, how will model actually learn, how model will distinguish between different words and find which belongs to ham or spam.

Solution 3 - After going through lecture videos by professors and some more research on NLP, I came across the fact of tokenization and found sklearn feature extraction library. There were different options to use but then I read that for the type of data we have count vectorize works the best and I used that in my preprocessing method and finally my data was preprocessed and ready to use. After this step it all started making sense of what needs to be done.

Problem 4 – ***Model selection*** **–** This was not actually a problem but something I did spend a lot of time to read about different classifier and trying to run my model on them to see which give me better accuracy. I tried different SVM and Naïve Bayes model. Finally I used SVC model from SVM. Finding different way to measure the performance was also a challenge but sklean library made it real easy/