AI Assignment 5 Report

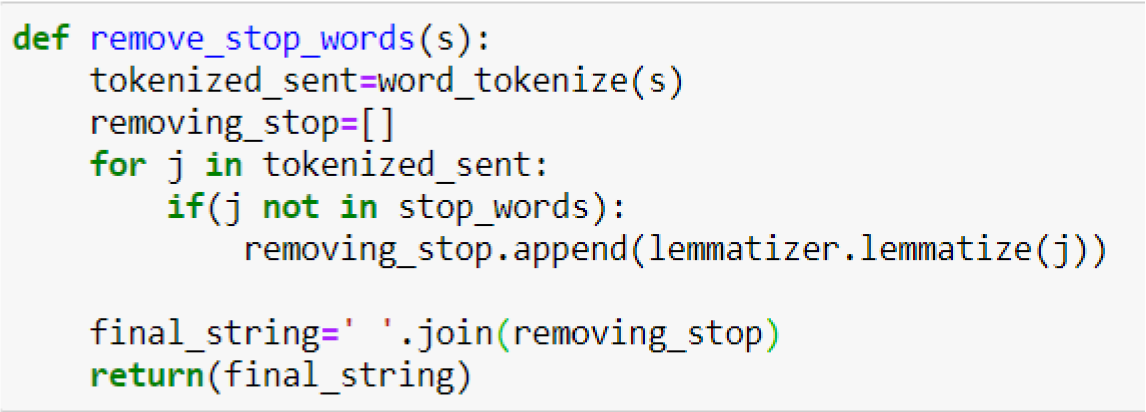
Name: Bhavya Narang

Roll No: 2019462

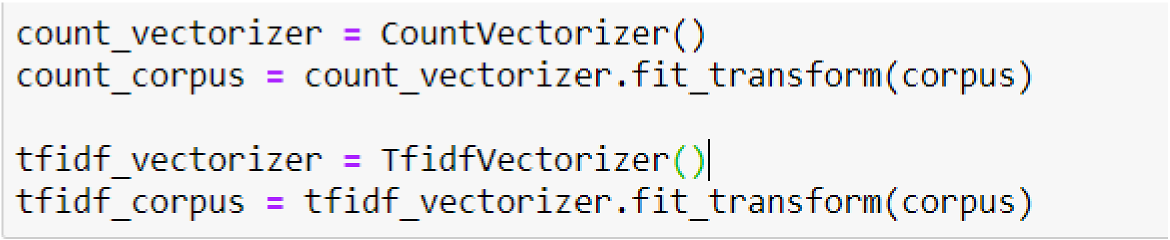
Steps:

Firstly I prepared data to train my models that I am using which are Naïve Bayes and MLP Classifier (ANN).

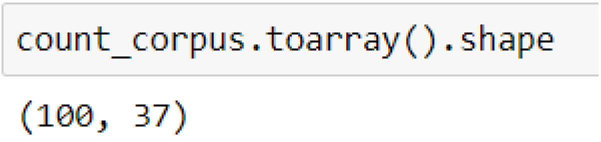
Then using this data I prepared count vectorizer and tfidf vectorizer and converted this text data to sparse encodings after removing stop words and lemmatizing the input tokens.



```
def remove_stop_words(s):
    tokenized_sent=word_tokenize(s)
    removing_stop=[]
    for j in tokenized_sent:
        if(j not in stop_words):
            removing_stop.append(lemmatizer.lemmatize(j))

    final_string=' '.join(removing_stop)
    return(final_string)
```

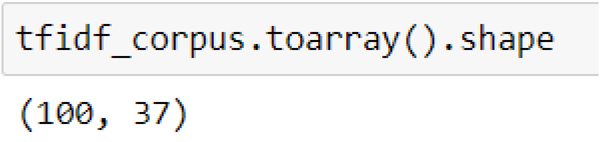


```
count_vectorizer = CountVectorizer()
count_corpus = count_vectorizer.fit_transform(corpus)

tfidf_vectorizer = TfidfVectorizer()
tfidf_corpus = tfidf_vectorizer.fit_transform(corpus)
```



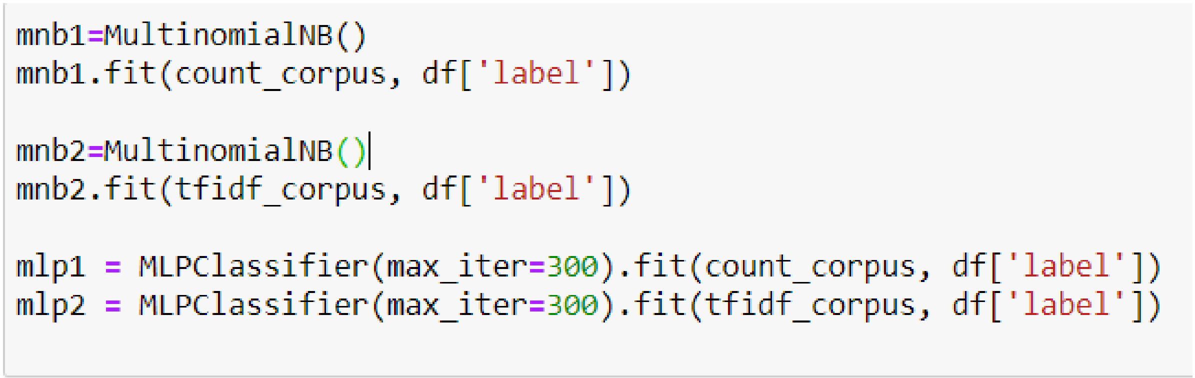
```
count_corpus.toarray().shape
```

```
(100, 37)
```

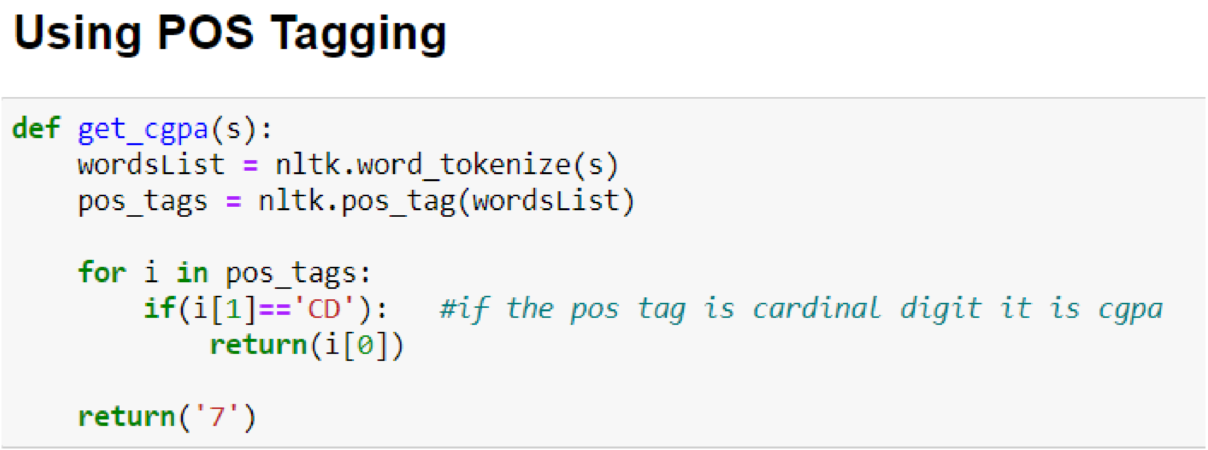


```
tfidf_corpus.toarray().shape
```

```
(100, 37)
```

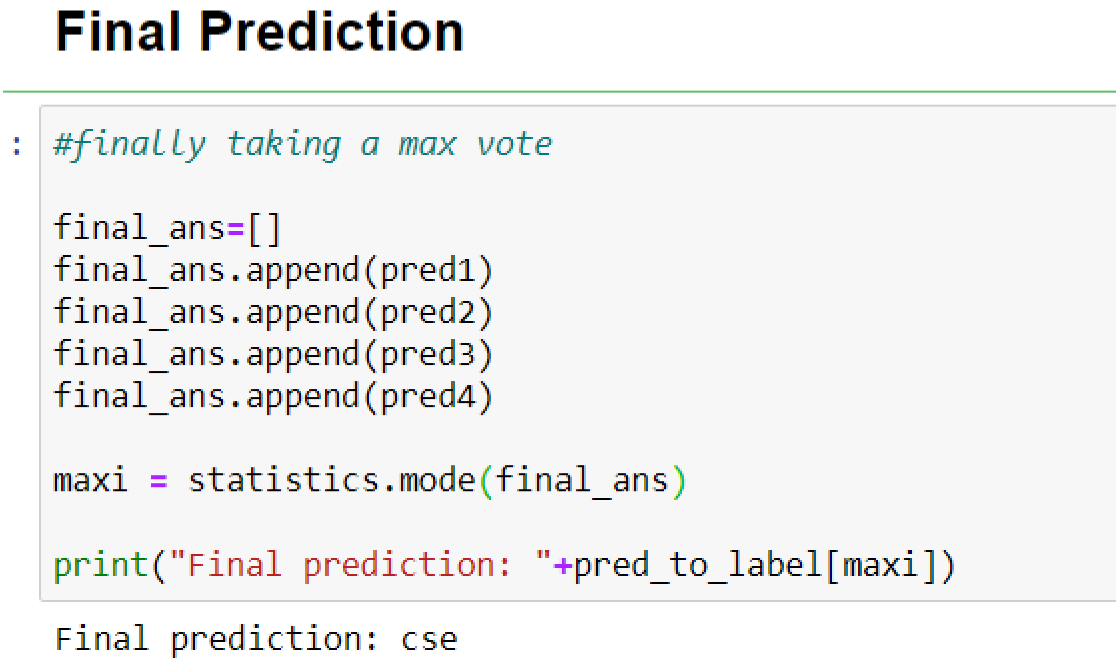


```
mnb1=MultinomialNB()
mnb1.fit(count_corpus, df['label'])

mnb2=MultinomialNB()
mnb2.fit(tfidf_corpus, df['label'])

mlp1 = MLPClassifier(max_iter=300).fit(count_corpus, df['label'])
mlp2 = MLPClassifier(max_iter=300).fit(tfidf_corpus, df['label'])
```



Using POS Tagging

```
def get_cgpa(s):
    wordsList = nltk.word_tokenize(s)
    pos_tags = nltk.pos_tag(wordsList)

    for i in pos_tags:
        if(i[1]=='CD'):    #if the pos tag is cardinal digit it is cgpa
            return(i[0])

    return('7')
```



Final Prediction

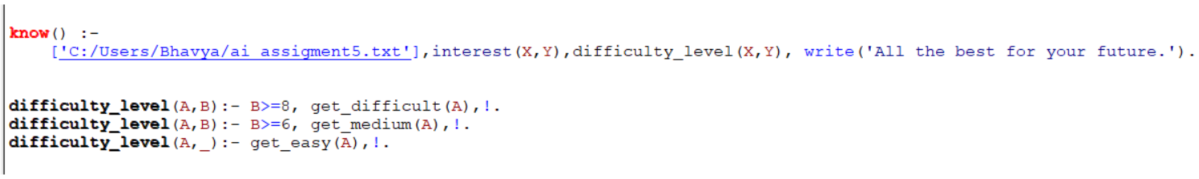
```
#finally taking a max vote

final_ans=[]
final_ans.append(pred1)
final_ans.append(pred2)
final_ans.append(pred3)
final_ans.append(pred4)

maxi = statistics.mode(final_ans)

print("Final prediction: "+pred_to_label[maxi])
```

```
Final prediction: cse
```

Finally I asked user to enter grade and interest in academics and predicted using all 4 models and then used a maxvote of

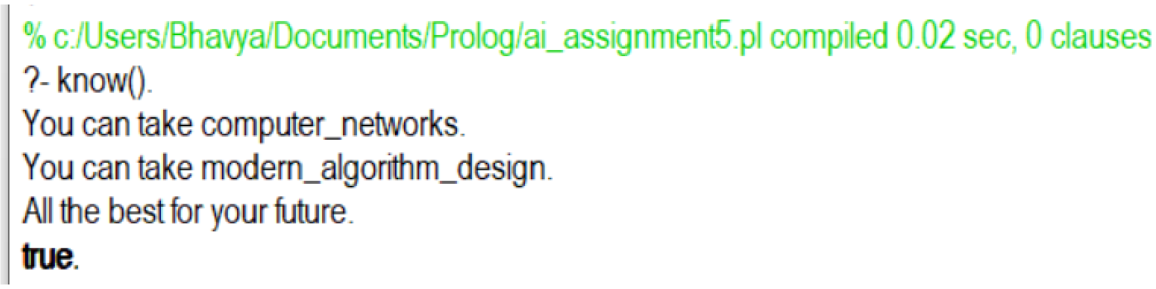
4 models and saved it in a text file to feed to my prolog program.

Prolog reading stored fact:

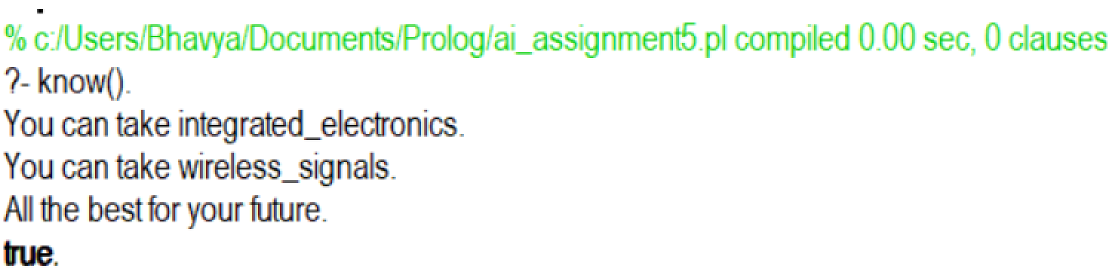


```
know() :-
    ['C:/Users/Bhavya/ai assigment5.txt'],interest(X,Y),difficulty_level(X,Y), write('All the best for your future.').


difficulty_level(A,B):- B>=8, get_difficult(A),!.
difficulty_level(A,B):- B>=6, get_medium(A),!.
difficulty_level(A,_):- get_easy(A),!.
```

Prolog output:



```
% c:/Users/Bhavya/Documents/Prolog/ai_assignment5.pl compiled 0.02 sec, 0 clauses
?- know().
You can take computer_networks.
You can take modern_algorithm_design.
All the best for your future.
true.
```



```
% c:/Users/Bhavya/Documents/Prolog/ai_assignment5.pl compiled 0.00 sec, 0 clauses
?- know().
You can take integrated_electronics.
You can take wireless_signals.
All the best for your future.
true.
```