Natural Language Processing

```
# Natural Language Processing

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Importing the dataset
dataset = pd.read_csv(r"D:\Samson - All Data\Naresh IT Institute\New folder\Restaurant_Reviews.tsv", delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
#nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer

corpus = []

for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
```

```
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values


# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)


from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(X_train, y_train)


# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)

from sklearn.metrics import accuracy_score
```

```
ac = accuracy_score(y_test, y_pred)

print(ac)


bias = classifier.score(X_train,y_train)

bias


variance = classifier.score(X_test,y_test)

variance


#=====================================================================
==================================

'''

CASE STUDY --> model is underfitted  & we got less accuracy


1> Implementation of tfidf vectorization , lets check bias, variance, ac, auc, roc

2> Impletemation of all classification algorihtm (logistic, knn, randomforest, decission tree,
svm, xgboost,lgbm,nb) with bow & tfidf

4> You can also reduce or increase test sample

5> xgboost & lgbm as well

6> you can also try the model with stopword



6> then please add more recores to train the data more records

7> ac ,bias, varian - need to equal scale ( no overfit & not underfitt)


'''
```