Programme Name: BCS
Course Code: CSC 3200
Course Name: ARTIFICIAL INTELLIGENCE
Group Project No. : 1
Date of Submission: 9th September 2020

Submitted By: **Submitted To:**

Student Names:Rabin Lamichhane, Faculty Name: Prakash Chandra
Prabin Silwal,
Sanjiv Shrestha,
Sushant Rayamajhi Department: P.O
IUKL ID: 041801900053,
041801900049,
041801900062,
041801900068
Semester: 5th Sem
Intake: June 2018

Introduction

Fake news has quickly become a society problem, being used to propagate false or rumor information in order to change peoples behavior. When news comes on internet, it's difficult to differentiate whether the news are fake or true. In order to solve this problem, we use fake news detection to detect the mislead information. The main purpose of this system will be detecting Fake News over Original or legitimate news. The system will be fed with all kinds of news or without filter, out of which the system will detect the false or fake news and generate the result. The main objectives of the system will be:

- To find fake or false news over truth or legit news.
- To help users get the correct information only.
- To reduce the fraud activities via news.

In this system, we'll be using supervised machine algorithm to detect the fake news. That means we'll predict the fake news based on the data given. We'll take the data sets from Kaggle to train and test the fake news. We'll be using decision tree algorithm to identify the fake news over the original news. We'll using the countvectorizer to convert text into numbers that computer understand. We'll using confusion matrix to measure the accuracy of the system.

Feature extractions

News Content Features

Now that fake news has been defined and the target has been set, it is needed to analyse what features can be used in order to classify fake news. Starting by looking at news
content, it can be seen that it is made of three principal raw components:

- **Headline**: Short summary of the news content that try to attract the reader.
- **Body Text**: The actual text content of the news.
- **Label**: Integrates whether news are true or fake.

Fake News Detection Flowchart

Text Data(Training)

Text Data(Testing)

Training Sample

Preprosessing

Test Sample

Count features

DecisionTreeClassifier

User Input

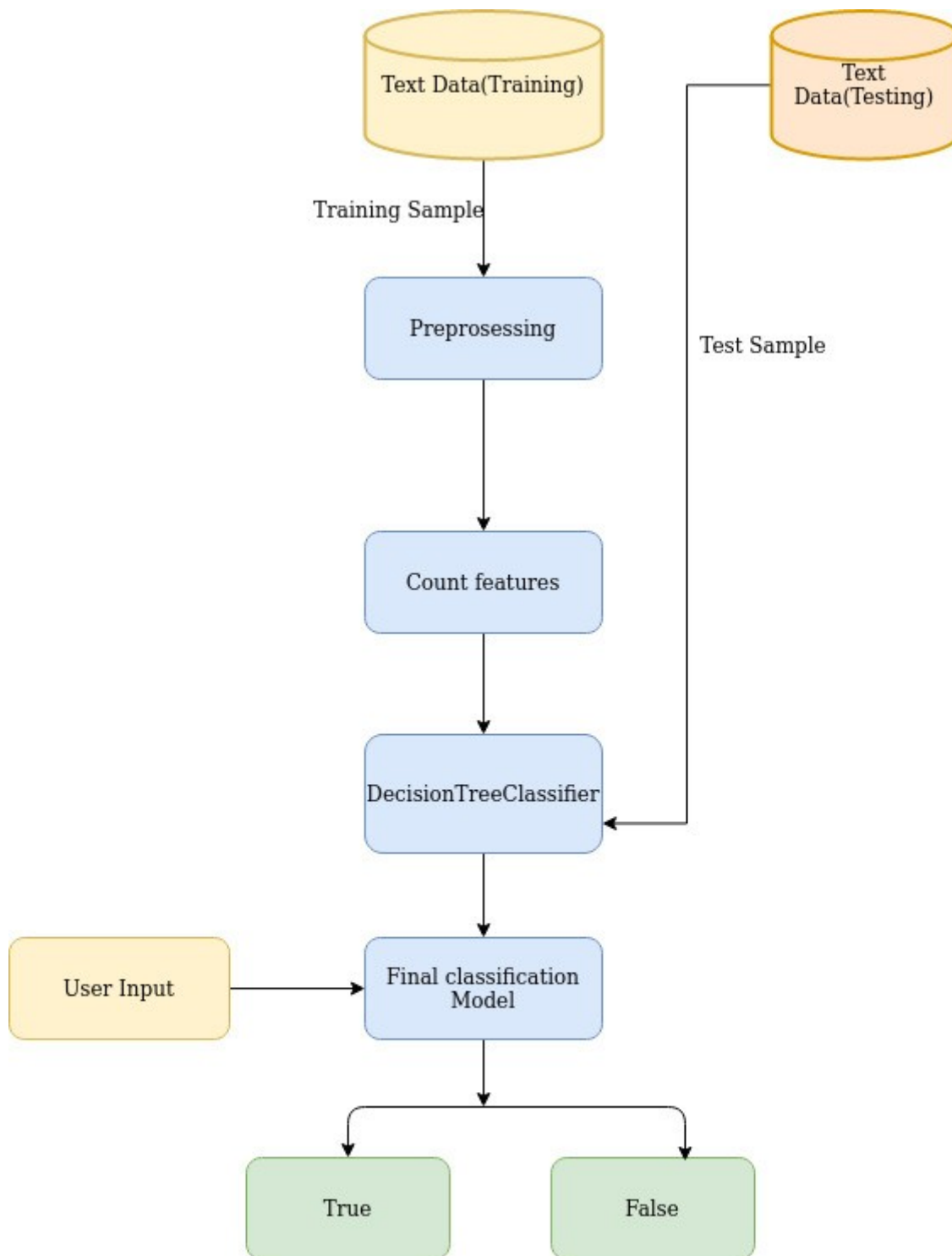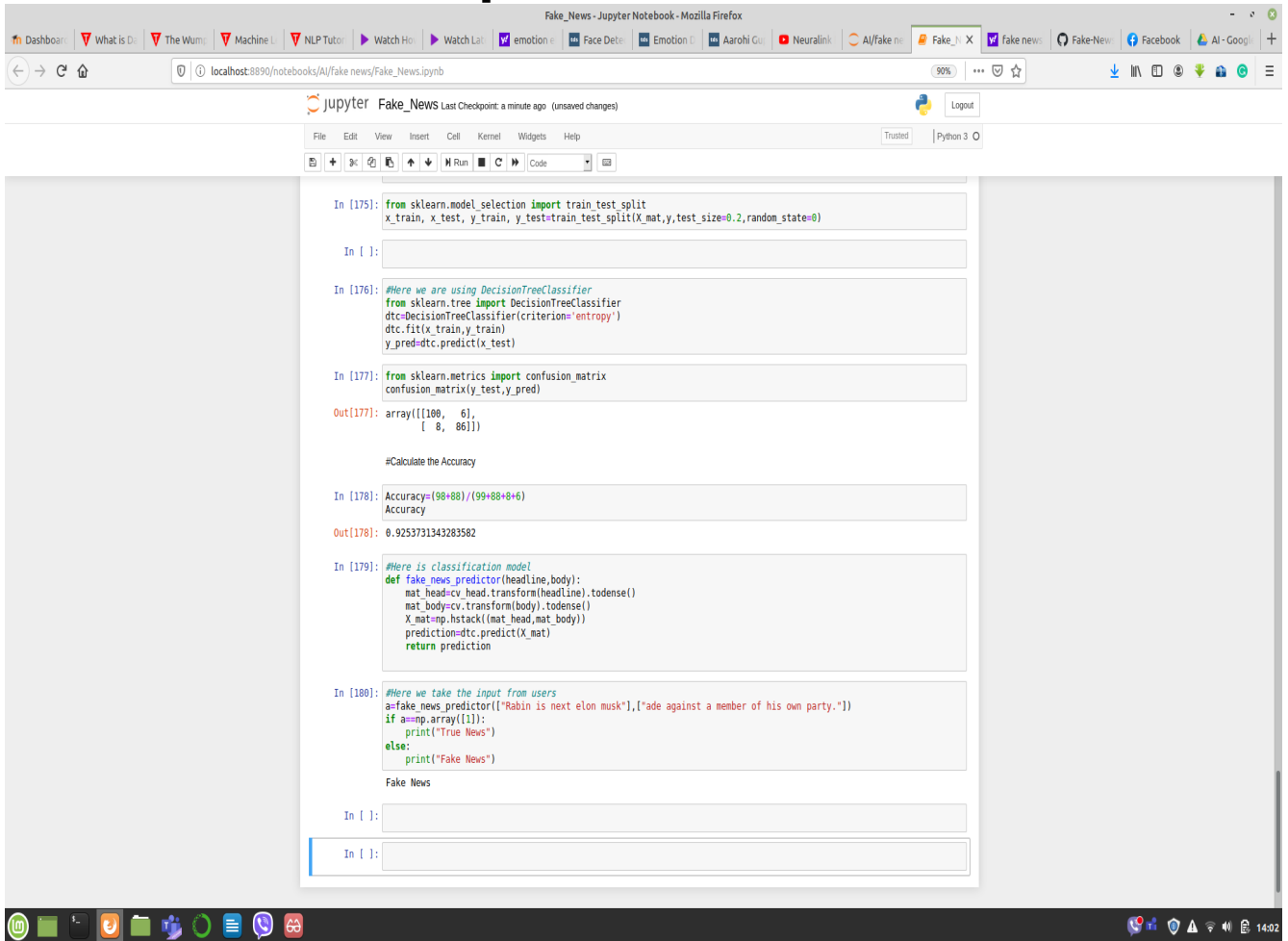Final classification Model

True

False

Fig:Fake News Detection Flow Chart

Output Screenshots

```
In [175]: from sklearn.model_selection import train_test_split
          x_train, x_test, y_train, y_test=train_test_split(X_mat,y,test_size=0.2,random_state=0)
```

```
In [ ]:
```

```
In [176]: #Here we are using DecisionTreeClassifier
          from sklearn.tree import DecisionTreeClassifier
          dtc=DecisionTreeClassifier(criterion='entropy')
          dtc.fit(x_train,y_train)
          y_pred=dtc.predict(x_test)
```

```
In [177]: from sklearn.metrics import confusion_matrix
          confusion_matrix(y_test,y_pred)
```

```
Out[177]: array([[100,   6],
                 [  8,  86]])
```

#Calculate the Accuracy

```
In [178]: Accuracy=(98+88)/(99+88+8+6)
          Accuracy
```

```
Out[178]: 0.9253731343283582
```
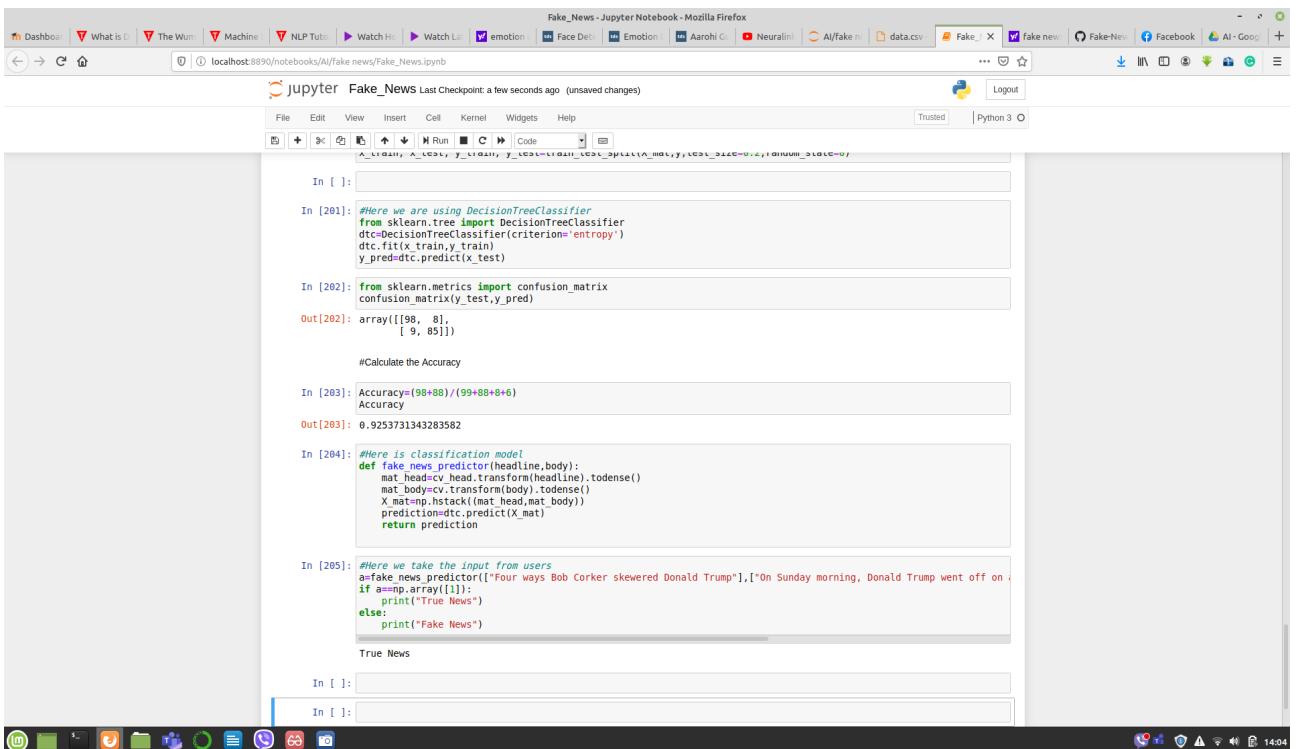
```
In [179]: #Here is classification model
          def fake_news_predictor(headline,body):
              mat_head=cv_head.transform(headline).todense()
              mat_body=cv.transform(body).todense()
              X_mat=np.hstack((mat_head,mat_body))
              prediction=dtc.predict(X_mat)
              return prediction
```

```
In [180]: #Here we take the input from users
          a=fake_news_predictor(["Rabin is next elon musk"],["ade against a member of his own party."])
          if a==np.array([1]):
              print("True News")
          else:
              print("Fake News")
```

Fake News

```
In [ ]:
```

```
In [ ]:
```

```
x_train, x_test, y_train, y_test=train_test_split(X_mat,y,test_size=0.2,random_state=0)
```

```
In [ ]:
```

```
In [201]: #Here we are using DecisionTreeClassifier
          from sklearn.tree import DecisionTreeClassifier
          dtc=DecisionTreeClassifier(criterion='entropy')
          dtc.fit(x_train,y_train)
          y_pred=dtc.predict(x_test)
```

```
In [202]: from sklearn.metrics import confusion_matrix
          confusion_matrix(y_test,y_pred)
```

```
Out[202]: array([[98,  8],
                 [ 9, 85]])
```

#Calculate the Accuracy

```
In [203]: Accuracy=(98+88)/(99+88+8+6)
          Accuracy
```

```
Out[203]: 0.9253731343283582
```

```
In [204]: #Here is classification model
          def fake_news_predictor(headline,body):
              mat_head=cv_head.transform(headline).todense()
              mat_body=cv.transform(body).todense()
              X_mat=np.hstack((mat_head,mat_body))
              prediction=dtc.predict(X_mat)
              return prediction
```

```
In [205]: #Here we take the input from users
          a=fake_news_predictor(["Four ways Bob Corker skewered Donald Trump"],["On Sunday morning, Donald Trump went off on
          if a==np.array([1]):
              print("True News")
          else:
              print("Fake News")
```

True News

```
In [ ]:
```

```
In [ ]:
```

Appendix

```python
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
#Read the data From Dataset
news=pd.read_csv("data.csv")
# shows the head
news.head()
# shows the body
news.Body[0]
#remove the URL column
news=news.drop(['URLs'],axis=1)
news.head()
news.isnull().sum()
news.describe()
news=news.dropna()
news.isnull().sum()
#Take only 1000 News Headlines
news=news[0:1000]
x=news.iloc[:,:-1].values
y=news.iloc[:,-1].values
x[0]
y[0]
cv=CountVectorizer(max_features=5000)
mat_body=cv.fit_transform(x[:,1]).todense()
mat_body
cv_head=CountVectorizer(max_features=5000)
mat_head=cv_head.fit_transform(x[:,0]).todense()
mat_head
X_mat=np.hstack((mat_head,mat_body))
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X_mat,y,test_size=
0.2,random_state=0)
#Here we are using DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier(criterion='entropy')
dtc.fit(x_train,y_train)
y_pred=dtc.predict(x_test)
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

```python
#Calculate the Accuracy
Accuracy=(98+88)/(99+88+8+6)
Accuracy
#Here is classification model
def fake_news_predictor(headline,body):
    mat_head=cv_head.transform(headline).todense()
    mat_body=cv.transform(body).todense()
    X_mat=np.hstack((mat_head,mat_body))
    prediction=dtc.predict(X_mat)
    return prediction
#Here we take the input from users
a=fake_news_predictor(["Four ways Bob Corker skewered
Donald Trump"],["On Sunday morning, Donald Trump went off
on a Twitter tirade against a member of his own party."])
if a==np.array([1]):
    print("True News")
else:
    print("Fake News")
```