Final Project :ML Classification-News Prediction(Fake or Real)

- IBM Machine Learning Profession Certificate
- By Sadhana Jarag

Content

- Dataset Description
- Main Objective of Analysis
- Applying various Classification models
- Machine learning analysis and findings
- Model Flaws and advance step

Data Description Section

Introduction :

"Misinformation can feel like an insurmountable problem, something that only tech giants and social media platforms can solve. But actually, we can each take steps to help stop it spreading, and these actions will have a direct impact on our own communities, friends and family.” – Jen Thomas, Creative Producer

Why News are Important:

Without the news, people would only be able to find out what was happening by asking people who had first-hand knowledge. This would massively reduce the information people would have about the world around them. The fact the news allows people to get up-to-date accounts of recent events is a major reason why it is important.

Why Identifying Fake News Is Important:

It is important to be able to spot fake news because people can be easily mislead by anything the media or someone can say about a topic also they may receive a bias opinion on a topic than getting both sides of the story. News articles and Columnist create fake news to gain more supporters, to spread hoax and lies, or to have more people read there article. Fake news is able to turn people against each other and makes it harder to know what is true or was is false an example of this can be with the Pacific Northwest Tree Octopus story which was a fake news article and was a form of click bait. It also ties into history for example if people require research on something that happened years ago and they find news articles with wrong information and cite them as a source of factual information people will get confuse and believe it.

Project Introduction

- In this project we have collect the data of news article with respect to their titles, authors and the summary kind of information as in text feature.
- Here we are trying to identify the fake and real news from the given data.

Dataset Description :Part 1

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1

Dataset Description :Part 2

Data Description:

- Here the data given is all about the news and we need to find out whether the data is fake or real.
- We have four input feature and one output feature which is telling that data is real or fake.

Feature info:

- 1. id: unique id for a news article.
- 2. title: the title of the news article.
- 3. author: author of the news article.
- 4. text: the text of the article that could be incomplete.
- 5. label: a label that marks whether the news article is real or fake.

  1 => fake news

  0 => real news

Dataset Description

	title	author	text
count	20242	18843	20761
unique	19803	4201	20386
top	Get Ready For Civil Unrest: Survey Finds That ...	Pam Key	
freq	5	243	75

Dataset Description

Check for duplicate Value: No duplicate Value found

id	title	author	text	label

Check for Null Values : Null values are in the datasets

Checking the null values in training data.

```
[166]: train_data.isnull().sum()

:[166]: id          0
        title     558
        author   1957
        text       39
        label       0
        dtype: int64
```

<u>**EDA :Exploratory Data Analysis**</u>

Main Objective of Analysis

- In this section I am showing the analysis which I have done on the data and find out some patterns related to data with relation target feature.
- After that I will apply different classification model and some hyperparameter tunning to get best predictive model in term of accuracy .
- In addition I will suggest the what more work we can perform on the data also some flaws of the model.

EDA :Exploratory Data Analysis

```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   id      20800 non-null  int64
 1   title   20242 non-null  object
 2   author  18843 non-null  object
 3   text    20761 non-null  object
 4   label   20800 non-null  int64
dtypes: int64(2), object(3)
memory usage: 812.6+ KB
```

- Numeric Feature : 2
- Categorical Feature :3
- Having Null values in Dataset
- Total Feature :5
- Input Feature :4
- Output Feature :1

EDA :Exploratory Data Analysis

```
[52]: train_data[(train_data['author']=='Anonymous') & (train_data['label']==1)]
```

[52]:

	id	title	author	text	label
120	120	NaN	Anonymous	Same people all the time , i dont know how you...	1
140	140	NaN	Anonymous	There is a lot more than meets the eye to this...	1
347	347	LesserOfTwoEvilism	Anonymous	2016 presidential campaign by Matt Sedillo \nH...	1
376	376	Realities Faced by Black Canadians are a Natio...	Anonymous	Tweet Widget by Robyn Maynard \nCanada, includ...	1
562	562	NaN	Anonymous	Field is correct about the 8a companies and Tr...	1
...	...	...	...	...	...
18720	18720	NaN	Anonymous	There is plenty of proof the machines are rigg...	1
18903	18903	NaN	Anonymous	There are lots of diiferent truths , when i he...	1
19171	19171	NaN	Anonymous	Same people all the time , i dont know how you...	1
20142	20142	The Deteriorating Situation in Ethiopia	Anonymous	Tweet Widget by Yohannes Woldemariam \nThe min...	1
20749	20749	Realities Faced by Black Canadians are a Natio...	Anonymous	Tweet Widget by Robyn Maynard \nCanada, includ...	1

77 rows × 5 columns

- -Insights
- Author with name ‘Anonymous’ having 77 rows in total and all are belong to category/label =1 means fake news.

EDA :Exploratory Data Analysis

3]:
```
train_data[train_data['title'].isnull()]
```

3]:

	id	title	author	text	label
53	53	NaN	Dairy✓TRUMP	Sounds like he has our president pegged. What ...	1
120	120	NaN	Anonymous	Same people all the time , i dont know how you...	1
124	124	NaN	SeekSearchDestory	You know, outside of any morality arguments, i...	1
140	140	NaN	Anonymous	There is a lot more than meets the eye to this...	1
196	196	NaN	Raffie	They got the heater turned up on high.	1
...	...	...	...	...	...
20568	20568	NaN	Cathy Milne	Amusing comment Gary! “Those week!” So, are ...	1
20627	20627	NaN	Ramona	No she doesn’t have more money than God, every...	1
20636	20636	NaN	Dave Lowery	Trump all the way!	1
20771	20771	NaN	Letsbereal	DYN's Statement on Last Week's Botnet Attack h...	1
20772	20772	NaN	beersession	Kinda reminds me of when Carter gave away the ...	1

558 rows × 5 columns

Insights : 558 rows with value ‘NaN’

EDA :Exploratory Data Analysis

```
[54]: train_data[(train_data['title'].isnull()) & (train_data['label']==1)]
```

[54]:

	id	title	author	text	label
53	53	NaN	Dairy✓TRUMP	Sounds like he has our president pegged. What ...	1
120	120	NaN	Anonymous	Same people all the time , i dont know how you...	1
124	124	NaN	SeekSearchDestory	You know, outside of any morality arguments, i...	1
140	140	NaN	Anonymous	There is a lot more than meets the eye to this...	1
196	196	NaN	Raffie	They got the heater turned up on high.	1
...	...	...	...	...	...
20568	20568	NaN	Cathy Milne	Amusing comment Gary! “Those week!” So, are ...	1
20627	20627	NaN	Ramona	No she doesn’t have more money than God, every...	1
20636	20636	NaN	Dave Lowery	Trump all the way!	1
20771	20771	NaN	Letsbereal	DYN's Statement on Last Week's Botnet Attack h...	1
20772	20772	NaN	beersession	Kinda reminds me of when Carter gave away the ...	1

558 rows × 5 columns

- All values in the feature title with ‘NaN’ belongs to label 1 means fake news

EDA :Exploratory Data Analysis

```
[55]: train_data[train_data['author'].isnull()]
```

[55]:

	id	title	author	text	label
6	6	Life: Life Of Luxury: Elton John's 6 Favorite ...	NaN	Ever wonder how Britain's most iconic pop pian...	1
8	8	Excerpts From a Draft Script for Donald Trump'...	NaN	Donald J. Trump is scheduled to make a highly ...	0
20	20	News: Hope For The GOP: A Nude Paul Ryan Has J...	NaN	Email \nSince Donald Trump entered the electio...	1
23	23	Massachusetts Cop's Wife Busted for Pinning Fa...	NaN	Massachusetts Cop's Wife Busted for Pinning Fa...	1
31	31	Israel is Becoming Pivotal to China's Mid-East...	NaN	Country: Israel While China is silently playin...	1
...	...	...	...	...	...
20718	20718	This Is The Best Picture In Human History | Da...	NaN	This Is The Best Picture In Human History By: ...	1
20728	20728	Trump warns of World War III if Clinton is ele...	NaN	Email Donald Trump warned in an interview Tues...	1
20745	20745	Thomas Frank Explores Whether Hillary Clinton ...	NaN	Thomas Frank Explores Whether Hillary Clinton ...	1
20768	20768	Osama bin Laden's older brother rents out luxu...	NaN	Osama bin Laden's older brother rents out luxu...	1
20786	20786	Government Forces Advancing at Damascus-Aleppo...	NaN	#FROMTHEFRONT #MAPS 22.11.2016 - 1,361 views 5...	1

1957 rows × 5 columns

Insights :

We can observe the feature ‘author’ with NaN =1957

EDA :Exploratory Data Analysis

```
56]: train_data[(train_data['author'].isnull()) & (train_data['label']==1)]
```

56]:

	id	title	author	text	label
6	6	Life: Life Of Luxury: Elton John's 6 Favorite ...	NaN	Ever wonder how Britain's most iconic pop pian...	1
20	20	News: Hope For The GOP: A Nude Paul Ryan Has J...	NaN	Email \nSince Donald Trump entered the electio...	1
23	23	Massachusetts Cop's Wife Busted for Pinning Fa...	NaN	Massachusetts Cop's Wife Busted for Pinning Fa...	1
31	31	Israel is Becoming Pivotal to China's Mid-East...	NaN	Country: Israel While China is silently playin...	1
43	43	Can I have one girlfriend without you bastards...	NaN	Can I have one girlfriend without you bastards...	1
...	...	...	...	...	...
20718	20718	This Is The Best Picture In Human History | Da...	NaN	This Is The Best Picture In Human History By: ...	1
20728	20728	Trump warns of World War III if Clinton is ele...	NaN	Email Donald Trump warned in an interview Tues...	1
20745	20745	Thomas Frank Explores Whether Hillary Clinton ...	NaN	Thomas Frank Explores Whether Hillary Clinton ...	1
20768	20768	Osama bin Laden's older brother rents out luxu...	NaN	Osama bin Laden's older brother rents out luxu...	1
20786	20786	Government Forces Advancing at Damascus-Aleppo...	NaN	#FROMTHEFRONT #MAPS 22.11.2016 - 1,361 views 5...	1

1931 rows × 5 columns

- Out of 1957 we can see here 1931 values belong to label =1 means 'Fake News'

EDA :Exploratory Data Analysis

```
train_data.groupby(train_data['author'])['label'].sum()
```

```
author
# 1 NWO Hatr                                                                                      17
-NO AUTHOR-                                                                                       54
10 Habits That Will Make Your Life Easier &amp; More Peaceful - Wellness Solutions                 1
10 More Beautiful Images That Remind You We Still Live In A Beautiful World, With Beautiful People - Upside Down Media    1
10 Movies That Could Change Your Understanding Of Life - Upside Down Media                         1
                                                                                                  ..
1                                                                                       تيري ميسان
1                                  جنگ ارزی آمریکا علیه ایران / مورد مطالعاتی سال 1390 - کدآمایی
1                                                                                سعيد هلال الشريفي
“SHOOT FIRST ASK QUESTIONS LATER” : WHAT HAPPENS TO A UFO WHEN TRACKED ON MILITARY RADAR - Black Barth    1
“Shoot First Ask Questions Later” : What Happens To A UFO When Tracked on Military Radar – Mystical Shire    1
Name: label, Length: 4201, dtype: int64
```

- Here I just want to know how many authors are with the same label
- We can see the result above

EDA :Exploratory Data Analysis

```
[345]: df[train_data['author'] == '# 1 NWO Hatr'].count()
```

```
[345]: id        17
       title     17
       author    17
       text      17
       label     17
       dtype: int64
```

```
]: train_data[(train_data['author'] == '# 1 NWO Hatr') & (train_data['label'] == 1)].count()
```

```
]: id        17
   title     17
   author    17
   text      17
   label     17
   dtype: int64
```

Insights :

We author with name "# 1 NOW Hatr " with label 1 only.

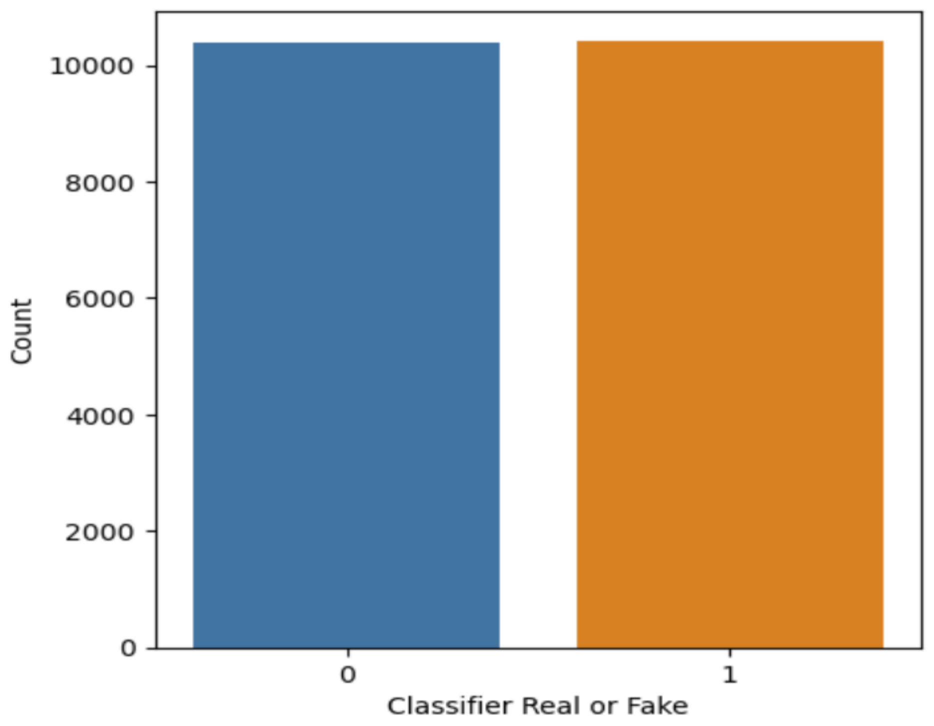
EDA :Exploratory Data Analysis

```
2]: train_data[(train_data['author'] == '-NO AUTHOR-')].count()

2]: id        54
    title     54
    author    54
    text      54
    label     54
    dtype: int64
```

```
9]: train_data[(train_data['author'] == '-NO AUTHOR-') & (train_data['label'] == 1)].count()

9]: id        54
    title     54
    author    54
    text      54
    label     54
    dtype: int64
```

Insights : Total 54 entries with author '-NO AUTHOR-' which has label 1 means fake news

EDA :Exploratory Data Analysis



Insights: Both real and fake values are same

Here we can see data is balanced when we consider all values(missing and redundant)

FE : Feature Engineering

Feature Engineering

```
def handle_missing_value(train_data):
    train_data = train_data.fillna(" ")
    return train_data

train = handle_missing_value(train_data)
```

```
n [373]: train.isnull().sum()

ut[373]: id        0
         title     0
         author    0
         text      0
         label     0
         dtype: int64
```

```
n [374]: train.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   id      20800 non-null  int64
 1   title   20800 non-null  object
 2   author  20800 non-null  object
 3   text    20800 non-null  object
 4   label   20800 non-null  int64
dtypes: int64(2), object(3)
memory usage: 812.6+ KB
```

- Here we fill all value with spaces

Feature Engineering

Check Data has imbalanced set or not

```
train['label'].value_counts()
```

```
1    10413
0    10387
Name: label, dtype: int64
```

```
imbalanced_count = (train[train['label']==1].label.count())-(train[train['label']==0].label.count())
```

```
print(imbalanced_count)
```

```
26
```

- As we can see data is not much imbalanced but better to use stratified technique while doing train test split.

Feature Engineering

Creating a variable "title_author" by merging columns "title" and "author"

```
[168]: train["title_author"] = train["title"]+" "+train["author"]
```

```
[146]: train.head(3)
```

[146]:

	id	title	author	text	label	title_author
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1	House Dem Aide: We Didn't Even See Comey's Let...
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0	FLYNN: Hillary Clinton, Big Woman on Campus - ...
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1	Why the Truth Might Get You Fired Consortiumne...

As we see there is more impact of title and author feature for choosing the label/target data so we merge into one cell.

Feature Engineering

Stemming and Data Cleaning

Stemming: This is the process of reducing the words into roots words and that will remove suffix and prefix from the word.

- Steps are followed:

1. Firstly, all the sequences except English characters are removed from the string.
2. Next, to avoid false predictions or ambiguity with upper and lowercase, all the characters in strings are converted to lowercase.
3. Next, all the sentences are tokenized into words.
4. To facilitate fast processing, stemming is applied to the tokenized words.
5. Next, words are joined together and stored in the corpus.

- Note: In this tutorial, we have used “title_author" column for classification task. Also, the loop inside the function runs over all the examples in the title_author column.

Feature Engineering

```
76]: port=PorterStemmer()
     def stem_data(content):
         stemmed_content=re.sub('[^a-zA-Z]',' ',content)
         stemmed_content=stemmed_content.lower()
         stemmed_content=stemmed_content.split()
         stemmed_content=[port.stem(word) for word in stemmed_content if not word in stopwords.words('english')]
         stemmed_content=' '.join(stemmed_content)
         return stemmed_content
```

```
77]: train['title_author']=train['title_author'].apply(stem_data)
```

Machine Learning and Analysis

- In the following analysis will compare between 4 different Classification model Logistic Regression, Naïve Bayes, Decision Tree and XGBoost in term of predicting the news whether its fake or real and as result obtained, we will decide whether we need to do hyperparameter tunning or not.
- Here we are using CountVectorizer and TF-IDF vectorizer to convert the words/sentences into matrix.
- In building a news detector, two intuitive considerations can be made to optimize the model accuracy. This model focuses on maximizing the vectorization of the input data set.
- Using a countVectorizer for news dataset that have similar news articles. Intuitively, it means if more similar news article are found int the dataset, it is more likely these news article will be valid or false depending on the majority label.
- The TFIDF works better in a dataset that has a lot of unique news article as it is able to highlight the weight of these unique words. Here I have done with both and get around similar accuracy.

Data Spitting: I have done with analysis with both count vectorizer as well as with TF-IDF vectorization.

Train Test split with TF-IDF vectorizer

```
4]: # train test split
tfidf = TfidfVectorizer(ngram_range =(2,2), max_features = 20000)
X_tf = tfidf.fit_transform(X).toarray() # matrix creation- words as columns, sentences as rows
X_train, X_test, y_train, y_test = train_test_split(X_tf, y, test_size =0.25, random_state =0,stratify=y)
```

Train Test spit with Countvectorizer

```
3]: # train test split
cv = CountVectorizer(ngram_range =(2,2), max_features = 20000)
X_cv= cv.fit_transform(X).toarray() # matrix creation- words as columns, sentences as rows
X_train1, X_test1, y_train1, y_test1 = train_test_split(X_cv,y, test_size =0.32, random_state =10,stratify=y)
```

Model Evaluation Function

Function to get all value with resepct to called machine algo by using Countvectorizer train test split

```
316]: def train(model, model_name):
          model.fit(X_train1,y_train1)
          print(f"Training accuracy of {model_name} is {model.score(X_train1,y_train1)}")
          print(f"testing accuracy of {model_name} is {model.score(X_test1,y_test1)}")
          y_pred1 = model.predict(X_test1)
          print(confusion_matrix(y_test1, y_pred1))
          print(classification_report(y_test1,y_pred1))
          accuracy= accuracy_score(y_test1, y_pred1)
          return accuracy
```

Function to get all value with resepct to called machine algo by using TF-IDF vecorrizer train test split

```
317]: def train_tfidf(model, model_name):
          model.fit(X_train,y_train)
          print(f"Training accuracy of {model_name} is {model.score(X_train,y_train)}")
          print(f"testing accuracy of {model_name} is {model.score(X_test,y_test)}")
          y_pred = model.predict(X_test)
          print(confusion_matrix(y_test, y_pred))
          print(classification_report(y_test,y_pred))
          accuracy= accuracy_score(y_test, y_pred)
          return accuracy
```

Model 1 :Logistic Regression Model

1. Countvectorizer : LogisticRegression

```
3]: model1_accuracy = train(LogisticRegression(),'LogisticRegression')
```

```
Training accuracy of LogisticRegression is 0.9959615384615385
testing accuracy of LogisticRegression is 0.9901923076923077
[[2552   45]
 [   6 2597]]
              precision    recall  f1-score   support

           0       1.00      0.98      0.99      2597
           1       0.98      1.00      0.99      2603

    accuracy                           0.99      5200
   macro avg       0.99      0.99      0.99      5200
weighted avg       0.99      0.99      0.99      5200
```

```
]: print("Accuracy of Logistic Regression on Count Vectorizer data",model1_accuracy*100)
```

```
Accuracy of Logistic Regression on Count Vectorizer data 99.01923076923077
```

Model 2 : Naïve Bayes Model

2. Countvectorizer : MultinomialNB

```
: model2_accuracy = train(MultinomialNB(),'MultinomialNB')
```

```
Training accuracy of MultinomialNB is 0.9916025641025641
testing accuracy of MultinomialNB is 0.9738461538461538
[[2585   12]
 [ 124 2479]]
              precision    recall  f1-score   support

           0       0.95      1.00      0.97      2597
           1       1.00      0.95      0.97      2603

    accuracy                           0.97      5200
   macro avg       0.97      0.97      0.97      5200
weighted avg       0.97      0.97      0.97      5200
```

```
: print("Accuracy of Multinomial NB on Count Vectorizer data",model2_accuracy*100)
```

```
Accuracy of Multinomial NB on Count Vectorizer data 97.38461538461539
```

Model 3 :Decision Tree

3. Countvectorizer : DecisionTreeClassifier

```
3]: model3_accuracy = train(DecisionTreeClassifier(),'DecisionTreeClassifier')
```

```
Training accuracy of DecisionTreeClassifier is 0.9999358974358974
testing accuracy of DecisionTreeClassifier is 0.9930769230769231
[[2575   22]
 [  14 2589]]
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2597
           1       0.99      0.99      0.99      2603

    accuracy                           0.99      5200
   macro avg       0.99      0.99      0.99      5200
weighted avg       0.99      0.99      0.99      5200
```

```
]: print("Accuracy of DecisionTree on Count Vectorizer data",model3_accuracy*100)
```

```
Accuracy of DecisionTree on Count Vectorizer data 99.3076923076923
```

Model 4 :XGBoost Classifier

4.Countvectorize : XGBoost Classifier

```
model4_accuracy = train(XGBClassifier(),'XGBoostClassifier')
```

```
Training accuracy of XGBoostClassifier is 0.9892948717948717
testing accuracy of XGBoostClassifier is 0.9875
[[2537   60]
 [   5 2598]]
              precision    recall  f1-score   support

           0       1.00      0.98      0.99      2597
           1       0.98      1.00      0.99      2603

    accuracy                           0.99      5200
   macro avg       0.99      0.99      0.99      5200
weighted avg       0.99      0.99      0.99      5200

Accuracy of Logistic Regression on Count Vectorizer data 98.75
```

Accuracy of Logistic Regression on Count Vectorizer data 98.75

```
print("Accuracy of XGboost on Count Vectorizer data",model4_accuracy*100)
```

```
Accuracy of XGboost on Count Vectorizer data 98.75
```

Model 5 :Logistic Regression with TF-IDF

1. TF-IDF vectorizer :LogisticRegression

```
model1_accuracy_tfidf = train_tfidf(LogisticRegression(),'LogisticRegression')
```

```
Training accuracy of LogisticRegression is 0.9905128205128205
testing accuracy of LogisticRegression is 0.9803846153846154
[[2501   96]
 [   6 2597]]
              precision    recall  f1-score   support

           0       1.00      0.96      0.98      2597
           1       0.96      1.00      0.98      2603

    accuracy                           0.98      5200
   macro avg       0.98      0.98      0.98      5200
weighted avg       0.98      0.98      0.98      5200
```

```
print("Accuracy of Logistic Regression on TF-IDF data",model1_accuracy_tfidf*100)
```

```
Accuracy of Logistic Regression on TF-IDF data 98.03846153846155
```

Model 6 :Naïve Bayes with TF-IDF

2. TF-IDF vectorizer :MultinomialNB

```
[ ]: model2_accuracy_tfidf = train_tfidf(MultinomialNB(),'MultinomialNB')
```

```
Training accuracy of MultinomialNB is 0.9930769230769231
testing accuracy of MultinomialNB is 0.9732692307692308
[[2586   11]
 [ 128 2475]]
              precision    recall  f1-score   support

           0       0.95      1.00      0.97      2597
           1       1.00      0.95      0.97      2603

    accuracy                           0.97      5200
   macro avg       0.97      0.97      0.97      5200
weighted avg       0.97      0.97      0.97      5200
```

```
[ ]: print("Accuracy of Naive Bayes MultiNB on TF-IDF data",model2_accuracy_tfidf*100)
```

```
Accuracy of Naive Bayes MultiNB on TF-IDF data 97.32692307692308
```

Model 7:DecisionTree with TF-IDF

3. TF-IDF vectorizer : DecisionTreeClassifier

```
model3_accuracy_tfidf = train_tfidf(DecisionTreeClassifier(),'DecisionTreeClassifier')
```

```
Training accuracy of DecisionTreeClassifier is 0.9999358974358974
testing accuracy of DecisionTreeClassifier is 0.9940384615384615
[[2576   21]
 [  10 2593]]
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      2597
           1       0.99      1.00      0.99      2603

    accuracy                           0.99      5200
   macro avg       0.99      0.99      0.99      5200
weighted avg       0.99      0.99      0.99      5200
```

```
print("Accuracy of DecisionTree classifier on TF-IDF data",model3_accuracy_tfidf*100)
```

```
Accuracy of DecisionTree classifier on TF-IDF data 99.40384615384616
```

Model 8:XG Boost Classifier with TF-IDF

4.TF-IDF vectorizer : XGBoost Classifier

```
: model4_accuracy_tfidf = train_tfidf(XGBClassifier(),'XGBoostClassifier')
```

```
Training accuracy of XGBoostClassifier is 0.989423076923077
testing accuracy of XGBoostClassifier is 0.9875
[[2537   60]
 [   5 2598]]
              precision    recall  f1-score   support

           0       1.00      0.98      0.99      2597
           1       0.98      1.00      0.99      2603

    accuracy                           0.99      5200
   macro avg       0.99      0.99      0.99      5200
weighted avg       0.99      0.99      0.99      5200
```

```
431]: print("Accuracy of XGB classifier on TF-IDF data",model4_accuracy_tfidf*100)
```

```
Accuracy of XGB classifier on TF-IDF data 98.75
```

Model Accuracy of all model with split TF-IDF

```
]: results_tfidf = pd.DataFrame([["Logistic Regression_tfidf",model1_accuracy],["Naive Bayes_tfidf",model2_accuracy],["Decision Tree
                   ["XGBOOST_tfidf",model4_accuracy]],columns=["Model","Accuracy"])
   results_tfidf
```

]:

	Model	Accuracy
0	Logistic Regression_tfidf	0.980385
1	Naive Bayes_tfidf	0.973846
2	Decision Tree_tfidf	0.993077
3	XGBOOST_tfidf	0.987500

Model Accuracy of all model with split Count Vectorizer

```
: results = pd.DataFrame([["Logistic Regression",model1_accuracy],["Naive Bayes",model2_accuracy],["Decision Tree",model3_accuracy]
                   ["XGBOOST",model4_accuracy]],columns=["Model","Accuracy"])

  results
```

:

	Model	Accuracy
0	Logistic Regression	0.980385
1	Naive Bayes	0.973269
2	Decision Tree	0.993462
3	XGBOOST	0.987500

Model Comparison, Model Flaws ,Advanced Step and further Suggestion

Here is the analysis

- All models are performed well so we need to check with respect to computational time, cost and memory consumption etc. to decide which one we can use.
- In term of simplicity ,we can say that Logistic regression and Naïve Bayes provided high predictive result and at the same time they are simplest and fastest model of the parameter, but Decision tree provided the best result in case of Countvectorizer and TF-IDF vectorizer.
- DecisionTree and Xgboost provided the very good result, but they took longer time to execute that means they need more time to train than other two algorithm.
- Here in this dataset both vectors perform well as we see the Countvectorizer perform well that TF-IDF vectorizer but if we can see the long run basis it is good to use TF-IDF vectorizer.
- At the end of the trade off if we have bigger dataset then there are chances of model can perform differently so in that case we have to use hyperparameter tunning for all the algorithm that we used here now or may be we have to try different machine algorithm to check the performance in compare with all these algorithm.

- You can fine the code at : sadhanajarag/IBM-Certification-Supervised-Machine-Algorithm (github.com)

Thank You!!!!

Supervised Machine Learning :Classification

By Sadhana Jarag