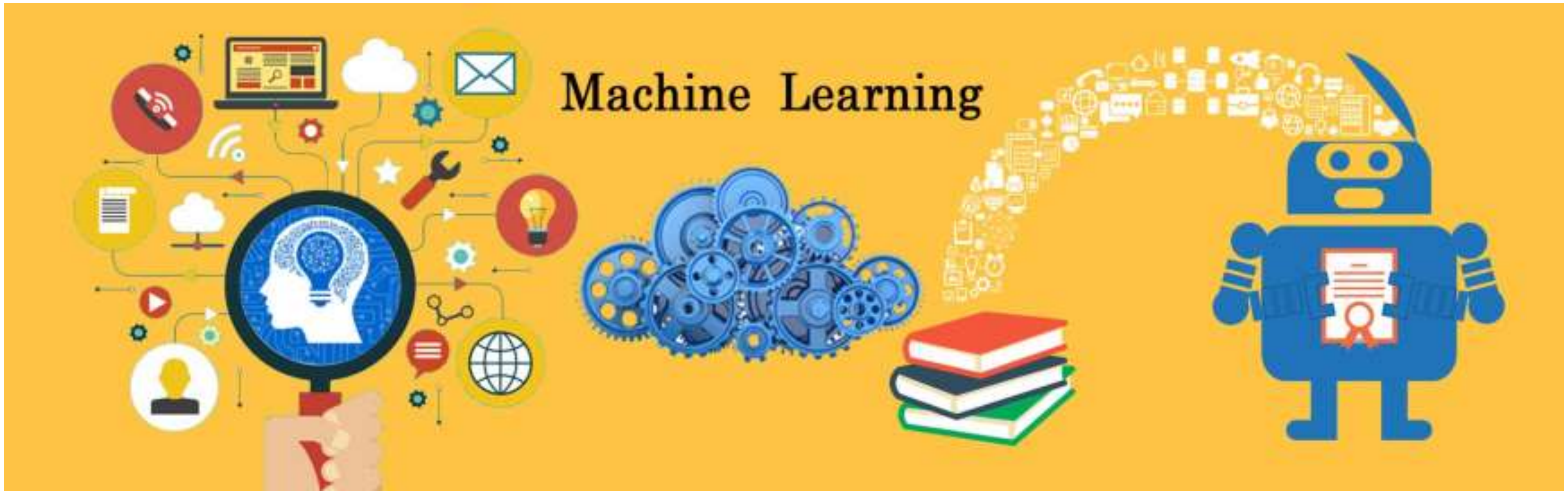
Fake News Detection Using Machine Learning in Python

By
Sri Sesha Sai Pavan Josyula
Enrolment No. 511217002
Department of Mining Engineering
IIEST, Shibpur

What is Machine Learning? What is its importance?



Disadvantages of Fake News

- Biased News
- Misleading Headings
- Propaganda

About this Project

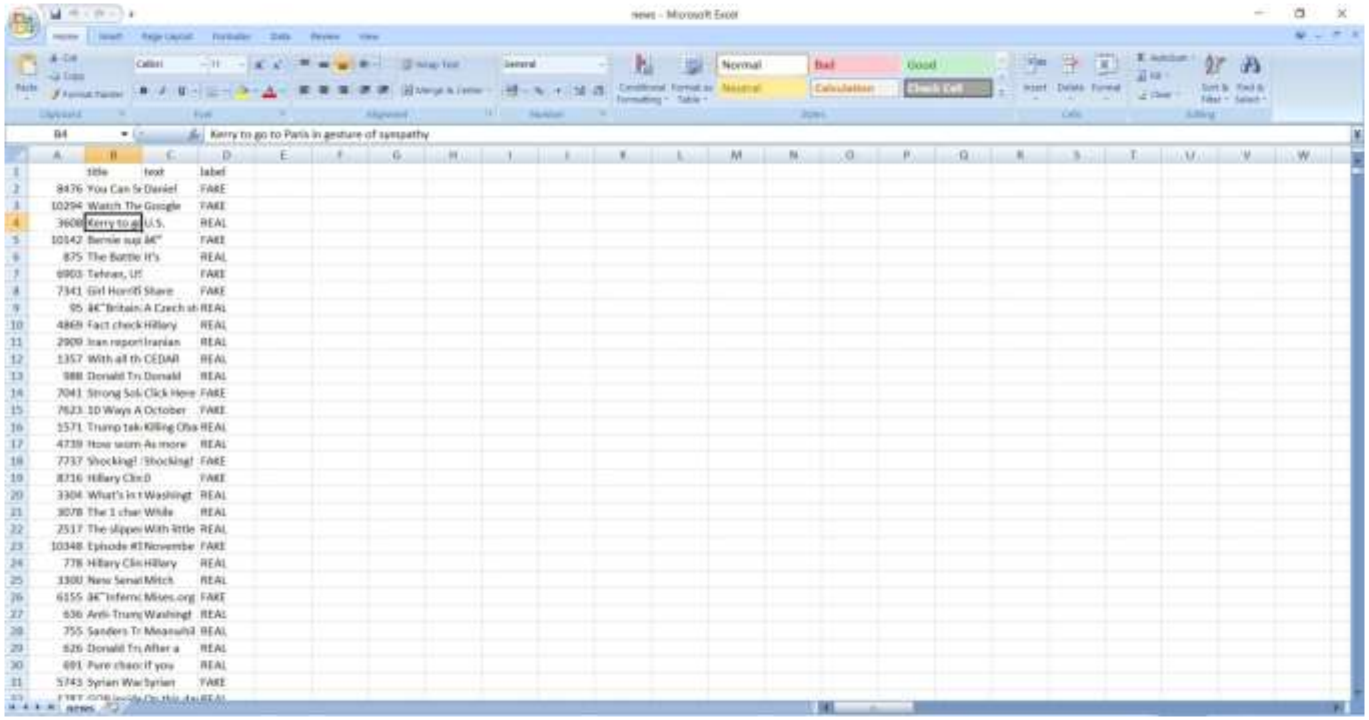
- I did my project in Jupyter Notebooks provided in IBM Skills Network Labs.

- Using *sklearn*, I built a TfidfVectorizer on my dataset. Then, I initialized Passive Aggressive Classifier and fit the model.

- In the end, the accuracy score tells us how well this model works.

The Dataset

- Provided by DataFlair, one of the online Data Science training websites.
- It’s in csv format and its shape is 6635x4



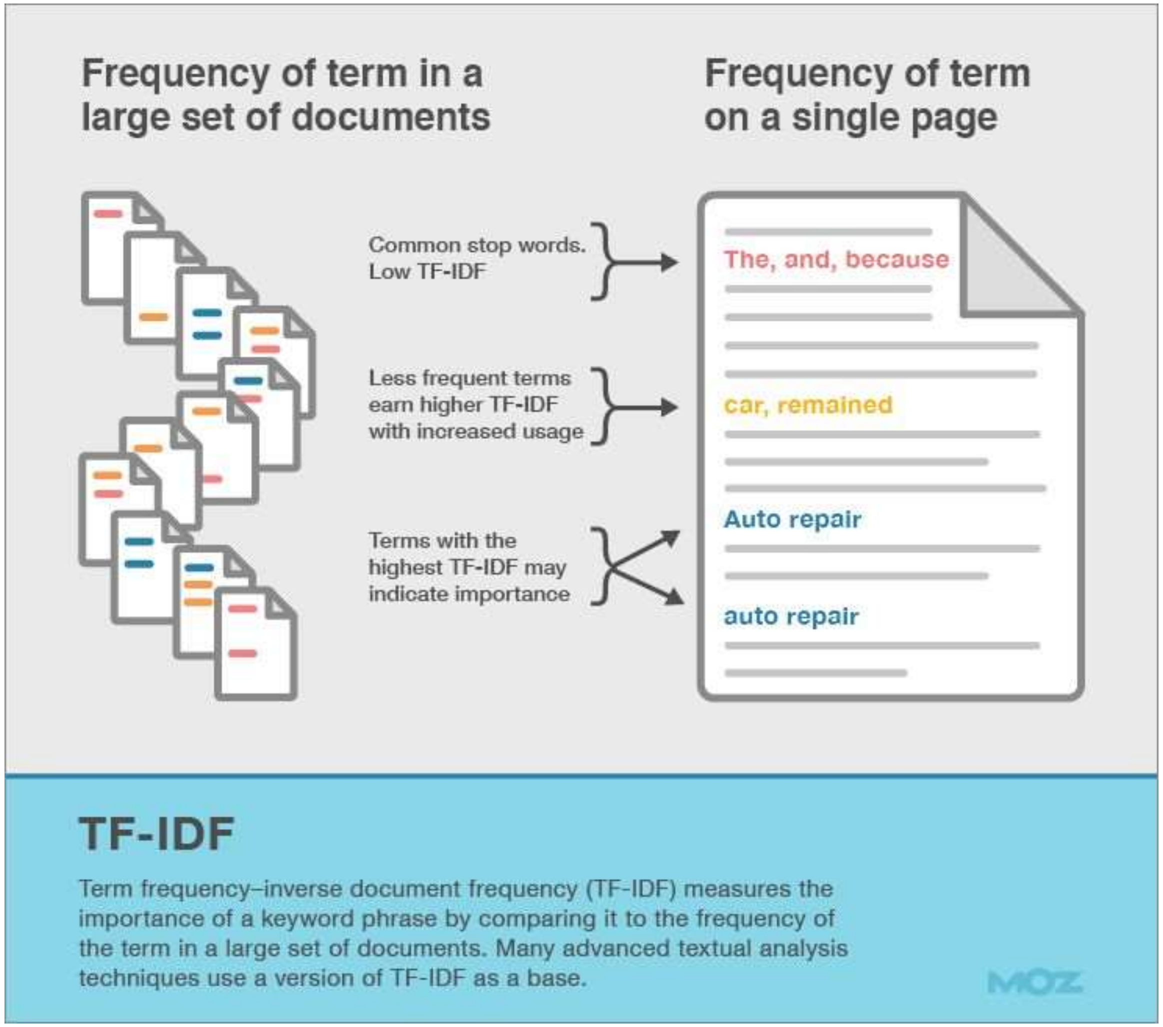
news - Microsoft Excel

B4 Kerry to go to Paris in gesture of sympathy

	A	B	C	D
1		title	text	label
2	8476	You Can Sr	Daniel	FAKE
3	10294	Watch The	Google	FAKE
4	3608	Kerry to go	U.S.	REAL
5	10142	Bernie sup	â€"	FAKE
6	875	The Battle	It's	REAL
7	6903	Tehran, US		FAKE
8	7341	Girl Horrifi	Share	FAKE
9	95	â€˜Britain	A Czech st	REAL
10	4869	Fact check	Hillary	REAL
11	2909	Iran report	Iranian	REAL
12	1357	With all th	CEDAR	REAL
13	988	Donald Tru	Donald	REAL
14	7041	Strong Sol	Click Here	FAKE
15	7623	10 Ways A	October	FAKE
16	1571	Trump tak	Killing Oba	REAL
17	4739	How wom	As more	REAL
18	7737	Shocking!	Shocking!	FAKE
19	8716	Hillary Clin	0	FAKE
20	3304	What's in t	Washingt	REAL
21	3078	The 1 char	While	REAL
22	2517	The slippe	With little	REAL
23	10348	Episode #1	Novembe	FAKE
24	778	Hillary Clin	Hillary	REAL
25	3300	New Senat	Mitch	REAL
26	6155	â€˜Inferno	Mises.org	FAKE
27	636	Anti-Trump	Washingt	REAL
28	755	Sanders Tr	Meanwhil	REAL
29	626	Donald Tru	After a	REAL
30	691	Pure chao	If you	REAL
31	5743	Syrian War	Syrian	FAKE

news

TF-IDF


Frequency of term in a large set of documents
Frequency of term on a single page
Common stop words. Low TF-IDF
Less frequent terms earn higher TF-IDF with increased usage
Terms with the highest TF-IDF may indicate importance
The, and, because
car, remained
Auto repair
auto repair

TF-IDF

Term frequency–inverse document frequency (TF-IDF) measures the importance of a keyword phrase by comparing it to the frequency of the term in a large set of documents. Many advanced textual analysis techniques use a version of TF-IDF as a base.

TF-IDF Explained

So a measure of the relevancy of a word to a document might be:

Term Frequency/Document Frequency

Or Term Frequency*Inverse Document Frequency

That is, take how often word appears in a document, over how often it just appears everywhere. That gives you a measure of how important and unique this word is for this document.

Applications of TF-IDF

- Automated text analysis
- To enhance the performance of search engines
- Useful for machine learning algorithms
- Keyword Extraction
- Deployed by Google, Yahoo, Bing

Use of TF-IDF in this project

- In this project, we make a vector of features that are extracted through TF-IDF.

- The function *TfidfVectorizer* provided in *sklearn* library extracts the TF-IDF features and arranges them in the form of a vector.

Data Sampling

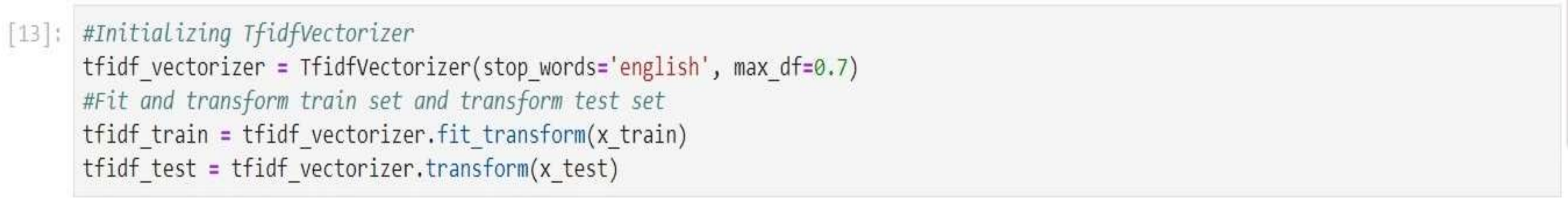
- The dataset has been split into train set and test set.
- 80% of dataset is *train_set* and the rest is *test_set*.

- The model is trained on trained on *train set* and will be tested on *test set*.

- The standard function *train_test_split* facilitates the user to split the dataset

Initializing *TfidfVectorizer*

- The *TfidfVectorizer* is fit and tranformed on the train set and test set.



```
[13]: #Initializing TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
#Fit and transform train set and transform test set
tfidf_train = tfidf_vectorizer.fit_transform(x_train)
tfidf_test = tfidf_vectorizer.transform(x_test)
```

Passive Aggressive Classifier

- It’s an algorithm that remains passive for a correct classification outcome, and turns aggressive in the event of miscalculation.

- It’s an effective classifier and provided in *sklearn* library.

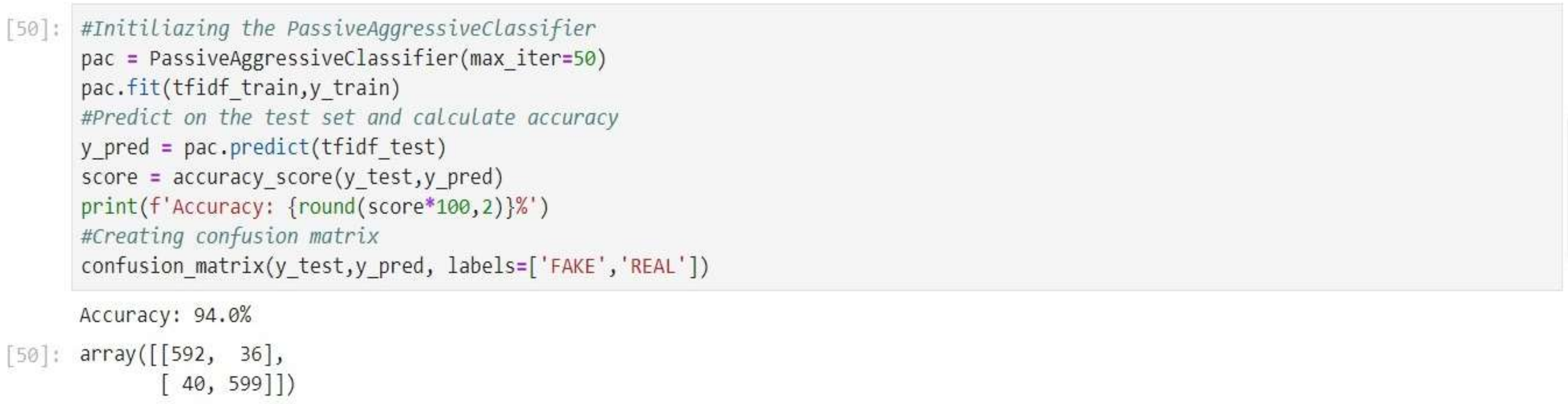
Initializing *PassiveAggressiveClassifier*

- *PassiveAggressiveClassifier* is fit and transformed on the *train set* on which *TfidfVectorizer* is already applied.

- We will predict on test set and will check the accuracy of the model using *accuracy_score* and will give an output of *confusion_matrix*.

Conclusion

- Thus, I've accomplished my machine learning model to classify the news as REAL or FAKE with an accuracy of 93-94%.



```
[50]: #Initiliazing the PassiveAggressiveClassifier
      pac = PassiveAggressiveClassifier(max_iter=50)
      pac.fit(tfidf_train,y_train)
      #Predict on the test set and calculate accuracy
      y_pred = pac.predict(tfidf_test)
      score = accuracy_score(y_test,y_pred)
      print(f'Accuracy: {round(score*100,2)}%')
      #Creating confusion matrix
      confusion_matrix(y_test,y_pred, labels=['FAKE','REAL'])

      Accuracy: 94.0%

[50]: array([[592,  36],
             [ 40, 599]])
```

References

- Alpaydin Ethem (2016), *Machine Learning: The New AI (MIT Press Essential Knowledge Series)*, 1st edition (Kindle Version), The MIT Press

- Muller C. Andreas & Guido Sarah (2016), *Introduction to Machine Learning with Python: A Guide for Data Scientists*, 1st edition (Kindle Version), O'Reilly Media

References (Contd.)

- Matthes Eric (2015), *Python Crash Course: A Hands-on, Project-based Introduction to Programming*, 1st edition (Kindle Version), No Starch Press

- McKinney Wes (2012), Python for Data Analysis: Data Wrangling with Pandas, Numpy, and Ipython, 2nd edition (Kindle Version), O’Reilly Media

Thank you