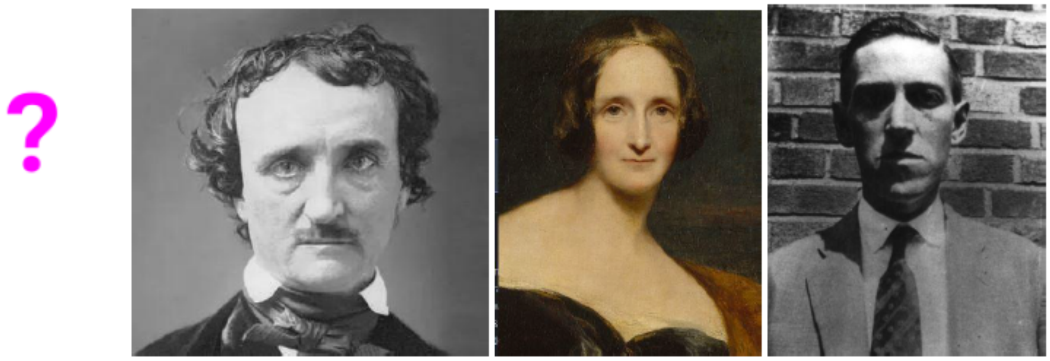
m.5.Assignment -> Spooky authorship identification via Apache Spark

Overview	Use Apache Spark and machine learning to determine sentence authorship labels.
Data	Dark, ominous, and introspective



DETAILS

Dataset Description:

The spooky author identification dataset contains text from works of fiction written by spooky authors of the public domain: Edgar Allan Poe, HP Lovecraft and Mary Shelley. The data was prepared by chunking larger texts into sentences using CoreNLP's MaxEnt sentence tokenizer resulting in an odd non-sentence here and there. Your objective is to accurately identify the author of the sentences in the test set.

- **id** - unique identifier for each sentence
- **text** - sentence written by one of the authors
- **author** - {EAP:Edgar Allan Poe}, {HPL:HP Lovecraft}; {MWS:Mary Wollstonecraft Shelley}

Objective:

- A. Accurately identify the author of the sentences in the test set.
- B. Perform ALL work using Apache Spark.

Dataset:

- Training consists of passages with an author label.

- Test has sentences with no author labels.

Competition Evaluation:

The submissions were evaluated based on multi-class logarithmic loss. The logarithmic loss assesses the uncertainty of the predicted probabilities, penalizing confident incorrect predictions. Lower log loss values indicated better performance.

Approach:

NLP techniques + machine learning algorithms. Feature engineering like bag-of-words, TF-IDF, word embeddings/Word2Vec. Perform algorithmic work with logistic regression, support vector machines, neural networks, and as appropriate.

TASKS

Stage 0: Import Data

1. Create a code notebook called: code_6_of_10_data_mine_<your_name>.ipynb
2. Load data into Spark data objects and explore structure, size, and distribution of information.

Load data into Spark DataFrame

In [1]:

```
# The notebook already created.
# Load data into Spark data objects and explore structure, size and di
from pyspark.sql import SparkSession
import sys
import os, sys

os.environ["PYSPARK_PYTHON"] = sys.executable
os.environ["PYSPARK_DRIVER_PYTHON"] = sys.executable


# Create a SparkSession
spark = SparkSession.builder \
                    .appName("SpookyAuthor") \
                    .master("local[*]") \
                    .config("spark.driver.memory", "12g") \
                    .config("spark.pyspark.python", sys.executable) \
                    .config("spark.pyspark.driver.python", sys.executable
                    .config("spark.python.worker.faulthandler.enabled"
                    .config("spark.sql.execution.pyspark.udf.faulthand
                    .config("spark.driver.allowMultipleContexts", "tru
                    .config("spark.executor.allowMultipleContexts", "t
                    .config("spark.python.worker.reuse", "true") \
                    .getOrCreate()

# Load the train.csv & test.csv dataset into a Spark DataFrame
train_df = spark.read.option("quote", "\"").option("escape", "\"").csv
test_df = spark.read.csv("data/test.csv", header=True, inferSchema=Tru

# Show the first few rows of the train and test DataFrame
print("Train DataFrame:\n")
train_df.show(3)

print("\nTest DataFrame:\n")
test_df.show(3)
```

```
WARNING: Using incubator modules: jdk.incubator.vector
Using Spark's default log4j profile: org/apache/spark/log4j2-defaults.p
roperties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use s
etLogLevel(newLevel).
25/07/28 07:15:33 WARN NativeCodeLoader: Unable to load native-hadoop l
ibrary for your platform... using builtin-java classes where applicable
Train DataFrame:

+-------+--------------------+------+
|     id|                text|author|
+-------+--------------------+------+
|id26305|This process, how...|   EAP|
|id17569|It never once occ...|   HPL|
|id11008|In his left hand ...|   EAP|
+-------+--------------------+------+
only showing top 3 rows

Test DataFrame:

+-------+--------------------+
|     id|                text|
+-------+--------------------+
|id02310|Still, as I urged...|
|id24541|If a fire wanted ...|
|id00134|And when they had...|
+-------+--------------------+
only showing top 3 rows
```

Explore structure, size, and distribution of information

In [2]:

```
# Structure of the train and test DataFrame
print("Train DataFrame Structure:\n")
train_df.printSchema()
```

```python
print("\nTest DataFrame Structure:\n")
test_df.printSchema()

# Size of the train and test DataFrame
print(f"\nTrain DataFrame Size: {train_df.count()}")
print(f"\nTest DataFrame Size: {test_df.count()}")

# Distribution of the train and test DataFrame
print("\nTrain DataFrame Distribution:\n")
train_df.describe().show()

print("\nTest DataFrame Distribution:\n")
test_df.describe().show()
```

```
Train DataFrame Structure:

root
 |-- id: string (nullable = true)
 |-- text: string (nullable = true)
 |-- author: string (nullable = true)


Test DataFrame Structure:

root
 |-- id: string (nullable = true)
 |-- text: string (nullable = true)


Train DataFrame Size: 19579

Test DataFrame Size: 8392

Train DataFrame Distribution:
```

```
+-------+-------+--------------------+------+
|summary|     id|                text|author|
+-------+-------+--------------------+------+
|  count|  19579|               19579| 19579|
|   mean|   NULL|                NULL|  NULL|
| stddev|   NULL|                NULL|  NULL|
|    min|id00001|" Odenheimer, res...|   EAP|
|    max|id27971|you could not hop...|   MWS|
+-------+-------+--------------------+------+


Test DataFrame Distribution:

+-------+-------+--------------------+
|summary|     id|                text|
+-------+-------+--------------------+
|  count|   8392|                8392|
|   mean|   NULL|                NULL|
| stddev|   NULL|                NULL|
|    min|id00008|""" ""Froissart""...|
|    max|id27970|yes, I hear it, a...|
+-------+-------+--------------------+
```

Stage 1: Data Preparation - Exploratory data analysis and text mining pre-processing

3. Perform exploratory data analysis and create visualizations and tables as needed.
4. Text Preprocessing: perform tasks like tokenization and stopwords removal to clean text data.
   - Tokenize - split the text into individual words aka tokens.
   - Remove stop.words - frequently used pronouns and personal references.
     - Top ten include: I, you, he, she, it, we, they, me, him, her
   - Lemmatization - convert words to their root (optional).
     - Lemmatization is a text normalization technique that reduces words to their base or dictionary form (lemma). Use to reduce inflected or derived words to their root form for better analysis and modeling outcomes.

Exploratory Data Analysis

Check the null value in train dataset

In [3]:

```python
from pyspark.sql.functions import when, col, count

# Check the null value in train dataset.
```

```
print(f"Null/Empty Value in train_df:")
train_df.select(
    [
      count(
        when(col(c).isNull(), c)
      ).alias(c)
      for c in train_df.columns
    ]
).show()

test_df.select(
    [
      count(
        when(col(c).isNull(), c)
      ).alias(c)
      for c in test_df.columns
    ]
).show()
```

```
Null/Empty Value in train_df:
+---+----+------+
| id|text|author|
+---+----+------+
|  0|   0|     0|
+---+----+------+

+---+----+
| id|text|
+---+----+
|  0|   0|
+---+----+
```

Show text length statistics

In [4]:

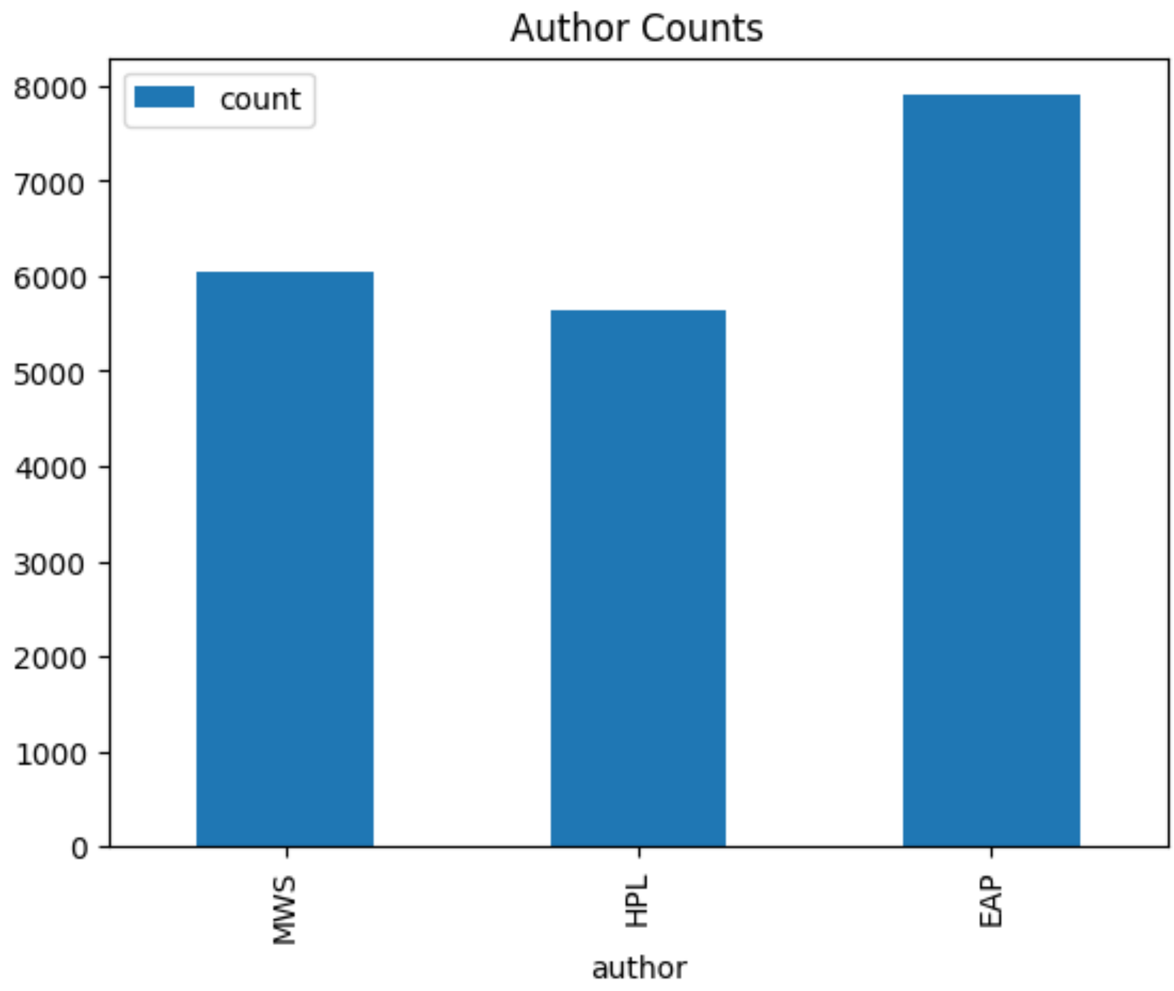
```
# Show text length statistics
print("\nText Length Statistics:")
train_df.select("text").summary().show()
```

```
Text Length Statistics:
+-------+--------------------+
|summary|                text|
+-------+--------------------+
|  count|               19579|
|   mean|                NULL|
| stddev|                NULL|
|    min|" Odenheimer, res...|
|    25%|                NULL|
|    50%|                NULL|
|    75%|                NULL|
|    max|you could not hop...|
+-------+--------------------+
```

Show counts of each author

```
In [5]: import matplotlib.pyplot as plt
        import pandas as pd

        # Show counts of each author
        print("Author Counts:")
        author_counts_df = train_df.select("author").groupBy("author").count()
        author_counts_df.show()
        author_counts_df.toPandas().plot(kind='bar', x='author', y='count', ti
        plt.show()
```

```
Author Counts:
+------+-----+
|author|count|
+------+-----+
|   MWS| 6044|
|   HPL| 5635|
|   EAP| 7900|
+------+-----+
```



Text Preprocessing

Perform tasks like tokenization and stopwords removal to clean text data.

- Tokenize - split the text into individual words aka tokens.
- Remove stop.words - frequently used pronouns and personal references.
    - Top ten include: I, you, he, she, it, we, they, me, him, her
    - *There are couple of ways to remove stop words from the text data. One way is to use the `StopWordsRemover` class from the `pyspark.ml.feature` module.*
- Lemmatization - convert words to their root (optional).
    - Lemmatization is a text normalization technique that reduces words to their base or dictionary form (lemma). Use to reduce inflected or derived words to their root form for better analysis and modeling outcomes.
    - *We have multiple ways to lemmatize the text data. One way is to use the `WordNetLemmatizer` class from the `nltk` library. Lemmatization is very helpful as it helps to reduce words to their base forms, allowing for more accurate analysis and identification of common word forms.*

In [6]:

```
import sys
print("Driver Python:", sys.executable, sys.version)
conf = spark.sparkContext.getConf()
print("spark.pyspark.python:", conf.get("spark.pyspark.python"))
print("spark.pyspark.driver.python:", conf.get("spark.pyspark.driver.p
```

```
Driver Python: /Users/jared/.pyenv/versions/3.10.16/bin/python 3.10.16
(main, Jun  9 2025, 19:05:54) [Clang 17.0.0 (clang-1700.0.13.5)]
spark.pyspark.python: /Users/jared/.pyenv/versions/3.10.16/bin/python
spark.pyspark.driver.python: /Users/jared/.pyenv/versions/3.10.16/bin/p
ython
```

In [7]:

```
# Prepare the NLTK resources
import nltk
from nltk.corpus import stopwords, wordnet
from nltk.stem import WordNetLemmatizer
import warnings
warnings.filterwarnings('ignore')

# Download the NLTK corpus if not already downloaded
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package stopwords to /Users/jared/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /Users/jared/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /Users/jared/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
[nltk_data] Downloading package punkt to /Users/jared/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /Users/jared/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package punkt_tab to /Users/jared/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
```

Out[7]: True

In [8]:

```python
from pyspark.ml.feature import Tokenizer
from pyspark.sql import functions
from pyspark.sql.functions import udf
from pyspark.sql.types import StringType
import string


# Get the list of stopwords
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

def remove_special_character(text : str) -> str:
    return text.translate(str.maketrans('', '', string.punctuation))

def remove_stopwords(text):
    if text:
        return ' '.join([word for word in text.split() if word.lower()
    return ''

def get_wordnet_pos(treebank_tag):
    if treebank_tag.startswith('J'):
        return wordnet.ADJ
    elif treebank_tag.startswith('V'):
        return wordnet.VERB
    elif treebank_tag.startswith('N'):
        return wordnet.NOUN
    elif treebank_tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN  # default to noun

def lemmatize_text(text):
    tokens = nltk.word_tokenize(text)
    pos_tags = nltk.pos_tag(tokens)
    lemmatized = [lemmatizer.lemmatize(word, get_wordnet_pos(pos)) for
    return ' '.join(lemmatized)
```

```python

def preprocess_text(text):
    text = lemmatize_text(text)
    # text = remove_special_character(text)
    # text = remove_stopwords(text)
    return text


train_pd_df = train_df.toPandas()
train_pd_df['preprocessed_text'] = train_pd_df['text'].apply(lambda st
train_df = spark.createDataFrame(train_pd_df)

test_pd_df = test_df.toPandas()
test_pd_df['preprocessed_text'] = test_pd_df['text'].apply(lambda str:
test_df = spark.createDataFrame(test_pd_df)


tokenizer = Tokenizer(inputCol="preprocessed_text", outputCol="tokens"

# Tokenize the text column
transformed_df = tokenizer.transform(train_df)
transformed_test_df = tokenizer.transform(test_df)

# # Show the first few rows of the filtered DataFrame
print("Tokenized Training DataFrame:\n")
transformed_df.select("tokens").show(3, truncate=False)
print("\nTokenized Test DataFrame:\n")
transformed_test_df.select("tokens").show(3, truncate=False)
```

```
Tokenized Training DataFrame:

```

```
+----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-------------------------------------------------------------+
|tokens
|
+----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-------------------------------------------------------------+
|[this, process, ,, however, ,, afford, me, no, mean, of, ascertain, th
e, dimension, of, my, dungeon, ;, a, i, might, make, it, circuit, ,, an
d, return, to, the, point, whence, i, set, out, ,, without, be, aware,
of, the, fact, ;, so, perfectly, uniform, seem, the, wall, .]|
|[it, never, once, occur, to, me, that, the, fumbling, might, be, a, me
re, mistake, .]
|
|[in, his, left, hand, be, a, gold, snuff, box, ,, from, which, ,, a, h
e, caper, down, the, hill, ,, cut, all, manner, of, fantastic, step, ,,
he, take, snuff, incessantly, with, an, air, of, the, great, possible,
self, satisfaction, .]                                       |
+----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-------------------------------------------------------------+
only showing top 3 rows

Tokenized Test DataFrame:

+----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------+
|tokens
|
+----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------+
|[still, ,, a, i, urge, our, leave, ireland, with, such, inquietude, an
d, impatience, ,, my, father, think, it, best, to, yield, .]
|
|[if, a, fire, want, fanning, ,, it, could, readily, be, fan, with, a,
newspaper, ,, and, a, the, government, grow, weak, ,, i, have, no, doub
t, that, leather, and, iron, acquire, durability, in, proportion, ,, fo
r, ,, in, a, very, short, time, ,, there, be, not, a, pair, of, bellow,
in, all, rotterdam, that, ever, stand, in, need, of, a, stitch, or, req
uire, the, assistance, of, a, hammer, .]|
|[and, when, they, have, break, down, the, frail, door, they, find, onl
y, this, :, two, cleanly, pick, human, skeleton, on, the, earthen, floo
r, ,, and, a, number, of, singular, beetle, crawl, in, the, shadowy, co
rner, .]
|
+----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------+
only showing top 3 rows
```

Note:

- Before create a new column for tokenzing, we need to lemmatize the text, then remove special characters and stopwords.
- If we do not lemmatize the text first, the tokenized words will be the base form of the word, which will not be very helpful for analysis.
- If we do not remove special characters and stopwords first, the tokenized

words will be very long and will not be very helpful for analysis.
- If we do lemmatize last, the functionality is not working as expected as they need context to decide what is the root form of the word.

Stage 2: Feature Extraction

5. Perform TF-IDF to quantify word importance `<term.frequency.inverse.doc.frequency>`
6. Normalize is scaling or standardizing the numerical features to a standard range or distribution.
    - In text mining, normalization vectorizes features with methods like TF-IDF, a numerical measurement, to ensure a consistent scale.
    - It handles variations in the magnitude of feature values impacting machine-learning algorithm performance. Normalize the features to ensure a similar scale and prevent features with larger values from dominating the analysis or modeling process.

In [9]:

```
from pyspark.ml.feature import HashingTF, IDF
from pyspark.ml import Pipeline

# Limit the dataset, my environment kept running out of memory. I shou
# transformed_df = transformed_df.limit(500) # Petty TODO: Remove this
# transformed_test_df = transformed_test_df.limit(25) # Petty TODO: Re

# Init HashingTF
##
   # can manipulate 'numFeatures' field in the constructor to potentia
   # once we implement the Machine Learning phase. That constructor ca
   # hashingtf = HashingTF(inputCol="tokens", outputCol="raw_frequenci
   #     - Default value is 262144
##
hashingtf = HashingTF(inputCol="tokens", outputCol="raw_frequencies")

# Init IDF
idf = IDF(inputCol="raw_frequencies", outputCol="features")

# Build pipeline
pipeline = Pipeline(stages=[hashingtf, idf])

# Fit and transform
model = pipeline.fit(transformed_df)
tfidf = model.transform(transformed_df)
model_test = pipeline.fit(transformed_test_df)
tfidf_test = model_test.transform(transformed_test_df)

# Show the first few rows of the tf-idf transformed dataframe
print("TF-IDF normalized features:")
```

```python
tfidf.select("features").show(3, truncate=False)
print("\nTest set also transformed (truncated):")
tfidf_test.select("features").show(3)
```

```
TF-IDF normalized features:
+----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
------------------------------------+
|features
|
+----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
------------------------------------+
|(262144,[8732,19036,20901,27576,30950,34188,38308,45404,46899,50001,88
590,89717,91767,95889,100941,102036,107107,108541,117491,131709,141331,
142239,145207,148880,167503,183339,183938,186312,209518,214862,219087,2
19915,221027,237388,240944,248200,256468,261675],[7.397357266296501,2.0
36730623534044,3.108592115558026,0.8152167139347949,1.425457874683359,
5.839212648249951,4.405800364152991,0.044969586283172255,5.427916619830
993,2.072722591431091,6.550059405909297,3.04493110139891,6.271346003440
277,1.2477762646451427,7.243206586469243,6.271346003440277,0.7931876316
013579,2.0739408656934453,2.6495307799068866,4.298767607302825,1.12195
89976525253,4.007333185232471,0.5407198584929742,4.261863050367351,4.70
6114183510672,3.99061970425873,3.332613173850691,2.550548946358035,4.7
06114183510672,6.448276711599355,1.611653523768082,0.5701795602250346,
7.579678823090456,3.4114644123018993,1.60918258241867,6.70421008573655
5,3.5055369681858743,4.16523621467828])|
|(262144,[15313,17046,27576,30950,39275,45404,48448,50001,95889,107107,
113673,121401,145207,252843,256468],[5.382454245754237,5.7713900519111
9,0.8152167139347949,1.425457874683359,7.802822374404665,0.044969586283
172255,1.3425267602333695,2.072722591431091,0.31194406616128567,0.79318
76316013579,3.6075018948425623,5.0619823504794645,0.5407198584929742,3.
8636707015882665,3.5055369681858743])
|
|(262144,[4106,22370,25217,28338,33860,42404,45404,48648,49120,55639,62
790,75292,95889,98424,101169,102006,107107,112947,126466,141331,143202,
145207,170414,199176,219087,224255,227860,232367,239343,245523,250855,2
61845,261870],[4.764270103667746,5.349664422931245,3.7932190406376556,
7.802822374404665,1.8967795593506787,2.3475012590469637,0.0449695862831
72255,5.146065467690006,1.9229879559681051,3.3885100762328153,4.6673281
58475516,6.663388091216301,0.6238881323225713,4.376932380152138,2.04155
7464335102,5.349664422931245,1.5863752632027157,15.370078677496563,1.66
02474788823054,1.1219589976525253,2.458098635042473,0.5407198584929742,
5.349664422931245,6.326915854595088,1.0744356825120547,3.58066092020921
54,4.912450616508501,3.7641667180431533,8.783651627416392,4.56414392224
0285,1.014132202581468,5.6926091740580755,3.434958053543288])
|
+----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
------------------------------------+
only showing top 3 rows

Test set also transformed (truncated):
```

```
25/07/28 07:16:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.0 MiB
25/07/28 07:16:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.0 MiB
```

```
+--------------------+
|            features|
+--------------------+
|(262144,[4629,190...|
|(262144,[1623,490...|
|(262144,[24980,25...|
+--------------------+
only showing top 3 rows
```

Stage 3: Machine Learning

7. Perform train\test split.
8. Perform algorithmic analysis to assess and predict test labels.

- a. Use as many algorithms as you need to get a good answer.
- b. Supervised: logistic regression, random forest, support vector machines, etc.
- c. Unsupervised: K-means, dimensionality reduction, PCA, etc.

In [10]:

```
from pyspark.ml.classification import DecisionTreeClassifier, Logistic
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml.feature import StringIndexer
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder, CrossV

# Transform authors into numeric (0-2)
#tfidf.show(3)
#tfidf_test.show(3)
label_indexer = StringIndexer(inputCol="author", outputCol="label")
labeled_df = label_indexer.fit(tfidf).transform(tfidf)
#labeled_df.show(3)

# Make a test set so we can have an idea of how the model does
train_data, val_data = labeled_df.randomSplit([0.8, 0.2], seed=42)

dt = DecisionTreeClassifier()
lr = LogisticRegression()
rf = RandomForestClassifier()
nb = NaiveBayes()

accuracy_eval = MulticlassClassificationEvaluator(metricName='accuracy
f1_eval = MulticlassClassificationEvaluator(metricName='f1')
grid = ParamGridBuilder().addGrid(nb.smoothing, [0.5, 1, 1.5, 2]).buil

model = dt.fit(train_data)
dt_prediction = model.transform(val_data)
dt_accuracy_score = accuracy_eval.evaluate(dt_prediction)
dt_f1_score = f1_eval.evaluate(dt_prediction)
print('Decision Tree Accuracy: ', dt_accuracy_score, '\nF1: ', dt_f1_s
dtImportances = model.featureImportances

model = lr.fit(train_data)
lr_prediction = model.transform(val_data)
lr_accuracy_score = accuracy_eval.evaluate(lr_prediction)
lr_f1_score = f1_eval.evaluate(lr_prediction)
print('Logistic Regression Accuracy: ', lr_accuracy_score, '\nF1: ', l
lrImportances = model.coefficientMatrix

model = rf.fit(train_data)
rf_prediction = model.transform(val_data)
rf_accuracy_score = accuracy_eval.evaluate(rf_prediction)
rf_f1_score = f1_eval.evaluate(rf_prediction)
print('Random Forest Accuracy: ', rf_accuracy_score, '\nF1: ', rf_f1_s
rfImportances = model.featureImportances

model = nb.fit(train_data)
```

```
nb_prediction = model.transform(val_data)
nb_accuracy_score = accuracy_eval.evaluate(nb_prediction)
nb_f1_score = f1_eval.evaluate(nb_prediction)
print('Naive Bayes Accuracy: ', nb_accuracy_score, '\nF1: ', nb_f1_sco
nbImportances = model.theta

cv = CrossValidator(estimator=nb, estimatorParamMaps=grid, evaluator=a
model = cv.fit(labeled_df)
cv_prediction = model.transform(labeled_df)
cv_accuracy_score = accuracy_eval.evaluate(cv_prediction)
cv_f1_score = f1_eval.evaluate(cv_prediction)
print('Cross Validator NB Accuracy:', cv_accuracy_score, '\nF1: ', cv_
best_model = model.bestModel
smoothing = best_model.getSmoothing()
cvImportances = f'Best smoothing parameter found by Cross Validator: {
```

```
25/07/28 07:16:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:16:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:16:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 6.6 MiB
25/07/28 07:16:45 WARN DAGScheduler: Broadcasting large task binary wit
h size 1033.6 KiB
25/07/28 07:16:46 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.7 MiB
25/07/28 07:16:50 WARN MemoryStore: Not enough space to cache rdd_163_5
in memory! (computed 630.3 MiB so far)
25/07/28 07:16:50 WARN BlockManager: Persisting block rdd_163_5 to disk
instead.
25/07/28 07:16:50 WARN MemoryStore: Not enough space to cache rdd_163_0
in memory! (computed 630.3 MiB so far)
25/07/28 07:16:50 WARN BlockManager: Persisting block rdd_163_0 to disk
instead.
25/07/28 07:16:51 WARN MemoryStore: Not enough space to cache rdd_163_7
in memory! (computed 630.3 MiB so far)
25/07/28 07:16:51 WARN BlockManager: Persisting block rdd_163_7 to disk
instead.
25/07/28 07:16:51 WARN MemoryStore: Not enough space to cache rdd_163_1
in memory! (computed 951.0 MiB so far)
25/07/28 07:16:51 WARN BlockManager: Persisting block rdd_163_1 to disk
instead.
25/07/28 07:16:51 WARN MemoryStore: Not enough space to cache rdd_163_4
in memory! (computed 951.0 MiB so far)
25/07/28 07:16:51 WARN BlockManager: Persisting block rdd_163_4 to disk
instead.
25/07/28 07:16:51 WARN MemoryStore: Not enough space to cache rdd_163_6
in memory! (computed 951.0 MiB so far)
25/07/28 07:16:51 WARN BlockManager: Persisting block rdd_163_6 to disk
instead.
25/07/28 07:16:51 WARN MemoryStore: Not enough space to cache rdd_163_2
in memory! (computed 951.0 MiB so far)
25/07/28 07:16:51 WARN BlockManager: Persisting block rdd_163_2 to disk
```

```
instead.
25/07/28 07:16:51 WARN MemoryStore: Not enough space to cache rdd_163_3
in memory! (computed 951.0 MiB so far)
25/07/28 07:16:51 WARN BlockManager: Persisting block rdd_163_3 to disk
instead.
25/07/28 07:17:00 WARN MemoryStore: Not enough space to cache rdd_163_7
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:01 WARN MemoryStore: Not enough space to cache rdd_163_2
in memory! (computed 1435.3 MiB so far)
25/07/28 07:17:01 WARN MemoryStore: Not enough space to cache rdd_163_0
in memory! (computed 1435.3 MiB so far)
25/07/28 07:17:01 WARN MemoryStore: Not enough space to cache rdd_163_4
in memory! (computed 1435.3 MiB so far)
25/07/28 07:17:01 WARN MemoryStore: Not enough space to cache rdd_163_6
in memory! (computed 1435.3 MiB so far)
25/07/28 07:17:02 WARN MemoryStore: Not enough space to cache rdd_163_1
in memory! (computed 65.0 MiB so far)
25/07/28 07:17:02 WARN MemoryStore: Not enough space to cache rdd_163_5
in memory! (computed 419.0 MiB so far)
25/07/28 07:17:03 WARN MemoryStore: Not enough space to cache rdd_163_3
in memory! (computed 113.0 MiB so far)
25/07/28 07:17:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.7 MiB
25/07/28 07:17:13 WARN MemoryStore: Not enough space to cache rdd_163_0
in memory! (computed 630.3 MiB so far)
25/07/28 07:17:13 WARN MemoryStore: Not enough space to cache rdd_163_2
in memory! (computed 630.3 MiB so far)
25/07/28 07:17:13 WARN MemoryStore: Not enough space to cache rdd_163_3
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:13 WARN MemoryStore: Not enough space to cache rdd_163_6
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:13 WARN MemoryStore: Not enough space to cache rdd_163_4
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:14 WARN MemoryStore: Not enough space to cache rdd_163_1
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:14 WARN MemoryStore: Not enough space to cache rdd_163_5
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:14 WARN MemoryStore: Not enough space to cache rdd_163_7
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:20 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.7 MiB
25/07/28 07:17:22 WARN MemoryStore: Not enough space to cache rdd_163_5
in memory! (computed 630.3 MiB so far)
25/07/28 07:17:22 WARN MemoryStore: Not enough space to cache rdd_163_2
in memory! (computed 630.3 MiB so far)
25/07/28 07:17:22 WARN MemoryStore: Not enough space to cache rdd_163_3
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:22 WARN MemoryStore: Not enough space to cache rdd_163_4
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:22 WARN MemoryStore: Not enough space to cache rdd_163_6
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:23 WARN MemoryStore: Not enough space to cache rdd_163_7
```

```
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:23 WARN MemoryStore: Not enough space to cache rdd_163_1
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:23 WARN MemoryStore: Not enough space to cache rdd_163_0
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:30 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.7 MiB
25/07/28 07:17:34 WARN MemoryStore: Not enough space to cache rdd_163_4
in memory! (computed 630.3 MiB so far)
25/07/28 07:17:34 WARN MemoryStore: Not enough space to cache rdd_163_0
in memory! (computed 630.3 MiB so far)
25/07/28 07:17:35 WARN MemoryStore: Not enough space to cache rdd_163_7
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:38 WARN MemoryStore: Not enough space to cache rdd_163_3
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:38 WARN MemoryStore: Not enough space to cache rdd_163_6
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:38 WARN MemoryStore: Not enough space to cache rdd_163_5
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:38 WARN MemoryStore: Not enough space to cache rdd_163_1
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:38 WARN MemoryStore: Not enough space to cache rdd_163_2
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:46 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.7 MiB
25/07/28 07:17:49 WARN MemoryStore: Not enough space to cache rdd_163_6
in memory! (computed 630.3 MiB so far)
25/07/28 07:17:49 WARN MemoryStore: Not enough space to cache rdd_163_3
in memory! (computed 630.3 MiB so far)
25/07/28 07:17:50 WARN MemoryStore: Not enough space to cache rdd_163_7
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:50 WARN MemoryStore: Not enough space to cache rdd_163_2
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:50 WARN MemoryStore: Not enough space to cache rdd_163_4
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:50 WARN MemoryStore: Not enough space to cache rdd_163_0
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:51 WARN MemoryStore: Not enough space to cache rdd_163_5
in memory! (computed 951.0 MiB so far)
25/07/28 07:17:51 WARN MemoryStore: Not enough space to cache rdd_163_1
in memory! (computed 951.0 MiB so far)
25/07/28 07:18:00 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:01 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
```

```
Decision Tree Accuracy:  0.510660700184259
F1:  0.49989926710253074
```

```
25/07/28 07:18:01 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:02 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:02 WARN InstanceBuilder: Failed to load implementation f
```

```
rom:dev.ludovic.netlib.blas.JNIBLAS
25/07/28 07:18:02 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:06 WARN DAGScheduler: Broadcasting large task binary wit
```

```
h size 4.1 MiB
25/07/28 07:18:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:09 WARN DAGScheduler: Broadcasting large task binary wit
```

```
h size 4.1 MiB
25/07/28 07:18:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
```

```
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.7 MiB
25/07/28 07:18:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.7 MiB
```

```
Logistic Regression Accuracy:  0.7457225585680443
F1:  0.7457210793333771
```

```
25/07/28 07:18:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:18:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 6.6 MiB
25/07/28 07:18:53 WARN DAGScheduler: Broadcasting large task binary wit
h size 1033.6 KiB
25/07/28 07:18:54 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.8 MiB
25/07/28 07:18:58 WARN MemoryStore: Not enough space to cache rdd_430_1
in memory! (computed 630.3 MiB so far)
25/07/28 07:18:58 WARN BlockManager: Persisting block rdd_430_1 to disk
instead.
25/07/28 07:18:58 WARN MemoryStore: Not enough space to cache rdd_430_6
in memory! (computed 630.3 MiB so far)
25/07/28 07:18:58 WARN BlockManager: Persisting block rdd_430_6 to disk
```

```
instead.
25/07/28 07:18:58 WARN MemoryStore: Not enough space to cache rdd_430_0
in memory! (computed 630.3 MiB so far)
25/07/28 07:18:58 WARN BlockManager: Persisting block rdd_430_0 to disk
instead.
25/07/28 07:18:58 WARN MemoryStore: Not enough space to cache rdd_430_7
in memory! (computed 951.1 MiB so far)
25/07/28 07:18:58 WARN BlockManager: Persisting block rdd_430_7 to disk
instead.
25/07/28 07:18:58 WARN MemoryStore: Not enough space to cache rdd_430_4
in memory! (computed 951.1 MiB so far)
25/07/28 07:18:58 WARN BlockManager: Persisting block rdd_430_4 to disk
instead.
25/07/28 07:18:58 WARN MemoryStore: Not enough space to cache rdd_430_5
in memory! (computed 951.1 MiB so far)
25/07/28 07:18:58 WARN BlockManager: Persisting block rdd_430_5 to disk
instead.
25/07/28 07:18:59 WARN MemoryStore: Not enough space to cache rdd_430_2
in memory! (computed 951.1 MiB so far)
25/07/28 07:18:59 WARN BlockManager: Persisting block rdd_430_2 to disk
instead.
25/07/28 07:18:59 WARN MemoryStore: Not enough space to cache rdd_430_3
in memory! (computed 951.1 MiB so far)
25/07/28 07:18:59 WARN BlockManager: Persisting block rdd_430_3 to disk
instead.
25/07/28 07:19:07 WARN MemoryStore: Not enough space to cache rdd_430_7
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:07 WARN MemoryStore: Not enough space to cache rdd_430_4
in memory! (computed 1435.4 MiB so far)
25/07/28 07:19:07 WARN MemoryStore: Not enough space to cache rdd_430_6
in memory! (computed 1435.4 MiB so far)
25/07/28 07:19:07 WARN MemoryStore: Not enough space to cache rdd_430_0
in memory! (computed 1435.4 MiB so far)
25/07/28 07:19:07 WARN MemoryStore: Not enough space to cache rdd_430_2
in memory! (computed 1435.4 MiB so far)
25/07/28 07:19:15 WARN MemoryStore: Not enough space to cache rdd_430_1
in memory! (computed 2.1 GiB so far)
25/07/28 07:19:15 WARN MemoryStore: Not enough space to cache rdd_430_5
in memory! (computed 2.1 GiB so far)
25/07/28 07:19:15 WARN MemoryStore: Not enough space to cache rdd_430_3
in memory! (computed 2.1 GiB so far)
25/07/28 07:19:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.8 MiB
25/07/28 07:19:21 WARN MemoryStore: Not enough space to cache rdd_430_7
in memory! (computed 630.3 MiB so far)
25/07/28 07:19:21 WARN MemoryStore: Not enough space to cache rdd_430_3
in memory! (computed 630.3 MiB so far)
25/07/28 07:19:22 WARN MemoryStore: Not enough space to cache rdd_430_0
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:22 WARN MemoryStore: Not enough space to cache rdd_430_2
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:22 WARN MemoryStore: Not enough space to cache rdd_430_6
```

```
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:22 WARN MemoryStore: Not enough space to cache rdd_430_4
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:22 WARN MemoryStore: Not enough space to cache rdd_430_5
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:22 WARN MemoryStore: Not enough space to cache rdd_430_1
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:27 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.8 MiB
25/07/28 07:19:30 WARN MemoryStore: Not enough space to cache rdd_430_1
in memory! (computed 630.3 MiB so far)
25/07/28 07:19:30 WARN MemoryStore: Not enough space to cache rdd_430_6
in memory! (computed 630.3 MiB so far)
25/07/28 07:19:31 WARN MemoryStore: Not enough space to cache rdd_430_3
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:31 WARN MemoryStore: Not enough space to cache rdd_430_5
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:31 WARN MemoryStore: Not enough space to cache rdd_430_2
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:31 WARN MemoryStore: Not enough space to cache rdd_430_0
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:31 WARN MemoryStore: Not enough space to cache rdd_430_7
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:31 WARN MemoryStore: Not enough space to cache rdd_430_4
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:36 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.9 MiB
25/07/28 07:19:38 WARN MemoryStore: Not enough space to cache rdd_430_0
in memory! (computed 630.3 MiB so far)
25/07/28 07:19:38 WARN MemoryStore: Not enough space to cache rdd_430_4
in memory! (computed 630.3 MiB so far)
25/07/28 07:19:38 WARN MemoryStore: Not enough space to cache rdd_430_2
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:38 WARN MemoryStore: Not enough space to cache rdd_430_6
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:38 WARN MemoryStore: Not enough space to cache rdd_430_3
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:38 WARN MemoryStore: Not enough space to cache rdd_430_1
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:38 WARN MemoryStore: Not enough space to cache rdd_430_7
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:38 WARN MemoryStore: Not enough space to cache rdd_430_5
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:43 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.9 MiB
25/07/28 07:19:46 WARN MemoryStore: Not enough space to cache rdd_430_0
in memory! (computed 630.3 MiB so far)
25/07/28 07:19:46 WARN MemoryStore: Not enough space to cache rdd_430_7
in memory! (computed 630.3 MiB so far)
25/07/28 07:19:46 WARN MemoryStore: Not enough space to cache rdd_430_3
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:46 WARN MemoryStore: Not enough space to cache rdd_430_6
```

```
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:46 WARN MemoryStore: Not enough space to cache rdd_430_2
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:46 WARN MemoryStore: Not enough space to cache rdd_430_4
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:46 WARN MemoryStore: Not enough space to cache rdd_430_1
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:46 WARN MemoryStore: Not enough space to cache rdd_430_5
in memory! (computed 951.1 MiB so far)
25/07/28 07:19:52 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.2 MiB
25/07/28 07:19:52 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.2 MiB
```

```
Random Forest Accuracy:  0.41932087391418793
F1:  0.2634327339985048
```

```
25/07/28 07:19:53 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:19:54 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:19:54 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:19:55 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
```

```
Naive Bayes Accuracy:  0.8370623848381153
F1:  0.8371148668772619
```

```
25/07/28 07:19:56 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:19:56 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:19:56 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:19:56 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:19:57 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:19:57 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:19:58 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:19:58 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:19:59 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:19:59 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:20:00 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:00 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:00 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:00 WARN DAGScheduler: Broadcasting large task binary wit
```

```
h size 4.1 MiB
25/07/28 07:20:01 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:20:01 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:20:01 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:01 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:02 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:20:02 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:20:02 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:02 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:02 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:02 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:03 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:20:03 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:20:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:20:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:20:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:20:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
Cross Validator NB Accuracy: 0.9246131058787477
F1:  0.9246895861427589
```

Stage 4: Evaluation & Visualization

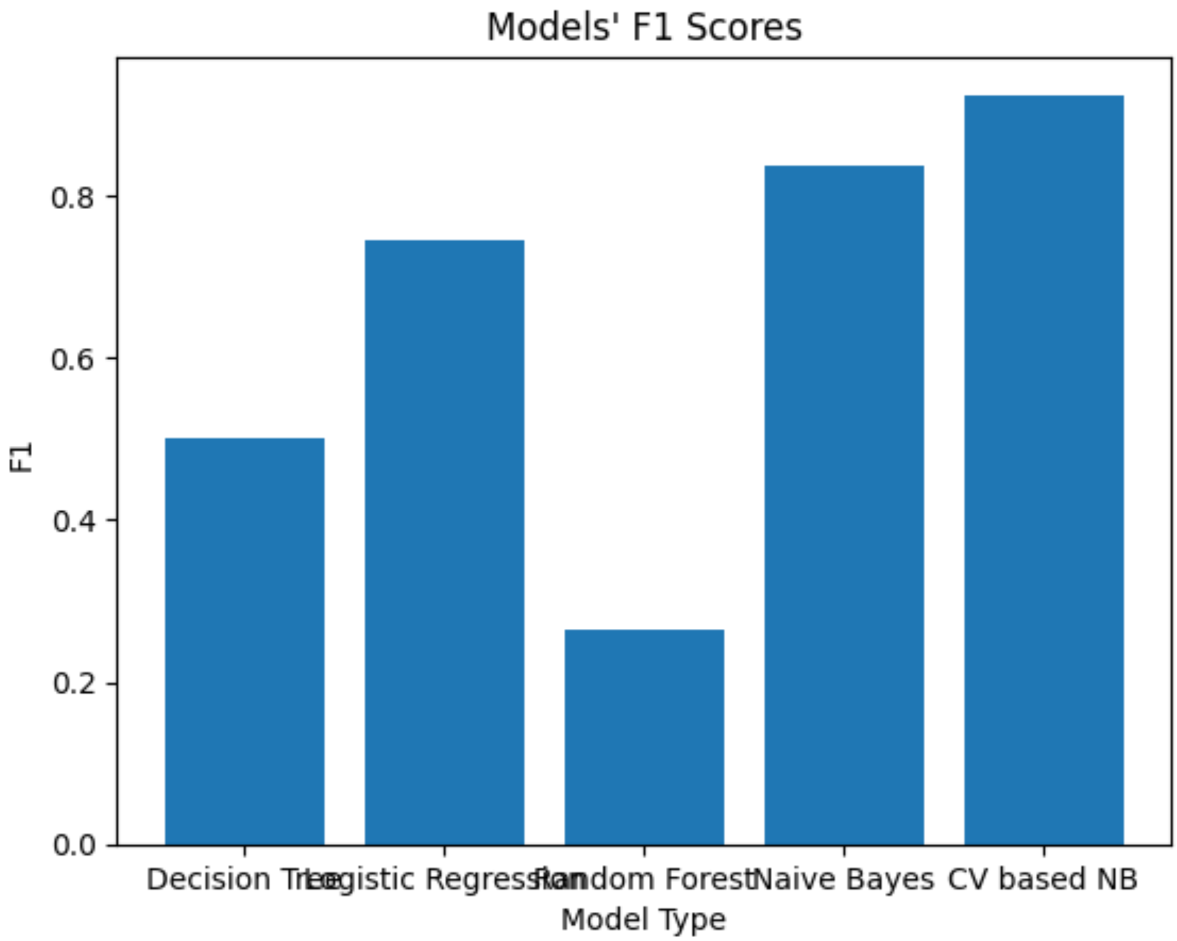
9. Choose a metric strategy to assess algorithmic performance like accuracy, precision, recall, or F1 score.
10. Visualize confusion matrix, correlations, and similar.
11. Identify important features contributing to classification.

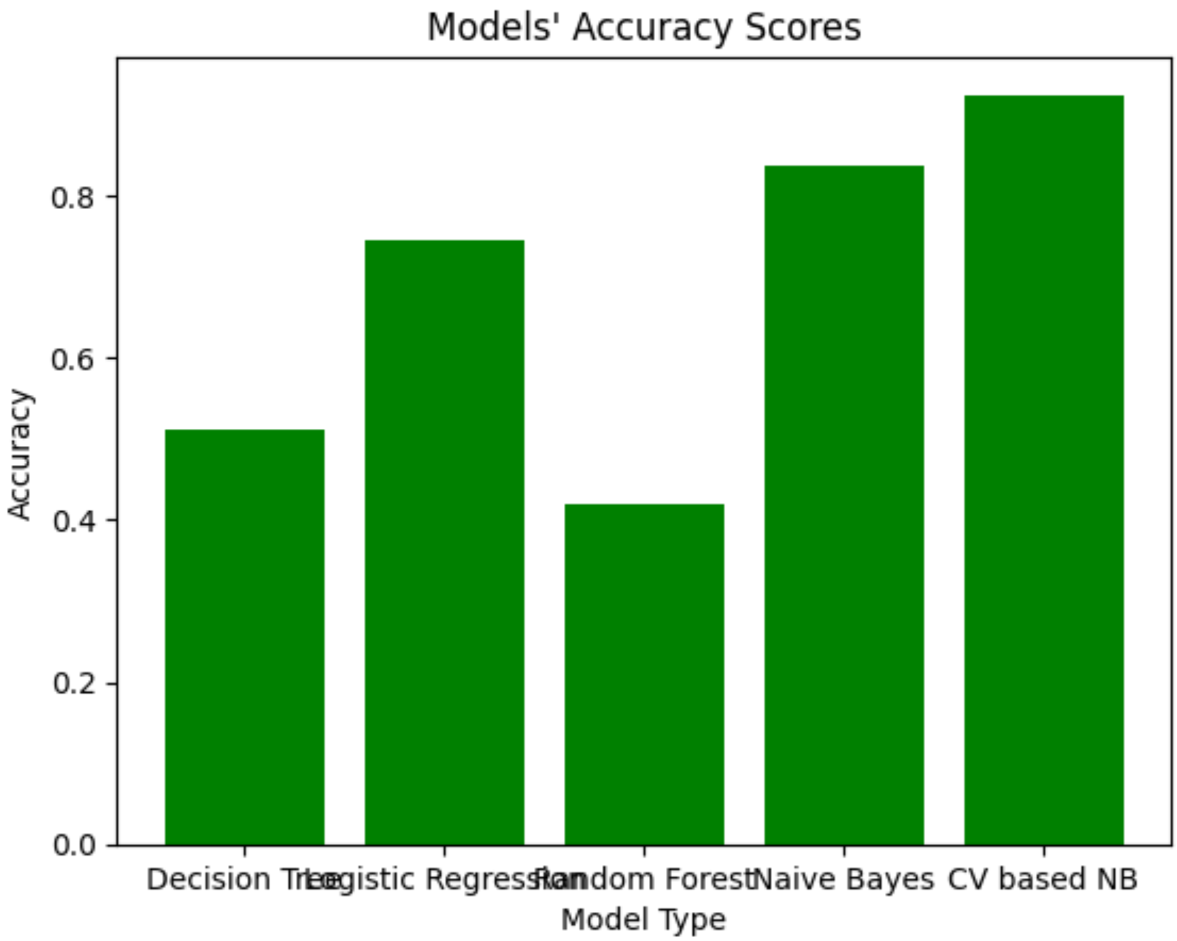
12. Write a 2-3 sentence minimum of findings, learnings, and what you would do next.

In [11]:

```
#Bar chart representing F1 values for each model:
plt.bar(["Decision Tree" , "Logistic Regression" , "Random Forest" , "
plt.xlabel("Model Type")
plt.ylabel("F1")
plt.title("Models' F1 Scores")
plt.show()

#Bar chart representing accuracy values for each model:
plt.bar(["Decision Tree" , "Logistic Regression" , "Random Forest" , "
plt.xlabel("Model Type")
plt.ylabel("Accuracy")
plt.title("Models' Accuracy Scores")
plt.show()

import numpy as np
#utility function to print a matrix from a numpy array
#df is a pandas dataframe
def showAsMatrix(df , title):
    matrix = plt.matshow(df)
    asArray = confusionMatrix.to_numpy()
    for (i, j), value in np.ndenumerate(asArray):
        plt.text(j , i , str(value) , ha = 'center' , va = 'center' ,
    plt.title(title)
    plt.show()

#confusion Matrices:
def printConfusionMatrix(predictionType , prefixString):
    confusionDF = predictionType.groupBy("label", "prediction").count(
    confusionDFPandas = confusionDF.toPandas()
    #StringIndexer docs: "By default, this is ordered by label frequen
    confusionMatrix = confusionDFPandas.pivot_table(index='label', col
    #We change the x and y axes to author names. We use StringIndexer
    #based on its frequency. The most occuring string is assigned 0, n
    namesMap = {0.0 : "EAP" , 1.0 : "MWS" , 2.0 : "HPL"}
    confusionMatrix = confusionMatrix.rename(index = namesMap , column
    print(f"{prefixString}:")
    print(confusionMatrix)
    #showAsMatrix(confusionMatrix , prefixString)

printConfusionMatrix(dt_prediction , "Decision Tree Confusion Matrix")
printConfusionMatrix(lr_prediction , "Logistic Regression Confusion Ma
printConfusionMatrix(rf_prediction , "Random Forest Confusion Matrix")
printConfusionMatrix(nb_prediction , "Naive Bayes Confusion Matrix")
printConfusionMatrix(cv_prediction , "Cross Validator based NB Confusi

#important Features:

#This function prints stats for the given importance data structure. T
```

```
#ambiguous data structure.
#We get the maximum value of the sparse vector (representing the most
def getImportantFeaturesSV(sparseVector , sparseVectorName):
    spMax = max(sparseVector.values)
    spMean = sum(sparseVector.values) / sparseVector.size
    print(f"{sparseVectorName} max value: {spMax}, mean: {spMean}, dif

def getImportantFeaturesM(matrix , matrixName):
    print(f"{matrixName}:")
    asArray = matrix.toArray()
    classes = asArray.shape[0]
    for i in range(classes):
        currentClass = asArray[i]
        classMin = currentClass.min()
        classMean = currentClass.mean()
        difference = classMean - classMin
        print(f"class {i} (one of the authors): Least: {classMin}, Mea

print("We use HashingTF in our computation, so we cannot identify the
print("We can get descriptive statistics about the hashed features and

getImportantFeaturesSV(dtImportances , "Decision Tree")
getImportantFeaturesM(lrImportances , "Logistic Regression Coefficient
getImportantFeaturesSV(rfImportances , "Random Forest")
getImportantFeaturesM(nbImportances , "Naive Bayes Theta Matrix")
print(cvImportances)
```


Models' F1 Scores
F1
0.0
0.2
0.4
0.6
0.8
Decision Tree
Logistic Regression
Random Forest
Naive Bayes
CV based NB
Model Type



```
25/07/28 07:20:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.7 MiB
```

```
Decision Tree Confusion Matrix:
prediction  EAP  MWS  HPL
label
EAP         989  148  406
MWS         430  350  410
HPL         379   86  601
```

```
25/07/28 07:20:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.7 MiB
25/07/28 07:20:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.2 MiB
```

```
Logistic Regression Confusion Matrix:
prediction   EAP  MWS  HPL
label
EAP         1160  227  156
MWS          191  902   97
HPL          181  114  771
```

```
25/07/28 07:20:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/28 07:20:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
```

```
Random Forest Confusion Matrix:
prediction   EAP  MWS  HPL
label
EAP         1540    3    0
MWS         1140   50    0
HPL         1063    0    3
```

```
25/07/28 07:20:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/28 07:20:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
```

```
Naive Bayes Confusion Matrix:
prediction   EAP   MWS  HPL
label
EAP         1286   167   90
MWS          122  1011   57
HPL          123    60  883
Cross Validator based NB Confusion Matrix:
prediction   EAP   MWS   HPL
label
EAP         7294   435   171
MWS          310  5647    87
HPL          319   154  5162
We use HashingTF in our computation, so we cannot identify the which wo
rd corresponds to an important feature.
We can get descriptive statistics about the hashed features and determi
ne how significant certain indices are against others in the Sparse Vec
tor, coefficient matrix, or theta.
Decision Tree max value: 0.2194843888795673, mean: 3.814697265625e-06,
difference: 0.21948057418230169.
Logistic Regression Coefficient Matrix:
class 0 (one of the authors): Least: -229.0579990671621, Mean: -0.01444
3138205124452, Difference: 229.04355592895695.
class 1 (one of the authors): Least: -202.9340306568158, Mean: -0.00854
8286845508915, Difference: 202.9254823699703.
class 2 (one of the authors): Least: -30.27769313341363, Mean: 0.022991
425050633358, Difference: 30.300684558464265.
Random Forest max value: 0.0637793205690683, mean: 3.8146972656249975e-
06, difference: 0.06377550587180268.
Naive Bayes Theta Matrix:
class 0 (one of the authors): Least: -13.646656365374696, Mean: -13.513
538131310824, Difference: 0.1331182340638719.
class 1 (one of the authors): Least: -13.504825639938831, Mean: -13.405
679383623628, Difference: 0.09914625631520302.
class 2 (one of the authors): Least: -13.512343276683792, Mean: -13.388
08775618743, Difference: 0.12425552049636224.
Best smoothing parameter found by Cross Validator: 2.0
```

```
25/07/28 07:20:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
```

CONCLUSIONS

In this assignment, we explored the improvement for the accuracy by changing the text preprocessing process and hyperparameter tuning.

- With old process, our model achieved Naive Bayes Accuracy is `0.8239 (82.4%)`. We applied 3 preprocessing techniques orderedly:

    1. Lemmatization
    2. Special character removal
    3. Stopwords removal

    For quick & dirty note, we noted the stopwords and special character definitely help to get the important words which was used for later process.

- For new process, we tried to change the text preprocessing process and hyperparameter tuning to get the better result. We applied only Lemmatization for preprocessing. The result is that we get without Cross Validator is `0.8371 (83.7%)`. It is improved `1.3%` compared to the old process. We then utilized the Cross Validator to optimize the smoothing parameter for the Naive Bayes model. The result this time is much better than the old process which is `0.9246 (92.5%)`.

Our result is highlight the importance of the preprocessing process and hyperparameter tuning which can help us to improve the accuracy of the model. We can see the importance of the whole word in the text. The stopwords and special character definitely help to understand the writing style of an author which is used for author identification. We also can see the benefit of using the Cross Validator to optimize the smoothing parameter for the Naive Bayes model.