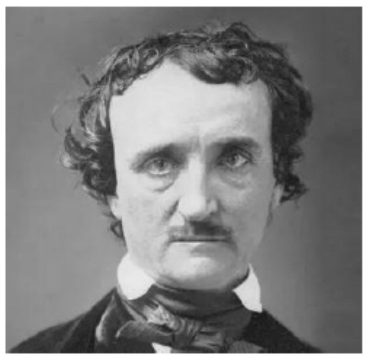
m.5.Assignment -> Spooky authorship identification via Apache Spark

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 [12]:

```
# 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", "5g") \
                    .config("spark.pyspark.python", sys.executable) \
                    .config("spark.pyspark.driver.python", sys.executa
                    .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)
```

```
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 [13]:

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

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
```

```python
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 [14]:
```
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

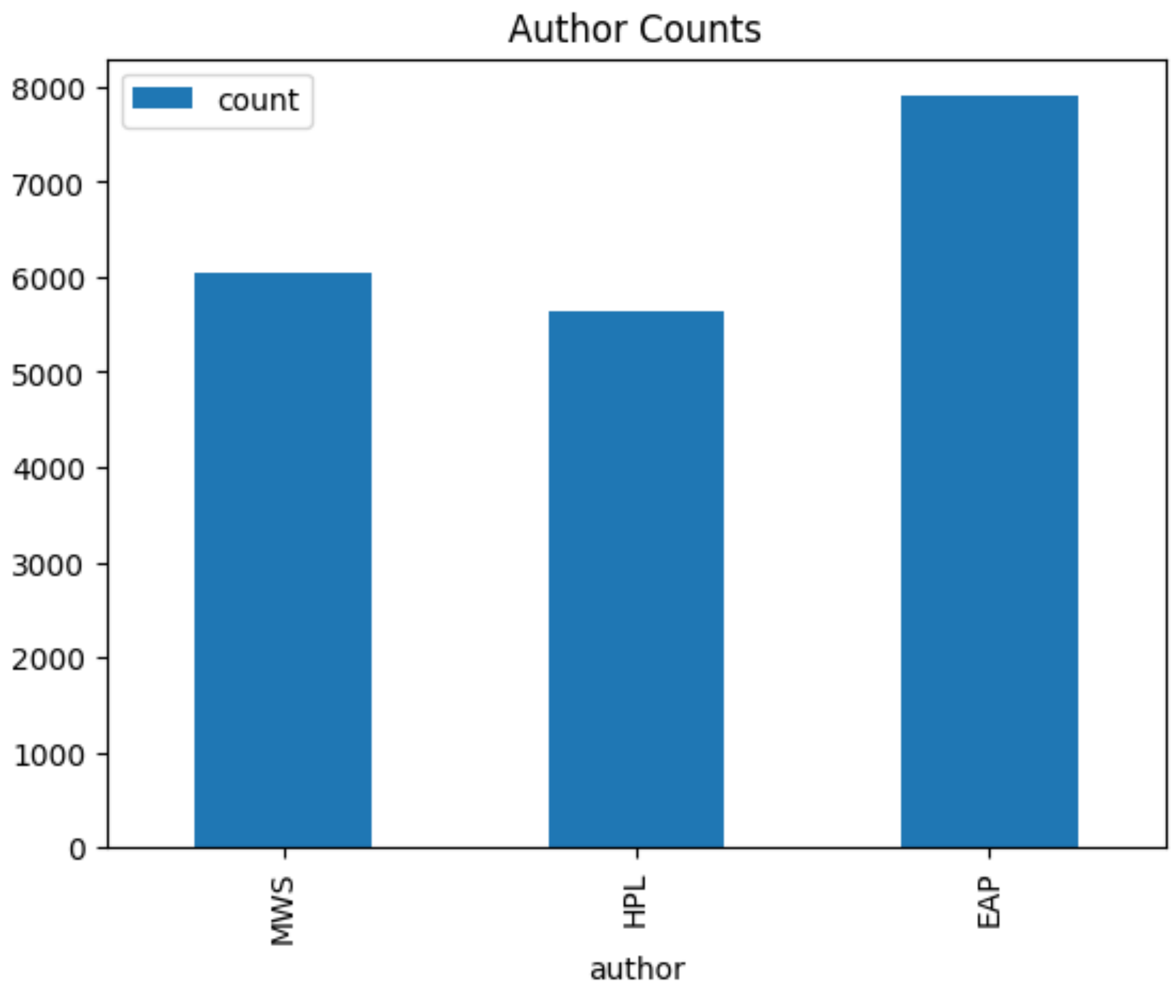
In [15]:
```
# Show text length statistics
print("\nText Length Statistics:")
train_df.select("text").summary().show()
```

```
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 [16]:
```
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 [17]:
```
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 [18]:
```
# 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[18]:
```
True
```

In [19]:

```
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)

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
|
+--------------------------------------------------------------------
--------------------------------------------------------------------
-------------------------+
|[process, however, afford, mean, ascertain, dimension, dungeon, might,
make, circuit, return, point, whence, set, without, aware, fact, perfec
tly, uniform, seem, wall]|
|[never, occur, fumbling, might, mere, mistake]
|
|[left, hand, gold, snuff, box, caper, hill, cut, manner, fantastic, st
ep, take, snuff, incessantly, air, great, possible, self, satisfaction]
|
+--------------------------------------------------------------------
--------------------------------------------------------------------
-------------------------+
only showing top 3 rows

Tokenized Test DataFrame:

+--------------------------------------------------------------------
--------------------------------------------------------------------
--------------------------------------------------------------------
--------------+
|tokens
|
+--------------------------------------------------------------------
--------------------------------------------------------------------
--------------------------------------------------------------------
--------------+
|[still, urge, leave, ireland, inquietude, impatience, father, think, b
est, yield]
|
|[fire, want, fanning, could, readily, fan, newspaper, government, gro
w, weak, doubt, leather, iron, acquire, durability, proportion, short,
time, pair, bellow, rotterdam, ever, stand, need, stitch, require, assi
stance, hammer]|
|[break, frail, door, find, two, cleanly, pick, human, skeleton, earthe
n, floor, number, singular, beetle, crawl, shadowy, corner]
|
+--------------------------------------------------------------------
--------------------------------------------------------------------
--------------------------------------------------------------------
--------------+
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 [20]:

```
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:")
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,[8732,34188,38308,46899,88590,89717,91767,100941,102036,13170
9,142239,148880,167503,183339,209518,214862,221027,237388,248200,25646
8,261675],[7.397357266296501,5.839212648249951,4.405800364152991,5.4279
16619830993,6.550059405909297,3.04493110139891,6.271346003440277,7.2432
06586469243,6.271346003440277,4.2987676073028025,4.007333185232471,4.26
1863050367351,4.706114183510672,3.99061970425873,4.706114183510672,6.44
8276711599355,7.579678823090456,3.4114644123018993,6.704210085736555,3.
5055369681858743,4.16523621467828])|
|(262144,[15313,17046,39275,113673,121401,256468],[5.382454245754237,5.
77139005191119,7.802822374404665,3.6075018948425623,5.0619823504794645,
3.5055369681858743])
|
|(262144,[4106,22370,28338,48648,55639,62790,75292,98424,102006,112947,
170414,199176,227860,232367,239343,245523,261845,261870],[4.76427010366
7746,5.349664422931245,7.802822374404665,5.146065467690006,3.3885100762
328153,4.667328158475516,6.663388091216301,4.376932380152138,5.34966442
2931245,15.370078677496563,5.349664422931245,6.326915854595088,4.912450
616508501,3.7641667180431533,8.783651627416392,4.564143922240285,5.6926
091740580755,3.433374521937644])
|
+----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
---------------------------------+
only showing top 3 rows

Test set also transformed (truncated):
```

```
25/07/22 18:29:52 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.0 MiB
25/07/22 18:29:52 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.0 MiB
+--------------------+
|            features|
+--------------------+
|(262144,[27506,31...|
|(262144,[1623,490...|
|(262144,[64603,72...|
+--------------------+
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 [21]:

```
from pyspark.ml.classification import DecisionTreeClassifier, Logistic
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml.feature import StringIndexer

# 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')

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
```

```python
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
```

```
25/07/22 18:29:52 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:29:53 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:29:53 WARN DAGScheduler: Broadcasting large task binary wit
h size 6.6 MiB
25/07/22 18:30:20 WARN DAGScheduler: Broadcasting large task binary wit
h size 1033.6 KiB
25/07/22 18:30:20 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.7 MiB
25/07/22 18:30:22 WARN MemoryStore: Not enough space to cache rdd_562_4
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:22 WARN MemoryStore: Not enough space to cache rdd_562_7
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:22 WARN MemoryStore: Not enough space to cache rdd_562_1
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:22 WARN BlockManager: Persisting block rdd_562_4 to disk
instead.
25/07/22 18:30:22 WARN BlockManager: Persisting block rdd_562_1 to disk
instead.
25/07/22 18:30:22 WARN MemoryStore: Not enough space to cache rdd_562_0
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:22 WARN BlockManager: Persisting block rdd_562_0 to disk
instead.
25/07/22 18:30:22 WARN BlockManager: Persisting block rdd_562_7 to disk
instead.
25/07/22 18:30:22 WARN MemoryStore: Not enough space to cache rdd_562_2
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:22 WARN BlockManager: Persisting block rdd_562_2 to disk
instead.
```

```
25/07/22 18:30:22 WARN MemoryStore: Not enough space to cache rdd_562_6
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:22 WARN BlockManager: Persisting block rdd_562_6 to disk
instead.
25/07/22 18:30:22 WARN MemoryStore: Not enough space to cache rdd_562_5
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:22 WARN BlockManager: Persisting block rdd_562_5 to disk
instead.
25/07/22 18:30:22 WARN MemoryStore: Not enough space to cache rdd_562_3
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:22 WARN BlockManager: Persisting block rdd_562_3 to disk
instead.
25/07/22 18:30:28 WARN MemoryStore: Not enough space to cache rdd_562_4
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:29 WARN MemoryStore: Not enough space to cache rdd_562_0
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:29 WARN MemoryStore: Not enough space to cache rdd_562_2
in memory! (computed 630.3 MiB so far)
25/07/22 18:30:29 WARN MemoryStore: Not enough space to cache rdd_562_6
in memory! (computed 951.0 MiB so far)
25/07/22 18:30:29 WARN MemoryStore: Not enough space to cache rdd_562_7
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:34 WARN MemoryStore: Not enough space to cache rdd_562_1
in memory! (computed 951.0 MiB so far)
25/07/22 18:30:35 WARN MemoryStore: Not enough space to cache rdd_562_3
in memory! (computed 951.0 MiB so far)
25/07/22 18:30:35 WARN MemoryStore: Not enough space to cache rdd_562_5
in memory! (computed 951.0 MiB so far)
25/07/22 18:30:38 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.7 MiB
25/07/22 18:30:39 WARN MemoryStore: Not enough space to cache rdd_562_6
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:39 WARN MemoryStore: Not enough space to cache rdd_562_1
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:40 WARN MemoryStore: Not enough space to cache rdd_562_0
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:40 WARN MemoryStore: Not enough space to cache rdd_562_5
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:40 WARN MemoryStore: Not enough space to cache rdd_562_2
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:40 WARN MemoryStore: Not enough space to cache rdd_562_4
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:40 WARN MemoryStore: Not enough space to cache rdd_562_3
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:40 WARN MemoryStore: Not enough space to cache rdd_562_7
in memory! (computed 630.3 MiB so far)
25/07/22 18:30:45 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.7 MiB
25/07/22 18:30:45 WARN MemoryStore: Not enough space to cache rdd_562_7
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:45 WARN MemoryStore: Not enough space to cache rdd_562_3
in memory! (computed 272.8 MiB so far)
```

```
25/07/22 18:30:45 WARN MemoryStore: Not enough space to cache rdd_562_0
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:46 WARN MemoryStore: Not enough space to cache rdd_562_5
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:46 WARN MemoryStore: Not enough space to cache rdd_562_1
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:46 WARN MemoryStore: Not enough space to cache rdd_562_6
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:46 WARN MemoryStore: Not enough space to cache rdd_562_2
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:46 WARN MemoryStore: Not enough space to cache rdd_562_4
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:51 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.7 MiB
25/07/22 18:30:51 WARN MemoryStore: Not enough space to cache rdd_562_3
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:51 WARN MemoryStore: Not enough space to cache rdd_562_4
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:52 WARN MemoryStore: Not enough space to cache rdd_562_7
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:52 WARN MemoryStore: Not enough space to cache rdd_562_2
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:52 WARN MemoryStore: Not enough space to cache rdd_562_1
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:52 WARN MemoryStore: Not enough space to cache rdd_562_0
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:52 WARN MemoryStore: Not enough space to cache rdd_562_5
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:52 WARN MemoryStore: Not enough space to cache rdd_562_6
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:57 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.7 MiB
25/07/22 18:30:58 WARN MemoryStore: Not enough space to cache rdd_562_2
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:58 WARN MemoryStore: Not enough space to cache rdd_562_7
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:58 WARN MemoryStore: Not enough space to cache rdd_562_4
in memory! (computed 272.8 MiB so far)
25/07/22 18:30:59 WARN MemoryStore: Not enough space to cache rdd_562_3
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:59 WARN MemoryStore: Not enough space to cache rdd_562_5
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:59 WARN MemoryStore: Not enough space to cache rdd_562_0
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:59 WARN MemoryStore: Not enough space to cache rdd_562_1
in memory! (computed 419.0 MiB so far)
25/07/22 18:30:59 WARN MemoryStore: Not enough space to cache rdd_562_6
in memory! (computed 419.0 MiB so far)
25/07/22 18:31:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
```

```
Decision Tree Accuracy:  0.45722558568044225
F1:  0.3483428101899689 3
```

```
25/07/22 18:31:04 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:05 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:06 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
```

```
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:08 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
```

```
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:09 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
```

```
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:10 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
```

```
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:12 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
```

```
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:13 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
```

```
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:14 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:15 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
```

```
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
```

```
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.7 MiB
25/07/22 18:31:19 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.7 MiB
```

```
Logistic Regression Accuracy:  0.7143985259278758
F1:  0.7145775035310242
```

```
25/07/22 18:31:19 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:19 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:31:20 WARN DAGScheduler: Broadcasting large task binary wit
h size 6.6 MiB
25/07/22 18:31:44 WARN DAGScheduler: Broadcasting large task binary wit
h size 1033.6 KiB
```

```
25/07/22 18:31:44 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.8 MiB
25/07/22 18:31:45 WARN MemoryStore: Not enough space to cache rdd_1093_
0 in memory! (computed 272.8 MiB so far)
25/07/22 18:31:45 WARN BlockManager: Persisting block rdd_1093_0 to dis
k instead.
25/07/22 18:31:45 WARN MemoryStore: Not enough space to cache rdd_1093_
3 in memory! (computed 272.8 MiB so far)
25/07/22 18:31:45 WARN BlockManager: Persisting block rdd_1093_3 to dis
k instead.
25/07/22 18:31:45 WARN MemoryStore: Not enough space to cache rdd_1093_
2 in memory! (computed 272.8 MiB so far)
25/07/22 18:31:45 WARN BlockManager: Persisting block rdd_1093_2 to dis
k instead.
25/07/22 18:31:45 WARN MemoryStore: Not enough space to cache rdd_1093_
5 in memory! (computed 272.8 MiB so far)
25/07/22 18:31:45 WARN BlockManager: Persisting block rdd_1093_5 to dis
k instead.
25/07/22 18:31:45 WARN MemoryStore: Not enough space to cache rdd_1093_
6 in memory! (computed 419.1 MiB so far)
25/07/22 18:31:46 WARN BlockManager: Persisting block rdd_1093_6 to dis
k instead.
25/07/22 18:31:46 WARN MemoryStore: Not enough space to cache rdd_1093_
1 in memory! (computed 419.1 MiB so far)
25/07/22 18:31:46 WARN BlockManager: Persisting block rdd_1093_1 to dis
k instead.
25/07/22 18:31:46 WARN MemoryStore: Not enough space to cache rdd_1093_
4 in memory! (computed 419.1 MiB so far)
25/07/22 18:31:46 WARN BlockManager: Persisting block rdd_1093_4 to dis
k instead.
25/07/22 18:31:46 WARN MemoryStore: Not enough space to cache rdd_1093_
7 in memory! (computed 419.1 MiB so far)
25/07/22 18:31:46 WARN BlockManager: Persisting block rdd_1093_7 to dis
k instead.
25/07/22 18:31:50 WARN MemoryStore: Not enough space to cache rdd_1093_
0 in memory! (computed 630.3 MiB so far)
25/07/22 18:31:50 WARN MemoryStore: Not enough space to cache rdd_1093_
7 in memory! (computed 419.1 MiB so far)
25/07/22 18:31:50 WARN MemoryStore: Not enough space to cache rdd_1093_
2 in memory! (computed 630.3 MiB so far)
25/07/22 18:31:51 WARN MemoryStore: Not enough space to cache rdd_1093_
4 in memory! (computed 419.1 MiB so far)
25/07/22 18:31:54 WARN MemoryStore: Not enough space to cache rdd_1093_
5 in memory! (computed 419.1 MiB so far)
25/07/22 18:31:54 WARN MemoryStore: Not enough space to cache rdd_1093_
1 in memory! (computed 419.1 MiB so far)
25/07/22 18:31:55 WARN MemoryStore: Not enough space to cache rdd_1093_
3 in memory! (computed 951.1 MiB so far)
25/07/22 18:31:57 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.8 MiB
25/07/22 18:31:58 WARN MemoryStore: Not enough space to cache rdd_1093_
7 in memory! (computed 177.0 MiB so far)
```

```
25/07/22 18:31:58 WARN MemoryStore: Not enough space to cache rdd_1093_
0 in memory! (computed 177.0 MiB so far)
25/07/22 18:31:58 WARN MemoryStore: Not enough space to cache rdd_1093_
2 in memory! (computed 177.0 MiB so far)
25/07/22 18:31:58 WARN MemoryStore: Not enough space to cache rdd_1093_
5 in memory! (computed 177.0 MiB so far)
25/07/22 18:31:58 WARN MemoryStore: Not enough space to cache rdd_1093_
3 in memory! (computed 177.0 MiB so far)
25/07/22 18:31:58 WARN MemoryStore: Not enough space to cache rdd_1093_
4 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:00 WARN MemoryStore: Not enough space to cache rdd_1093_
1 in memory! (computed 951.1 MiB so far)
25/07/22 18:32:03 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.8 MiB
25/07/22 18:32:03 WARN MemoryStore: Not enough space to cache rdd_1093_
2 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:03 WARN MemoryStore: Not enough space to cache rdd_1093_
1 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:03 WARN MemoryStore: Not enough space to cache rdd_1093_
7 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:03 WARN MemoryStore: Not enough space to cache rdd_1093_
0 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:03 WARN MemoryStore: Not enough space to cache rdd_1093_
4 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:03 WARN MemoryStore: Not enough space to cache rdd_1093_
5 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:05 WARN MemoryStore: Not enough space to cache rdd_1093_
3 in memory! (computed 951.1 MiB so far)
25/07/22 18:32:07 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.8 MiB
25/07/22 18:32:07 WARN MemoryStore: Not enough space to cache rdd_1093_
1 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:07 WARN MemoryStore: Not enough space to cache rdd_1093_
2 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:07 WARN MemoryStore: Not enough space to cache rdd_1093_
7 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:07 WARN MemoryStore: Not enough space to cache rdd_1093_
5 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:07 WARN MemoryStore: Not enough space to cache rdd_1093_
3 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:07 WARN MemoryStore: Not enough space to cache rdd_1093_
0 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:09 WARN MemoryStore: Not enough space to cache rdd_1093_
4 in memory! (computed 951.1 MiB so far)
25/07/22 18:32:11 WARN DAGScheduler: Broadcasting large task binary wit
h size 8.9 MiB
25/07/22 18:32:12 WARN MemoryStore: Not enough space to cache rdd_1093_
0 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:12 WARN MemoryStore: Not enough space to cache rdd_1093_
4 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:12 WARN MemoryStore: Not enough space to cache rdd_1093_
5 in memory! (computed 177.0 MiB so far)
```

```
25/07/22 18:32:12 WARN MemoryStore: Not enough space to cache rdd_1093_
1 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:12 WARN MemoryStore: Not enough space to cache rdd_1093_
3 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:12 WARN MemoryStore: Not enough space to cache rdd_1093_
7 in memory! (computed 177.0 MiB so far)
25/07/22 18:32:14 WARN MemoryStore: Not enough space to cache rdd_1093_
2 in memory! (computed 951.1 MiB so far)
25/07/22 18:32:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.2 MiB
25/07/22 18:32:16 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.2 MiB
```

```
Random Forest Accuracy:  0.41458278494340617
F1:  0.25258294638097384
```

```
25/07/22 18:32:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:32:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:32:17 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
25/07/22 18:32:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
```

```
Naive Bayes Accuracy:  0.8239010265859437
F1:  0.8239138405654137
```

Stage 4: Evaluation & Visualization

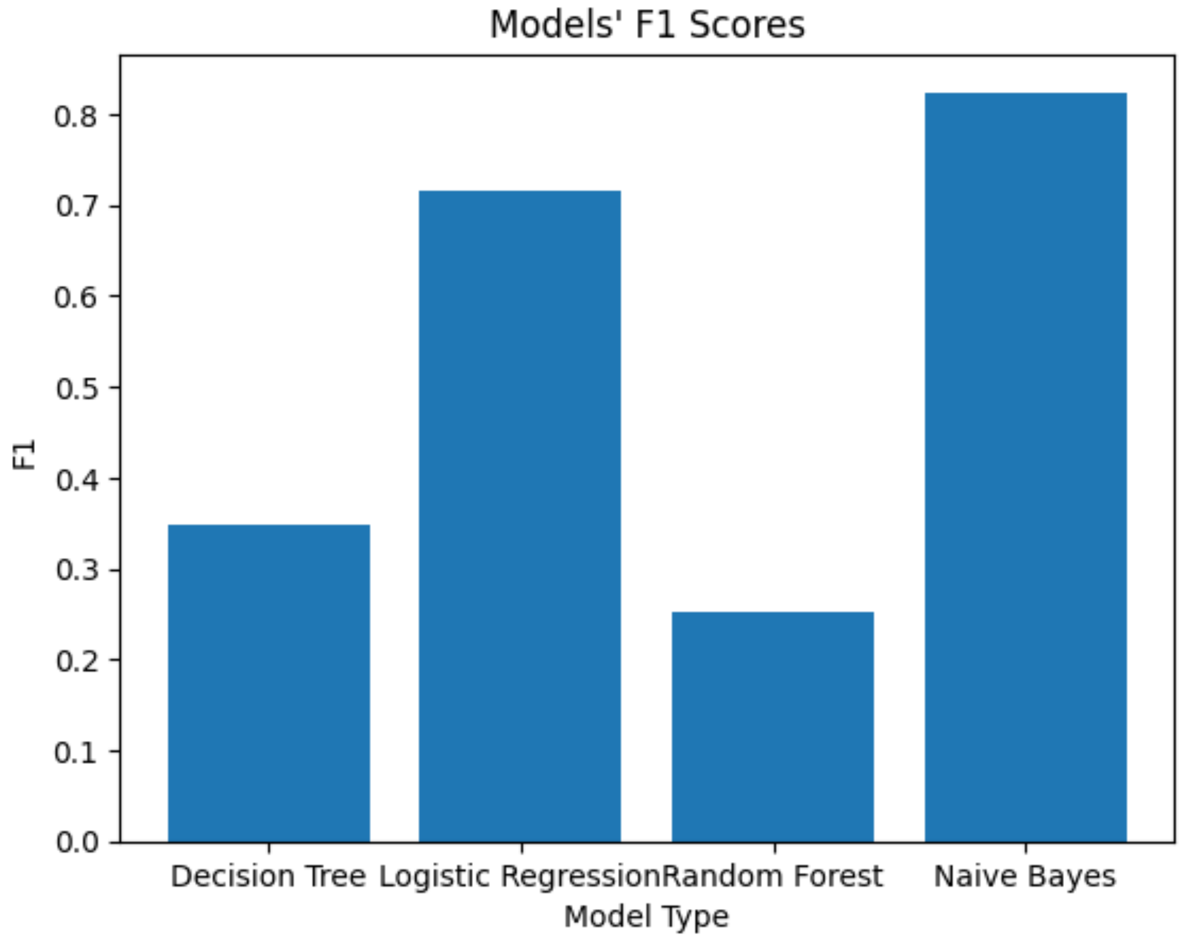
9. Choose a metric strategy to assess algorithmic performance like accuracy, precision, recall, or F1 score.
10. Visualize confusion matrix, correlations, and similar.
11. Identify important features contributing to classification.
12. Write a 2-3 sentence minimum of findings, learnings, and what you would do next.

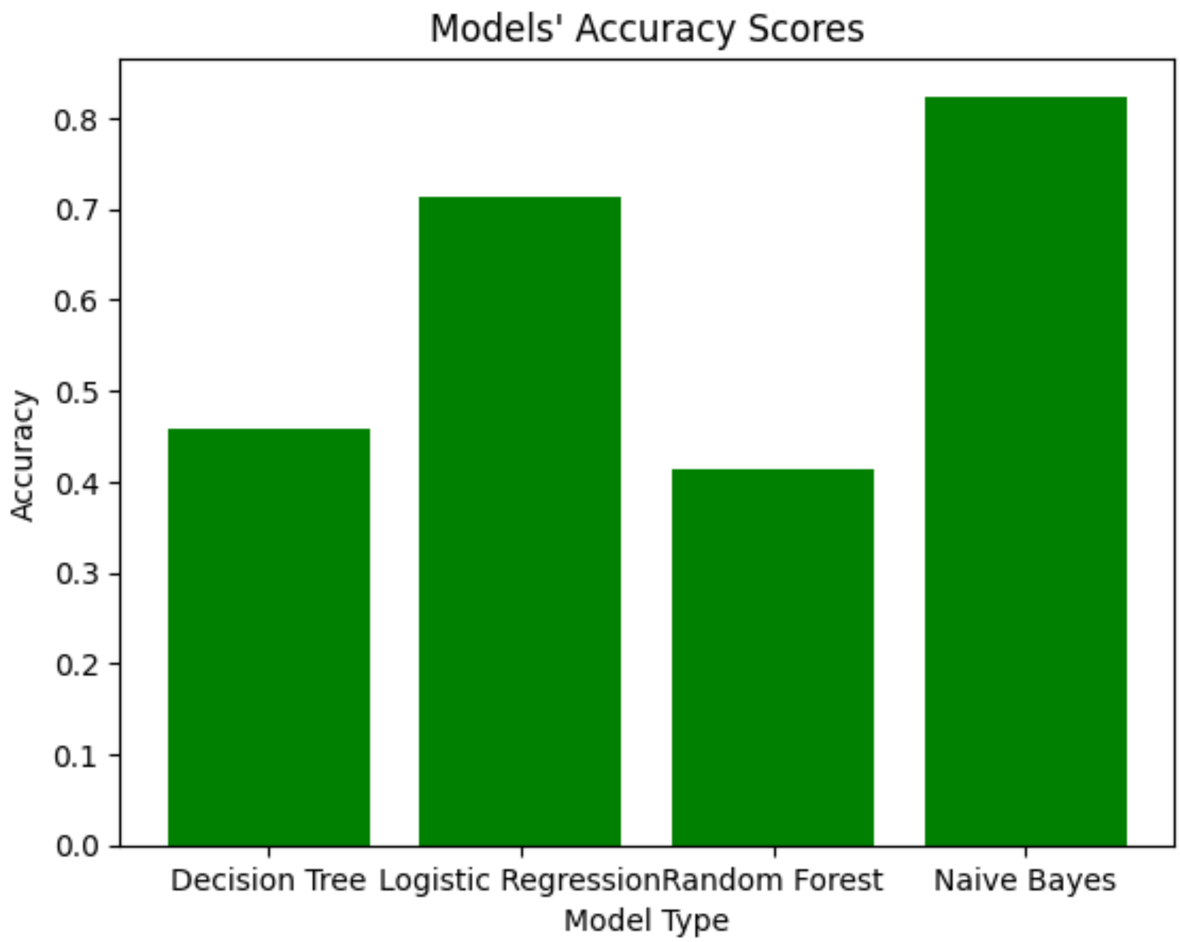
In [22]:

```python
#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")

#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")
```






```
25/07/22 18:32:18 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:32:19 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:32:19 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.7 MiB
Decision Tree Confusion Matrix:
prediction   EAP  MWS  HPL
label
EAP         1511    9   23
MWS         1051  117   22
HPL          952    5  109
25/07/22 18:32:19 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.7 MiB
25/07/22 18:32:19 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.2 MiB
Logistic Regression Confusion Matrix:
prediction   EAP  MWS  HPL
label
EAP         1092  258  193
MWS          195  877  118
HPL          174  147  745
```

```
25/07/22 18:32:19 WARN DAGScheduler: Broadcasting large task binary wit
h size 4.1 MiB
25/07/22 18:32:19 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
```

```
Random Forest Confusion Matrix:
prediction   EAP  MWS  HPL
label
EAP         1543    0    0
MWS         1164   26    0
HPL         1060    0    6
Naive Bayes Confusion Matrix:
prediction   EAP  MWS  HPL
label
EAP         1264  166  113
MWS          128  990   72
HPL          127   63  876
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: 0.25314092867737387, mean: 3.814697265625e-06,
difference: 0.25313711398010824.
Logistic Regression Coefficient Matrix:
class 0 (one of the authors): Least: -87.98888422213571, Mean: -0.01704
844054384448, Difference: 87.97183578159186.
class 1 (one of the authors): Least: -147.23891295363183, Mean: -0.0228
73091573409447, Difference: 147.2160398620584.
class 2 (one of the authors): Least: -110.81888991803767, Mean: 0.03992
1532117253924, Difference: 110.85881145015492.
Random Forest max value: 0.04378939115327409, mean: 3.814697265625001e-
06, difference: 0.04378557645600847.
Naive Bayes Theta Matrix:
class 0 (one of the authors): Least: -13.466557706750198, Mean: -13.338
330643484388, Difference: 0.12822706326580935.
class 1 (one of the authors): Least: -13.327273657600413, Mean: -13.231
574742435393, Difference: 0.09569891516501983.
class 2 (one of the authors): Least: -13.372665150185114, Mean: -13.251
636733969848, Difference: 0.12102841621526572.
```

```
25/07/22 18:32:20 WARN DAGScheduler: Broadcasting large task binary wit
h size 10.1 MiB
```

CONCLUSIONS

We observe principally that greater quantities of training data produce more accurate models. Those teammates that were able to train the models with the full data sets saw up to 82 percent accuracy on the Naive Bayes model. Likewise, on my machine, on which I need to greatly limit training and testing dataframes to simply complete a run, the Naive Bayes achieves a 58 percent accuracy.

Additionally, the use of the HashingTF pipeline stage has the side effect that hashed tokens are no longer directly accessable because of collisions. Therefore, we can investigate hash values to learn some interesting facts about the models. We see for the Decision Tree and Random Forest that its maximum importance value vastly outweighs the mean for the sparse vector from which it is sourced. This could be an indication of extremely important weights at the top, or that this is a flawed approach. We notice closer differences for each class within the Random Forest coefficient matrix and Naive Bayes theta matrix.