In [1]:
```python
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split

import re
import string
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
```

```
C:\Users\Owner\Desktop\a\Lib\site-packages\pandas\core\arrays\masked.py:60: UserWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (version '1.3.5' currently install
ed).
  from pandas.core import (
```

In [2]:
```python
df = pd.read_csv(r"C:\Users\Owner\Downloads\Sentiment_Analysis\reviews_badminton\data.csv")
df.head()
```

Out[2]:

	Reviewer Name	Review Title	Place of Review	Up Votes	Down Votes	Month	Review text	Ratings
0	Kamal Suresh	Nice product	Certified Buyer, Chirakkal	889.0	64.0	Feb 2021	Nice product, good quality, but price is now r...	4
1	Flipkart Customer	Don't waste your money	Certified Buyer, Hyderabad	109.0	6.0	Feb 2021	They didn't supplied Yonex Mavis 350. Outside ...	1
2	A. S. Raja Srinivasan	Did not meet expectations	Certified Buyer, Dharmapuri	42.0	3.0	Apr 2021	Worst product. Damaged shuttlecocks packed in ...	1
3	Suresh Narayanasamy	Fair	Certified Buyer, Chennai	25.0	1.0	NaN	Quite O. K. , but nowadays the quality of the...	3
4	ASHIK P A	Over priced	NaN	147.0	24.0	Apr 2016	Over pricedJust â?¹620 ..from retailer.I didn'...	1

In [3]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8518 entries, 0 to 8517
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Reviewer Name    8508 non-null   object 
 1   Review Title     8508 non-null   object 
 2   Place of Review  8468 non-null   object 
 3   Up Votes         8508 non-null   float64
 4   Down Votes       8508 non-null   float64
 5   Month            8053 non-null   object 
 6   Review text      8510 non-null   object 
 7   Ratings          8518 non-null   int64  
dtypes: float64(2), int64(1), object(5)
memory usage: 532.5+ KB
```

In [4]:
```python
df.isnull().sum()
```

Out[4]:
```
Reviewer Name       10
Review Title        10
Place of Review     50
Up Votes            10
Down Votes          10
Month              465
Review text          8
Ratings              0
dtype: int64
```

In [5]:
```python
df = df.dropna()
```

In [6]:
```python
df.isnull().sum()
```

Out[6]:
```
Reviewer Name      0
Review Title       0
Place of Review    0
Up Votes           0
Down Votes         0
Month              0
Review text        0
Ratings            0
dtype: int64
```

In [7]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 8013 entries, 0 to 8507
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Reviewer Name    8013 non-null   object 
 1   Review Title     8013 non-null   object 
 2   Place of Review  8013 non-null   object 
 3   Up Votes         8013 non-null   float64
 4   Down Votes       8013 non-null   float64
 5   Month            8013 non-null   object 
 6   Review text      8013 non-null   object 
 7   Ratings          8013 non-null   int64  
dtypes: float64(2), int64(1), object(5)
memory usage: 563.4+ KB
```

In [8]:
```python
ratings_dict = {5.0 : 1 , 4.0 : 1 , 3.0 : 0 , 2.0 : 0 , 1.0 : 0}

#now by using the ratings columns we r adding new column

df['sentiment'] = df['Ratings'].map(ratings_dict)
```

In [9]:
```python
df.head()
```

Out[9]:

	Reviewer Name	Review Title	Place of Review	Up Votes	Down Votes	Month	Review text	Ratings	sentiment
0	Kamal Suresh	Nice product	Certified Buyer, Chirakkal	889.0	64.0	Feb 2021	Nice product, good quality, but price is now r...	4	1
1	Flipkart Customer	Don't waste your money	Certified Buyer, Hyderabad	109.0	6.0	Feb 2021	They didn't supplied Yonex Mavis 350. Outside ...	1	0
2	A. S. Raja Srinivasan	Did not meet expectations	Certified Buyer, Dharmapuri	42.0	3.0	Apr 2021	Worst product. Damaged shuttlecocks packed in ...	1	0
5	Baji Sankar	Mind-blowing purchase	Certified Buyer, Hyderabad	173.0	45.0	Oct 2018	Good quality product. Delivered on time.READ MORE	5	1
6	Flipkart Customer	Must buy!	Certified Buyer, Doom Dooma	403.0	121.0	Jan 2020	BEST PURCHASE It is a good quality and is more...	5	1

In [10]:
```python
x = df['Review text']
y = df['sentiment']
```

In [11]:
```python
#now lets split

x_train , x_test , y_train , y_test = train_test_split(x,y , test_size= 0.25 , random_state= 50)
```

In [12]:
```python
#now doing the data cleaning and preprocessing on train and test data

# Preprocessing functions
def clean_text(text):
    text = re.sub(r"[^a-zA-Z]", " ", text)
    text = re.sub(r'\W+', ' ', text)
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = text.lower()
    stop_words = set(stopwords.words('english'))
    words = text.split()
    cleaned_words = [word for word in words if word not in stop_words]
    return ' '.join(cleaned_words)

def lemmatize_text(text):
    lemmatizer = WordNetLemmatizer()
    tokens = nltk.word_tokenize(text)
    lemmatized_words = [lemmatizer.lemmatize(word) for word in tokens]
    return ' '.join(lemmatized_words)
```

In [13]:
```python
#now we r applying the above text cleaning to the x_train

x_train = x_train.apply(clean_text)
x_train = x_train.apply(lemmatize_text)
x_train.shape
```

Out[13]:
```
(6009,)
```

In [14]:
```python
#now apply cleaend data to x_test

x_test = x_test.apply(clean_text)
x_test = x_test.apply(lemmatize_text)
x_test.shape
```

Out[14]:
```
(2004,)
```

In [15]:
```python
!pip install mlflow
```

```
Requirement already satisfied: mlflow in c:\users\owner\desktop\a\lib\site-packages (2.11.3)
Requirement already satisfied: click<9,>=7.0 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (8.1.7)
Requirement already satisfied: cloudpickle<4 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (2.2.1)
Requirement already satisfied: entrypoints<1 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (0.4)
Requirement already satisfied: gitpython<4,>=3.1.9 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (3.1.42)
Requirement already satisfied: pyyaml<7,>=5.1 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (6.0)
Requirement already satisfied: protobuf<5,>=3.12.0 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (4.25.3)
Requirement already satisfied: pytz<2025 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (2024.1)
Requirement already satisfied: requests<3,>=2.17.3 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (2.29.0)
Requirement already satisfied: packaging<24 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (23.0)
Requirement already satisfied: importlib-metadata!=4.7.0,<8,>=3.7.0 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (6.0.0)
Requirement already satisfied: sqlparse<1,>=0.4.0 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (0.4.4)
Requirement already satisfied: alembic!=1.10.0,<2 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (1.13.1)
Requirement already satisfied: docker<8,>=4.0.0 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (7.0.0)
Requirement already satisfied: Flask<4 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (2.2.2)
Requirement already satisfied: numpy<2 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (1.26.4)
Requirement already satisfied: scipy<2 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (1.12.0)
Requirement already satisfied: pandas<3 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (2.2.1)
Requirement already satisfied: querystring-parser<2 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (1.2.4)
Requirement already satisfied: sqlalchemy<3,>=1.4.0 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (1.4.39)
Requirement already satisfied: scikit-learn<2 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (1.2.2)
Requirement already satisfied: pyarrow<16,>=4.0.0 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (11.0.0)
Requirement already satisfied: markdown<4,>=3.3 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (3.4.1)
Requirement already satisfied: matplotlib<4 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (3.7.1)
Requirement already satisfied: graphene<4 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (3.3)
Requirement already satisfied: waitress<4 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (3.0.0)
Requirement already satisfied: Jinja2<4,>=3.0 in c:\users\owner\desktop\a\lib\site-packages (from mlflow) (3.1.2)
Requirement already satisfied: Mako in c:\users\owner\desktop\a\lib\site-packages (from alembic!=1.10.0,<2->mlflow) (1.3.2)
Requirement already satisfied: typing-extensions>=4 in c:\users\owner\desktop\a\lib\site-packages (from alembic!=1.10.0,<2->mlflow) (4.10.0)
Requirement already satisfied: colorama in c:\users\owner\desktop\a\lib\site-packages (from click<9,>=7.0->mlflow) (0.4.6)
Requirement already satisfied: urllib3>=1.26.0 in c:\users\owner\desktop\a\lib\site-packages (from docker<8,>=4.0.0->mlflow) (1.26.16)
Requirement already satisfied: pywin32>=304 in c:\users\owner\desktop\a\lib\site-packages (from docker<8,>=4.0.0->mlflow) (305.1)
Requirement already satisfied: Werkzeug>=2.2.2 in c:\users\owner\desktop\a\lib\site-packages (from Flask<4->mlflow) (3.0.1)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\owner\desktop\a\lib\site-packages (from Flask<4->mlflow) (2.1.2)
Requirement already satisfied: gitdb<5,>=4.0.1 in c:\users\owner\desktop\a\lib\site-packages (from gitpython<4,>=3.1.9->mlflow) (4.0.11)
Requirement already satisfied: graphql-core<3.3,>=3.1 in c:\users\owner\desktop\a\lib\site-packages (from graphene<4->mlflow) (3.2.3)
Requirement already satisfied: graphql-relay<3.3,>=3.1 in c:\users\owner\desktop\a\lib\site-packages (from graphene<4->mlflow) (3.2.0)
Requirement already satisfied: aniso8601<10,>=8 in c:\users\owner\desktop\a\lib\site-packages (from graphene<4->mlflow) (9.0.1)
Requirement already satisfied: zipp>=0.5 in c:\users\owner\desktop\a\lib\site-packages (from importlib-metadata!=4.7.0,<8,>=3.7.0->mlflow) (3.11.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\owner\desktop\a\lib\site-packages (from Jinja2<4,>=3.0->mlflow) (2.1.5)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\owner\desktop\a\lib\site-packages (from matplotlib<4->mlflow) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\owner\desktop\a\lib\site-packages (from matplotlib<4->mlflow) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\owner\desktop\a\lib\site-packages (from matplotlib<4->mlflow) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\owner\desktop\a\lib\site-packages (from matplotlib<4->mlflow) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in c:\users\owner\desktop\a\lib\site-packages (from matplotlib<4->mlflow) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\owner\desktop\a\lib\site-packages (from matplotlib<4->mlflow) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\owner\desktop\a\lib\site-packages (from matplotlib<4->mlflow) (2.9.0.post0)
Requirement already satisfied: tzdata>=2022.7 in c:\users\owner\desktop\a\lib\site-packages (from pandas<3->mlflow) (2024.1)
Requirement already satisfied: six in c:\users\owner\desktop\a\lib\site-packages (from querystring-parser<2->mlflow) (1.16.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\owner\desktop\a\lib\site-packages (from requests<3,>=2.17.3->mlflow) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\owner\desktop\a\lib\site-packages (from requests<3,>=2.17.3->mlflow) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\owner\desktop\a\lib\site-packages (from requests<3,>=2.17.3->mlflow) (2023.5.7)
Requirement already satisfied: joblib>=1.1.1 in c:\users\owner\desktop\a\lib\site-packages (from scikit-learn<2->mlflow) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\owner\desktop\a\lib\site-packages (from scikit-learn<2->mlflow) (3.3.0)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\owner\desktop\a\lib\site-packages (from sqlalchemy<3,>=1.4.0->mlflow) (2.0.1)
Requirement already satisfied: smmap<6,>=3.0.1 in c:\users\owner\desktop\a\lib\site-packages (from gitdb<5,>=4.0.1->gitpython<4,>=3.1.9->mlflow) (5.0.1)
```

In [16]:
```python
import mlflow
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import GridSearchCV
from sklearn.pipeline import Pipeline
import time
import joblib
import os
```

In [17]:
```python
import warnings
warnings.filterwarnings("ignore")

# mlflow.set_tracking_uri("sqlite:///mlflow_1.db")

mlflow.set_experiment("sentimental_analysis_prediction")
```

Out[17]:
```
<Experiment: artifact_location='file:///C:/Users/Owner/mlruns/460781658887541175', creation_time=1711639873270, experiment_id='460781658887541175', last_update_time=1711639873270,
lifecycle_stage='active', name='sentimental_analysis_prediction', tags={}>
```

In [18]:
```python
#pipeline creation

pipelines = {
    'knn': Pipeline([
        ('tfidf', TfidfVectorizer()),
        ('classifier', KNeighborsClassifier())
    ]),
    'svc': Pipeline([
        ('tfidf', TfidfVectorizer()),
        ('classifier', SVC())
    ]),
    'logistic_regression': Pipeline([
        ('tfidf', TfidfVectorizer()),
        ('classifier', LogisticRegression())
    ]),
    'random_forest': Pipeline([
        ('tfidf', TfidfVectorizer()),
        ('classifier', RandomForestClassifier())
    ]),
    'decision_tree': Pipeline([
        ('tfidf', TfidfVectorizer()),
        ('classifier', DecisionTreeClassifier())
    ])
}

# Define parameter grid for each algorithm
param_grids = {
    'knn': [
        {
            'tfidf__max_features': [1000, 2000, 3000],
            'classifier__n_neighbors': [3, 5, 7],
            'classifier__p': [1, 2, 3]
        }
    ],
    'svc': [
        {
            'tfidf__max_features': [1000, 2000, 3000],
            'classifier__kernel': ['rbf'],
            'classifier__C': [0.1, 1, 10]
        },
        {
            'tfidf__max_features': [1000, 2000, 3000],
            'classifier__kernel': ['linear'],
            'classifier__C': [0.1, 1, 10]
        }
    ],
    'logistic_regression': [
        {
            'tfidf__max_features': [1000, 2000, 3000],
            'classifier__C': [0.1, 1, 10],
            'classifier__penalty': ['l1', 'l2']
        }
    ],
    'random_forest': [
        {
            'tfidf__max_features': [1000, 2000, 3000],
            'classifier__n_estimators': [50, 100, 200]
        }
    ],
    'decision_tree': [
        {
            'tfidf__max_features': [1000, 2000, 3000],
            'classifier__max_depth': [None, 5, 10]
        }
    ],

}
```

In [19]:
```python
best_models = {}

# Run the Pipeline
for algo in pipelines.keys():
    print("*"*10, algo, "*"*10)
    grid_search = GridSearchCV(estimator=pipelines[algo],
                               param_grid=param_grids[algo],
                               cv=5,
                               scoring='accuracy',
                               return_train_score=True,
                               verbose=1
                              )

    mlflow.sklearn.autolog(max_tuning_runs=None)

    with mlflow.start_run() as run:
        %time grid_search.fit(x_train, y_train)

    # print('Score on Train Data: ', grid_search.best_score_)
    print('Score on Test Data: ', grid_search.score(x_test, y_test))
```

```
********** knn **********
2024/03/28 21:41:18 WARNING mlflow.utils.git_utils: Failed to import Git (the Git executable is probably not on your PATH), so Git SHA is not available. Error: Failed to initializ
e: Bad git executable.
The git executable must be specified in one of the following ways:
    - be included in your $PATH
    - be set via $GIT_PYTHON_GIT_EXECUTABLE
    - explicitly set via git.refresh()

All git commands will error until this is rectified.

This initial message can be silenced or aggravated in the future by setting the
$GIT_PYTHON_REFRESH environment variable. Use one of the following values:
    - quiet|q|silence|s|silent|none|n|0: for no message or exception
    - warn|w|warning|log|l|1: for a warning message (logged at level CRITICAL, displayed by default)
    - error|e|exception|raise|r|2: for a raised exception

Example:
    export GIT_PYTHON_REFRESH=quiet

2024/03/28 21:41:18 WARNING mlflow.sklearn: Unrecognized dataset type <class 'pandas.core.series.Series'>. Dataset logging skipped.
Fitting 5 folds for each of 27 candidates, totalling 135 fits
CPU times: total: 8min 27s
Wall time: 4min 6s
Score on Test Data:  0.872255489021956
********** svc **********
2024/03/28 21:45:26 WARNING mlflow.sklearn: Unrecognized dataset type <class 'pandas.core.series.Series'>. Dataset logging skipped.
Fitting 5 folds for each of 18 candidates, totalling 90 fits
CPU times: total: 3min 19s
Wall time: 3min 32s
Score on Test Data:  0.8827345309381237
********** logistic_regression **********
2024/03/28 21:49:00 WARNING mlflow.sklearn: Unrecognized dataset type <class 'pandas.core.series.Series'>. Dataset logging skipped.
Fitting 5 folds for each of 18 candidates, totalling 90 fits
CPU times: total: 20.7 s
Wall time: 29.1 s
Score on Test Data:  0.8842315369261478
********** random_forest **********
2024/03/28 21:49:30 WARNING mlflow.sklearn: Unrecognized dataset type <class 'pandas.core.series.Series'>. Dataset logging skipped.
Fitting 5 folds for each of 9 candidates, totalling 45 fits
CPU times: total: 6min 31s
Wall time: 7min 3s
Score on Test Data:  0.8847305389221557
********** decision_tree **********
2024/03/28 21:56:34 WARNING mlflow.sklearn: Unrecognized dataset type <class 'pandas.core.series.Series'>. Dataset logging skipped.
Fitting 5 folds for each of 9 candidates, totalling 45 fits
CPU times: total: 18.8 s
Wall time: 27.9 s
Score on Test Data:  0.8637724550898204
```

In [20]:
```python
# Stop the auto logger
mlflow.sklearn.autolog(disable=True)
```

In [23]:
```python
## we can use Custom Experiment Tracking and Database Integration with MLFlow

dev = "Sneha Dahat"
best_models = {}

for algo in pipelines.keys():
    print("*"*10, algo, "*"*10)
    grid_search = GridSearchCV(estimator=pipelines[algo],
                               param_grid=param_grids[algo],
                               cv=5,
                               scoring='accuracy',
                               return_train_score=True,
                               verbose=1
                              )

    # Fit
    start_fit_time = time.time()
    grid_search.fit(x_train, y_train)
    end_fit_time = time.time()

    # Predict
    start_predict_time = time.time()
    y_pred = grid_search.predict(x_test)
    end_predict_time = time.time()

    # Saving the best model
    model_path = f'Best Models/{algo}.pkl'   #you can define your path and models
    joblib.dump(grid_search.best_estimator_, model_path)
    model_size = os.path.getsize(model_path)

    # Pring Log
    print('Train Score: ', grid_search.best_score_)
    print('Test Score: ', grid_search.score(x_test, y_test))
    print("Fit Time: ", end_fit_time - start_fit_time)
    print("Predict Time: ", end_predict_time - start_predict_time)
    print("Model Size: ", model_size)

    print()

    # Start the experiment run
    with mlflow.start_run() as run:
        # Log tags with mlflow.set_tag()
        mlflow.set_tag("developer", dev)

        # Log Parameters with mlflow.log_param()
        mlflow.log_param("algorithm", algo)
        mlflow.log_param("hyperparameter_grid", param_grids[algo])
        mlflow.log_param("best_hyperparameter", grid_search.best_params_)

        # Log Metrics with mlflow.log_metric()
        mlflow.log_metric("train_score", grid_search.best_score_)
        mlflow.log_metric("test_score", grid_search.score(x_test, y_test))
        mlflow.log_metric("fit_time", end_fit_time - start_fit_time)
        mlflow.log_metric("predict_time", end_predict_time - start_predict_time)
        mlflow.log_metric("model_size", model_size)
```

```
********** knn **********
Fitting 5 folds for each of 27 candidates, totalling 135 fits
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[23], line 28
     26 # Saving the best model
     27 model_path = f'Best Models/{algo}.pkl'   #you can define your path and models
---> 28 joblib.dump(grid_search.best_estimator_, model_path)
     29 model_size = os.path.getsize(model_path)
     31 # Pring Log

File ~\Desktop\a\Lib\site-packages\joblib\numpy_pickle.py:552, in dump(value, filename, compress, protocol, cache_size)
    550         NumpyPickler(f, protocol=protocol).dump(value)
    551 elif is_filename:
--> 552     with open(filename, 'wb') as f:
    553         NumpyPickler(f, protocol=protocol).dump(value)
    554 else:

FileNotFoundError: [Errno 2] No such file or directory: 'Best Models/knn.pkl'
```

In [ ]: