```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer,
TfidfVectorizer
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report, accuracy_score



d = pd.read_csv('/content/data.csv')



t = d['Sentence'].values
l = d['Sentiment'].values



tr_d, tt_d, tr_l, tt_l = train_test_split(t, l, test_size=0.2)



def evaluate_model(model, train_features, train_labels, test_features,
test_labels):
    model.fit(train_features, train_labels)
    predictions = model.predict(test_features)
    accuracy = accuracy_score(test_labels, predictions)
    print(classification_report(test_labels, predictions))
    print(f"Accuracy: {accuracy}")
    return accuracy


sample_sentences = ['This is an outstanding Movie. Don\'t dare to beat
Vijay Setupati. Once it was dared by Hrithik Roshan, not equivalent to
his nail.']



models = {
    'SVM': SVC(),
    'Naive Bayes': MultinomialNB(),
    'Decision Tree': DecisionTreeClassifier(),
    'Random Forest': RandomForestClassifier(),
    'MLP Classifier': MLPClassifier(max_iter=300)
}
```

```
vectorizers = {
    'CountVectorizer': CountVectorizer(),
    'TfidfVectorizer': TfidfVectorizer()
}

results = {}

for vec_name, vectorizer in vectorizers.items():
    print(f"\nUsing {vec_name}:\n")


    fe_tr = vectorizer.fit_transform(tr_d)


    fe_tt = vectorizer.transform(tt_d)

    results[vec_name] = {}

    for model_name, model in models.items():
        print(f"\n{model_name}:\n")
        accuracy = evaluate_model(model, fe_tr, tr_l, fe_tt, tt_l)
        results[vec_name][model_name] = accuracy


        nfe = vectorizer.transform(sample_sentences)
        nop = model.predict(nfe)
        print(f"Prediction for sample sentences: {nop}")


print("\nAccuracy Results:")
for vec_name in results:
    print(f"\n{vec_name}:")
    for model_name in results[vec_name]:
        print(f"{model_name}: {results[vec_name][model_name]}")
```

```
Using CountVectorizer:


SVM:

              precision    recall  f1-score   support

    negative       0.40      0.10      0.16       172
     neutral       0.68      0.90      0.77       624
    positive       0.72      0.59      0.65       373

    accuracy                           0.68      1169
   macro avg       0.60      0.53      0.53      1169
weighted avg       0.65      0.68      0.64      1169
```

```
Accuracy: 0.6809238665526091
Prediction for sample sentences: ['neutral']

Naive Bayes:

              precision    recall  f1-score   support

    negative       0.48      0.34      0.39       172
     neutral       0.72      0.84      0.78       624
    positive       0.72      0.61      0.66       373

    accuracy                           0.69      1169
   macro avg       0.64      0.60      0.61      1169
weighted avg       0.68      0.69      0.68      1169

Accuracy: 0.6937553464499572
Prediction for sample sentences: ['positive']

Decision Tree:

              precision    recall  f1-score   support

    negative       0.22      0.21      0.22       172
     neutral       0.66      0.64      0.65       624
    positive       0.60      0.65      0.63       373

    accuracy                           0.58      1169
   macro avg       0.50      0.50      0.50      1169
weighted avg       0.58      0.58      0.58      1169

Accuracy: 0.5808383233532934
Prediction for sample sentences: ['neutral']

Random Forest:

              precision    recall  f1-score   support

    negative       0.17      0.10      0.13       172
     neutral       0.66      0.83      0.73       624
    positive       0.75      0.58      0.65       373

    accuracy                           0.64      1169
   macro avg       0.53      0.50      0.50      1169
weighted avg       0.62      0.64      0.62      1169

Accuracy: 0.6398631308810949
Prediction for sample sentences: ['neutral']

MLP Classifier:
```

```
              precision    recall  f1-score   support

    negative       0.25      0.21      0.23       172
     neutral       0.70      0.76      0.73       624
    positive       0.75      0.69      0.72       373

    accuracy                           0.66      1169
   macro avg       0.57      0.55      0.56      1169
weighted avg       0.65      0.66      0.65      1169

Accuracy: 0.6578272027373824
Prediction for sample sentences: ['positive']

Using TfidfVectorizer:


SVM:

              precision    recall  f1-score   support

    negative       0.23      0.08      0.12       172
     neutral       0.68      0.88      0.77       624
    positive       0.78      0.62      0.69       373

    accuracy                           0.68      1169
   macro avg       0.56      0.53      0.53      1169
weighted avg       0.64      0.68      0.65      1169

Accuracy: 0.6809238665526091
Prediction for sample sentences: ['neutral']

Naive Bayes:

              precision    recall  f1-score   support

    negative       0.67      0.02      0.04       172
     neutral       0.63      0.97      0.76       624
    positive       0.71      0.37      0.49       373

    accuracy                           0.64      1169
   macro avg       0.67      0.46      0.43      1169
weighted avg       0.66      0.64      0.57      1169

Accuracy: 0.6415739948674081
Prediction for sample sentences: ['neutral']

Decision Tree:

              precision    recall  f1-score   support

    negative       0.21      0.24      0.23       172
```

```
     neutral       0.65      0.62      0.64       624
    positive       0.59      0.60      0.60       373

    accuracy                           0.56      1169
   macro avg       0.49      0.49      0.49      1169
weighted avg       0.57      0.56      0.56      1169

Accuracy: 0.5585970915312233
Prediction for sample sentences: ['neutral']

Random Forest:

              precision    recall  f1-score   support

    negative       0.17      0.09      0.12       172
     neutral       0.66      0.83      0.74       624
    positive       0.77      0.58      0.66       373

    accuracy                           0.64      1169
   macro avg       0.53      0.50      0.51      1169
weighted avg       0.62      0.64      0.62      1169

Accuracy: 0.6449957228400343
Prediction for sample sentences: ['neutral']

MLP Classifier:

              precision    recall  f1-score   support

    negative       0.25      0.20      0.23       172
     neutral       0.70      0.76      0.73       624
    positive       0.73      0.69      0.71       373

    accuracy                           0.65      1169
   macro avg       0.56      0.55      0.55      1169
weighted avg       0.64      0.65      0.65      1169

Accuracy: 0.6544054747647562
Prediction for sample sentences: ['positive']

Accuracy Results:

CountVectorizer:
SVM: 0.6809238665526091
Naive Bayes: 0.6937553464499572
Decision Tree: 0.5808383233532934
Random Forest: 0.6398631308810949
MLP Classifier: 0.6578272027373824

TfidfVectorizer:
SVM: 0.6809238665526091
```

```
Naive Bayes: 0.6415739948674081
Decision Tree: 0.5585970915312233
Random Forest: 0.6449957228400343
MLP Classifier: 0.6544054747647562
```

Accuracy Results:

CountVectorizer: SVM: 0.6809238665526091 Naive Bayes: 0.6937553464499572 Decision Tree: 0.5808383233532934 Random Forest: 0.6398631308810949 MLP Classifier: 0.6578272027373824

TfidfVectorizer: SVM: 0.6809238665526091 Naive Bayes: 0.6415739948674081 Decision Tree: 0.5585970915312233 Random Forest: 0.6449957228400343 MLP Classifier: 0.6544054747647562

BEST MODELS: CountVectorizer: Naive Bayes (0.6937553464499572) TfidfVectorizer: SVM (0.6809238665526091)