

```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split, cross_val_score
        from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.svm import SVC
        from sklearn.metrics import classification_report

        # Load the dataset
        df = pd.read_csv("islamic_lie_dataset.csv")
        df.shape
```

Out[1]: (100, 2)

```
In [2]: df_no_duplicates = df.drop_duplicates()
        df_no_duplicates.to_csv("islamic_lie_dataset_no_duplicates.csv", index=False)
        df_no_duplicates.shape
```

Out[2]: (20, 2)

```

In [3]: df = pd.read_csv("islamic_lie_dataset_no_duplicates.csv")
# Split the data into features (X) and labels (y)
X = df["Text"]
y = df["Label"]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a TF-IDF vectorizer
tfidf_vectorizer = TfidfVectorizer(max_features=1000, stop_words="english")

# Fit and transform the training data
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)

# Transform the testing data
X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Initialize an SVM classifier
svm_classifier = SVC(kernel="linear", random_state=42)

# Cross-validation to evaluate the model
cv_scores = cross_val_score(svm_classifier, X_train_tfidf, y_train, cv=5)

# Fit the model on the training data
svm_classifier.fit(X_train_tfidf, y_train)

# Make predictions on the test data
y_pred = svm_classifier.predict(X_test_tfidf)

# Evaluate the model
print("Cross-Validation Mean Accuracy:", cv_scores.mean())
print("Test Set Classification Report:")
print(classification_report(y_test, y_pred))

```

Cross-Validation Mean Accuracy: 0.18333333333333333

Test Set Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2.0
1	0.00	0.00	0.00	2.0
accuracy			0.00	4.0
macro avg	0.00	0.00	0.00	4.0
weighted avg	0.00	0.00	0.00	4.0

In []: