

1. Importing Libraries & Datasets

```
# Load All Libbus
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from wordcloud import WordCloud
import re
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix

# Load All Dss
dataframe_hinhi = pd.read_csv("hindi_data.csv")
dataframe_lassi = pd.read_csv("punjabi_lexicon.csv")
dataframe_dhokla = pd.read_csv("gujarati_data.csv")
dataframe_vadapav = pd.read_csv("final_marathi_data.csv")
```

2. Data Exploration

```
print("Hindi Language Dataset:-")
dataframe_hinhi

Hindi Language Dataset:-

{"summary":{"\n  \"name\": \"dataframe_hinhi\", \"rows\": 3641, \"\n\n  \"fields\": [\n    {\n      \"column\": \"Sentence\", \"\n\n      \"properties\": {\n        \"dtype\": \"string\", \"\n\n        \"num_unique_values\": 3641, \"\n\n        \"samples\": [\n          \"\n          u091f\\u0915\\u0930\\u093e\\u0935\", \"\n          \"\\u0938\\u0924\\u0930\\u094d\\u0915\\u0924\\u093e\", \"\n          \"damage\\n\n          \", \"\n          \"semantic_type\": \"\", \"\n\n          \"description\": \"\\n\n          }\n        }, \"\n        {\n          \"column\": \"Score\", \"\n          \"properties\": {\n            \"dtype\": \"number\", \"\n\n            \"std\": 0, \"\n            \"min\": -1, \"\n            \"max\": 1, \"\n\n            \"num_unique_values\": 2, \"\n\n            \"samples\": [\n              1, \"\n              -1\n            ], \"\n            \"semantic_type\": \"\", \"\n\n            \"description\": \"\\n\n            }\n          ]\n        }\n      ], \"type\": \"dataframe\", \"variable_name\": \"dataframe_hinhi\"}

dataframe_hinhi.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3641 entries, 0 to 3640
```

```
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  -
0    Sentence    3641 non-null    object
1    Score        3641 non-null    int64
dtypes: int64(1), object(1)
memory usage: 57.0+ KB
```

```
print("Punjabi Language Dataset:-")
dataframe_lassi
```

Punjabi Language Dataset:-

```
{
  "summary": {
    "name": "dataframe_lassi",
    "rows": 19010,
    "fields": [
      {
        "column": "Word",
        "properties": {
          "dtype": "string",
          "num_unique_values": 19010,
          "samples": [
            "\u0a18\u0a5c",
            "\u0a24\u0a32",
            "\u0a16\u0a71\u0a17"
          ],
          "semantic_type": ""
        },
        "description": "Positive Score",
        "properties": {
          "dtype": "number",
          "std": 0.18028028140498936,
          "min": 0.0,
          "max": 1.0,
          "num_unique_values": 388,
          "samples": [
            0.3881579,
            0.31944445,
            0.2625
          ],
          "semantic_type": ""
        },
        "description": "Negative Score",
        "properties": {
          "dtype": "number",
          "std": 0.20587854135614211,
          "min": 0.0,
          "max": 1.0,
          "num_unique_values": 409,
          "samples": [
            0.32692307,
            0.575,
            0.359375
          ],
          "semantic_type": ""
        }
      ]
    },
    "type": "dataframe",
    "variable_name": "dataframe_lassi"
  }
}
```

```
dataframe_lassi.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19010 entries, 0 to 19009
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0    Word        19010 non-null    object
1    Positive Score  19010 non-null    float64
2    Negative Score  19010 non-null    float64
dtypes: float64(2), object(1)
memory usage: 445.7+ KB
```

```
print("Marathi Language Dataset:-")
dataframe_vadapav
```

Marathi Language Dataset:-

```
{"summary":{"\n  \"name\": \"dataframe_vadapav\",\n  \"rows\": 15164,\n  \"fields\": [\n    {\n      \"column\": \"Sentence\",\n      \"properties\": {\n        \"dtype\": \"string\",\n        \"num_unique_values\": 15072,\n        \"samples\": [\n          \"\\u0926\\u093e\\u092d\\u094b\\u0932\\u0915\\u0930\\u092a\\u093e\\u0928\\u0938\\u0930\\u0947\\u0902\\u091a\\u0947\\u092e\\u093e\\u0930\\u0947\\u0915\\u0930\\u0940\\u0939\\u0940\\u092d\\u093e\\u091c\\u092a\\u091a\\u0947\\u0909\\u092e\\u0947\\u0926\\u0935\\u093e\\u0930\\u0926\\u093f\\u0938\\u0924\\u0940\\u0932\\u092e\\u093e\\u0932\\u0947\\u0917\\u093e\\u0935\\u092c\\u0949\\u092e\\u094d\\u092c\\u0938\\u094d\\u092b\\u094b\\u091f\\u093e\\u0924\\u0940\\u0932\\u0906\\u0930\\u094b\\u092a\\u0940\\u092a\\u094d\\u0930\\u091c\\u094d\\u091e\\u093e\\u0920\\u093e\\u0915\\u0942\\u0930\\u091a\\u094d\\u092f\\u093e\\u0909\\u092e\\u0947\\u0926\\u0935\\u093e\\u0930\\u0940\\u091a\\u0947\\u092d\\u093e\\u091c\\u092a\\u093e\\u0915\\u0921\\u0942\\u0928\\u0939\\u094b\\u0923\\u093e\\u0930\\u0947\\u0938\\u092e\\u0930\\u094d\\u0925\\u0928\\u092a\\u093e\\u0939\\u0924\\u093e\\u0928\\u0925\\u0941\\u0930\\u093e\\u092e\\u0917\\u094b\\u0921\\u0938\\u0947\\u091c\\u093f\\u0935\\u0902\\u0924\\u0905\\u0938\\u0924\\u093e\\u0924\\u0930\\u0924\\u094d\\u092f\\u093e\\u0932\\u093e\\u0939\\u0940\\u092d\\u093e\\u0930\\u0924\\u0940\\u092f\\u091c\\u0928\\u0924\\u093e\\u092a\\u0915\\u094d\\u0937\\u093e\\u0928\\u0947\\u0909\\u092e\\u0947\\u0926\\u0935\\u093e\\u0930\\u0940\\u0926\\u093f\\u0932\\u0940\\",\n          \"\\u0906\\u0926\\u0930\\u094d\\u0936\\u0930\\u093e\\u091c\\u094d\\u092f\\u0915\\u0930\\u094d\\u0924\\u094d\\u092f\\u093e\\u0930\\u093e\\u091c\\u092e\\u093e\\u0924\\u093e\\u092a\\u0941\\u0923\\u094d\\u092f\\u0936\\u094d\\u0932\\u094b\\u0915\\u0905\\u0939\\u093f\\u0932\\u094d\\u092f\\u093e\\u0926\\u0947\\u0935\\u0940\\u0939\\u094b\\u0933\\u0915\\u0930\\u092f\\u093e\\u0902\\u0928\\u093e\\u092a\\u0941\\u0923\\u094d\\u092f\\u0924\\u093f\\u0925\\u0940\\u0928\\u093f\\u092e\\u093f\\u0924\\u094d\\u0924\\u092d\\u093e\\u0935\\u092a\\u0942\\u0930\\u094d\\u0923\\u0906\\u0926\\u0930\\u093e\\u0902\\u091c\\u0932\\u0940\\u0924\\u094d\\u092f\\u093e\\u0902\\u0928\\u0940\\u0930\\u093e\\u091c\\u0915\\u093e\\u0930\\u0923\\u0938\\u092e\\u093e\\u091c\\u0915\\u093e\\u0930\\u0923\\u0927\\u0930\\u094d\\u092e\\u092a\\u0930\\u092a\\u0930\\u093e\\u092f\\u0923\\u0924\\u0947\\u091a\\u094d\\u092f\\u093e\\u0915\\u094d\\u0937\\u0947\\u0924\\u094d\\u0930\\u093e\\u0924\\u0906\\u0926\\u0930\\u094d\\u0936\\u0928\\u093f\\u0930\\u094d\\u092e\\u093e\\u0923\\u0915\\u0947\\u0932\\u093e\\u091c\\u093e\\u0924\\u0927\\u0930\\u094d\\u092e\\u092a\\u0902\\u0925\\u092a\\u094d\\u0930\\u093e\\u0902\\u0924\\u093e\\u0902\\u091a\\u094d\\u092f\\u093e\\u0938\\u0940\\u092e\\u093e\\u0913\\u0932\\u093e\\u0902\\u0921\\u0942\\u0928\\u0932\\u094b\\u0915\\u0915\\u0932\\u094d\\u092f\\u093e\\u0923\\u093e\\u091a\\u0940\\u0915\\u093e\\u092e\\u0902\\u0915\\u0947\\u0932\\u0940\\u092a\\u0930\\u0930\\u093e\\u091c\\u094d\\u092f\\u093e\\u0902\\u0936\\u0940\\u0938\\u0939\\u0915\\u093e\\u0930\\u094d\\u092f\\u093e\\u091a\\u0947\\u0938\\u0932\\u094b\\u0916\\u094d\\u092f\\u093e\\u091a\\u0947\\u0938\\u0902\\u092c\\u0902\\u0927\\u0930\\u093e\\u0916\\u0932\\
```

```

u0947\\",\\n          \\\"\\u0917 \\u0926\\u093f \\u092e\\u093e\\u0921\\u0917\\u0942\\u0933\\u0915\\u0930 \\u092f\\u093e\\u0902\\u091a\\u0947\\u0938\\u0941\\u092a\\u0941\\u0924\\u094d\\u0930 \\u0906\\u0923\\u093f \\u091c\\u094d\\u092f\\u0947\\u0937\\u094d\\u0920 \\u092a\\u0924\\u094d\\u0930\\u0915\\u093e\\u0930 \\u0936\\u094d\\u0930\\u0940\\u0927\\u0930 \\u092e\\u093e\\u0921\\u0917\\u0942\\u0933\\u0915\\u0930 \\u092f\\u093e\\u0902\\u091a\\u094d\\u092f\\u093e \\u0928\\u093f\\u0927\\u0928\\u093e\\u091a\\u0947 \\u0935\\u0943\\u0924\\u094d\\u0924 \\u0938\\u0902\\u092a\\u0942\\u0930\\u094d\\u0923\\u092e\\u0939\\u093e\\u0930\\u093e\\u0937\\u094d\\u091f\\u094d\\u0930\\u093e\\u0932\\u093e \\u091a\\u091f\\u0915\\u093e \\u0932\\u093e\\u0935\\u0923\\u093e\\u0930\\u0947 \\u0906\\u0939\\u0947 \\u092a\\u0924\\u094d\\u0930\\u0915\\u093e\\u0930 \\u0915\\u093e\\u0926\\u0902\\u092c\\u0930\\u0940\\u0915\\u093e\\u0930 \\u0932\\u0947\\u0916\\u0915 \\u0935\\u093f\\u091a\\u093e\\u0930\\u0935\\u0902\\u0924 \\u092f\\u093e\\u0936\\u093f\\u0935\\u093e\\u092f \\u0935\\u093f\\u0935\\u093f\\u0927 \\u0915\\u094d\\u0937\\u0947\\u0924\\u094d\\u0930\\u093e\\u0924\\u0940\\u0932 \\u0924\\u094d\\u092f\\u093e\\u0902\\u091a\\u0947 \\u0915\\u093e\\u0930\\u094d\\u092f\\u0905\\u0928\\u0947\\u0915\\u093e\\u0902\\u0938\\u093e\\u0920\\u0940\\u092a\\u094d\\u0930\\u0947\\u0930\\u0923\\u093e\\u0926\\u093e\\u092f\\u0940 \\u0906\\u0939\\u0947 \\u0924\\u094d\\u092f\\u093e\\u0902\\u0928\\u093e \\u092e\\u093e\\u091d\\u0940 \\u092d\\u093e\\u0935\\u092a\\u0942\\u0930\\u094d\\u0923 \\u0936\\u094d\\u0930\\u0926\\u094d\\u0927\\u093e\\u0902\\u091c\\u0932\\u0940\\\"\\n      ],\\n      \\\"semantic_type\\\": \\\"\\\",\\n      \\\"description\\\": \\\"\\\"\\n    }\\n  },\\n  {\\n    \\\"column\\\": \\\"Score\\\",\\n    \\\"properties\\\": {\\n      \\\"dtype\\\": \\\"number\\\",\\n      \\\"std\\\": 0,\\n      \\\"min\\\": -1,\\n      \\\"max\\\": 1,\\n      \\\"num_unique_values\\\": 3,\\n      \\\"samples\\\": [\\n        1,\\n        -1,\\n        0\\n      ],\\n      \\\"semantic_type\\\": \\\"\\\",\\n      \\\"description\\\": \\\"\\\"\\n    }\\n  }\\n ]\\n}\\n\\\",\\\"type\\\":\\\"dataframe\\\",\\\"variable_name\\\":\\\"dataframe_vadapav\\\"}

```

```
dataframe_vadapav.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15164 entries, 0 to 15163
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Sentence    15164 non-null  object
1   Score       15164 non-null  int64
dtypes: int64(1), object(1)
memory usage: 237.1+ KB

```

```

print("Gujrati Language Dataset:-")
dataframe_dhokla

```

```
Gujrati Language Dataset:-
```



```

# converting text dataframe into numerical features
X_hindi = tfidf_vectorizer.fit_transform(dataframe_hinhi['Sentence'])
X_punjabi = tfidf_vectorizer.transform(dataframe_lassi['Word'])
X_gujrati = tfidf_vectorizer.transform(dataframe_dhokla['Sentence'])
X_marathi = tfidf_vectorizer.transform(dataframe_vadapav['Sentence'])

# target variables
y_hindi = dataframe_hinhi['Score']
y_punjabi = (dataframe_lassi['Positive Score'] >
dataframe_lassi['Negative Score']).astype(int)
y_gujrati = dataframe_dhokla['Score']
y_marathi = dataframe_vadapav['Score']

# splitting in ts & ts
X_hindi_train, X_hindi_test, y_hindi_train, y_hindi_test =
train_test_split(X_hindi, y_hindi, test_size=0.2, random_state=42)
X_marathi_train, X_marathi_test, y_marathi_train, y_marathi_test =
train_test_split(X_marathi, y_marathi, test_size=0.2, random_state=42)
X_gujrati_train, X_gujrati_test, y_gujrati_train, y_gujrati_test =
train_test_split(X_gujrati, y_gujrati, test_size=0.2, random_state=42)
X_punjabi_train, X_punjabi_test, y_punjabi_train, y_punjabi_test =
train_test_split(X_punjabi, y_punjabi, test_size=0.2, random_state=42)

# train the Random Forest Classifier
model_hindi = RandomForestClassifier()
model_hindi.fit(X_hindi_train, y_hindi_train)

RandomForestClassifier()

model_marathi = RandomForestClassifier()
model_marathi.fit(X_marathi_train, y_marathi_train)

RandomForestClassifier()

model_gujrati = RandomForestClassifier()
model_gujrati.fit(X_gujrati_train, y_gujrati_train)

RandomForestClassifier()

model_punjabi = RandomForestClassifier()
model_punjabi.fit(X_punjabi_train, y_punjabi_train)

RandomForestClassifier()

# modal evalution on the test set
y_hindi_pred = model_hindi.predict(X_hindi_test)
accuracy_hindi = accuracy_score(y_hindi_test, y_hindi_pred)
print(f"Accuracy on the Hindi test set: {accuracy_hindi}")

Accuracy on the Hindi test set: 0.663923182441701

```

```

y_marathi_pred = model_marathi.predict(X_marathi_test)
accuracy_marathi = accuracy_score(y_marathi_test, y_marathi_pred)
print(f"Accuracy on the marathi test set: {accuracy_marathi}")

```

Accuracy on the marathi test set: 0.6660072535443455

```

y_gujrati_pred = model_gujrati.predict(X_gujrati_test)
accuracy_gujrati = accuracy_score(y_gujrati_test, y_gujrati_pred)
print(f"Accuracy on the gujrati test set: {accuracy_gujrati}")

```

Accuracy on the gujrati test set: 0.5536332179930796

```

y_punjabi_pred = model_punjabi.predict(X_punjabi_test)
accuracy_punjabi = accuracy_score(y_punjabi_test, y_punjabi_pred)
print(f"Accuracy on the punjabi test set: {accuracy_punjabi}")

```

Accuracy on the punjabi test set: 0.7185691741188848

generating classification report

```

class_report_hindi = classification_report(y_hindi_test, y_hindi_pred)
print("Classification Report for Hindi:")
print(class_report_hindi)

```

Classification Report for Hindi:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| -1 | 0.69 | 0.87 | 0.77 | 465 |
| 1 | 0.57 | 0.30 | 0.39 | 264 |
| accuracy | | | 0.66 | 729 |
| macro avg | 0.63 | 0.59 | 0.58 | 729 |
| weighted avg | 0.64 | 0.66 | 0.63 | 729 |

```

class_report_marathi = classification_report(y_marathi_test,
y_marathi_pred)
print("Classification Report for Marathi:")
print(class_report_marathi)

```

Classification Report for Marathi:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| -1 | 0.65 | 0.75 | 0.69 | 963 |
| 0 | 0.65 | 0.68 | 0.66 | 1066 |
| 1 | 0.71 | 0.58 | 0.64 | 1004 |
| accuracy | | | 0.67 | 3033 |
| macro avg | 0.67 | 0.67 | 0.66 | 3033 |
| weighted avg | 0.67 | 0.67 | 0.66 | 3033 |


```

class_report_punjabi = classification_report(y_punjabi_test,
y_punjabi_pred)
print("Classification Report for Punjabi:")
print(class_report_punjabi)

```

Classification Report for Punjabi:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.72 | 1.00 | 0.84 | 2732 |
| 1 | 0.00 | 0.00 | 0.00 | 1070 |
| accuracy | | | 0.72 | 3802 |
| macro avg | 0.36 | 0.50 | 0.42 | 3802 |
| weighted avg | 0.52 | 0.72 | 0.60 | 3802 |

```

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.

```

```

_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.

```

```

_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.

```

```

_warn_prf(average, modifier, msg_start, len(result))

```

```

class_report_gujrati = classification_report(y_gujrati_test,
y_gujrati_pred)
print("Classification Report for Gujarati:")
print(class_report_gujrati)

```

Classification Report for Gujarati:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.55 | 1.00 | 0.71 | 320 |
| 1 | 0.00 | 0.00 | 0.00 | 258 |
| accuracy | | | 0.55 | 578 |
| macro avg | 0.28 | 0.50 | 0.36 | 578 |
| weighted avg | 0.31 | 0.55 | 0.39 | 578 |

```

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in labels with no predicted

```



```

samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))

# generateing confusion matrix
conf_matrix_hindi = confusion_matrix(y_hindi_test, y_hindi_pred)
print("Confusion Matrix for Hindi:")
print(conf_matrix_hindi)

Confusion Matrix for Hindi:
[[405  60]
 [185  79]]

conf_matrix_hindi = confusion_matrix(y_hindi_test, y_hindi_pred)
print("Confusion Matrix for Hindi:")
print(conf_matrix_hindi)

Confusion Matrix for Hindi:
[[405  60]
 [185  79]]

conf_matrix_gujrati = confusion_matrix(y_gujrati_test, y_gujrati_pred)
print("Confusion Matrix for Gujrati:")
print(conf_matrix_gujrati)

Confusion Matrix for Gujrati:
[[320   0]
 [258   0]]

```

4. Sentiment Analysis

```

# Sentiment Analysis for Hindi
# thresholds for sentiment classification
positive_threshold = 0.6
negative_threshold = 0.4

def convert_to_sentiment_scores(predictions):
    """
    Convert model predictions to sentiment scores (positive, negative,
    neutral).

```

```

"""
sentiment_scores = []
for pred in predictions:
    if pred >= positive_threshold:
        sentiment_scores.append('Positive')
    elif pred <= negative_threshold:
        sentiment_scores.append('Negative')
    else:
        sentiment_scores.append('Neutral')
return sentiment_scores

def calculate_sentiment_distribution(sentiment_scores):
    """
    Calculate the distribution of sentiment labels.
    """
    unique_sentiments, counts = np.unique(sentiment_scores,
return_counts=True)
    return dict(zip(unique_sentiments, counts))

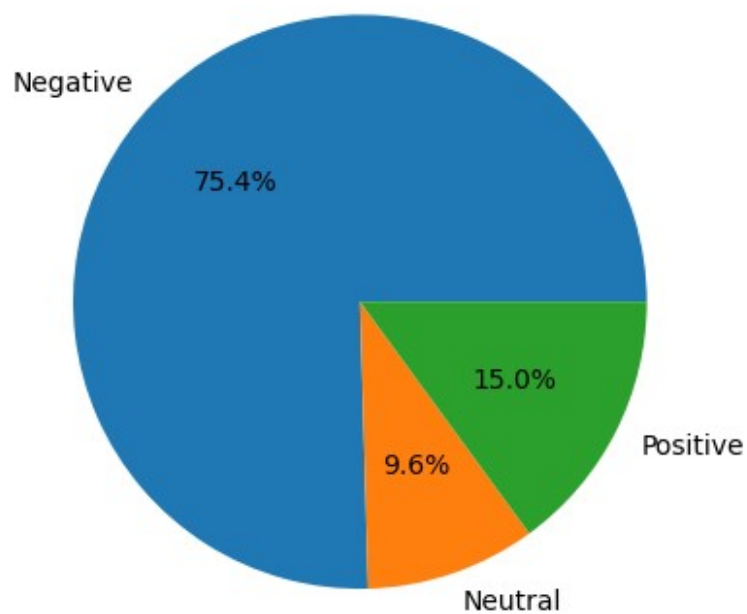
def plot_sentiment_distribution(sentiment_distribution):
    """
    Plot the distribution of sentiment labels.
    """
    labels, counts = zip(*sentiment_distribution.items())
    plt.pie(counts, labels=labels, autopct='%1.1f%%')
    plt.title('Sentiment Distribution')
    plt.show()

# sentiment labels for Hindi test data
y_hindi_pred = model_hindi.predict_proba(X_hindi_test)[: , 1]
sentiment_scores_hindi = convert_to_sentiment_scores(y_hindi_pred)

# pie chart for sentiment distribution
sentiment_distribution_hindi =
calculate_sentiment_distribution(sentiment_scores_hindi)
plot_sentiment_distribution(sentiment_distribution_hindi)

```

Sentiment Distribution



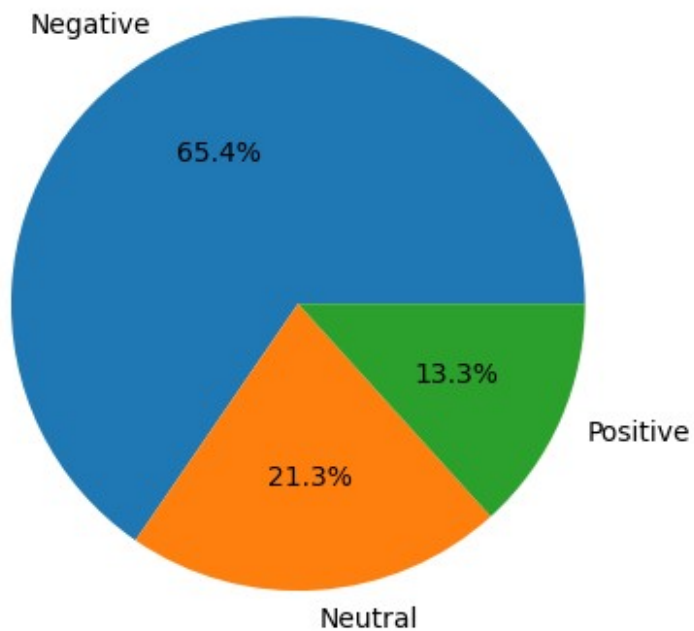
```
text_hindi = ' '.join(preprocessed_hindi_text)
font_path = '/content/Mangal Regular.ttf'
wordcloud_hindi = WordCloud(width=800, height=400,
background_color='white', font_path=font_path).generate(text_hindi)
plt.figure(figsize=(10, 8))
plt.imshow(wordcloud_hindi, interpolation='bilinear')
plt.title('Word Cloud for Hindi')
plt.axis('off')
plt.show()
```

A vibrant word cloud composed of numerous Hindi words in various shades of green, yellow, and orange. The words are arranged in a dense, overlapping manner, with some appearing much larger than others. Recognizable words include "परम", "सम", "हल", "अन", "चक्र", "शर", "धन", "भक्त", "कर्म", "जगत्", "पद्म", "वन्द्य", "प्रिय", "स्नेह", "आशा", "विश्वास", "दया", "करुणा", "मित्रता", "साहस", "वीर्य", "तप", "ब्रह्म", "मोक्ष", "निर्वाण", "सर्वज्ञ", "सर्वशक्तिमान्". The background features stylized green leaves and branches, giving it a fresh, organic feel.

```
y_marathi_pred = model_marathi.predict_proba(X_marathi_test)[: , 1]
sentiment_scores_marathi = convert_to_sentiment_scores(y_marathi_pred)
```

```
sentiment_distribution_marathi =  
calculate_sentiment_distribution(sentiment_scores_marathi)  
plot_sentiment_distribution(sentiment_distribution_marathi)
```

Sentiment Distribution



```
dataframe_marathi = pd.read_csv("/content/final_marathi_data.csv")
marathi_text = ' '.join(dataframe_marathi['Sentence'])
wordcloud_marathi = WordCloud(width=800, height=400,
background_color='white', font_path='/content/Mangal
Regular.ttf').generate(marathi_text)
plt.figure(figsize=(10, 8))
plt.imshow(wordcloud_marathi, interpolation='bilinear')
plt.title('Word Cloud for Marathi')
plt.axis('off')
plt.show()
```

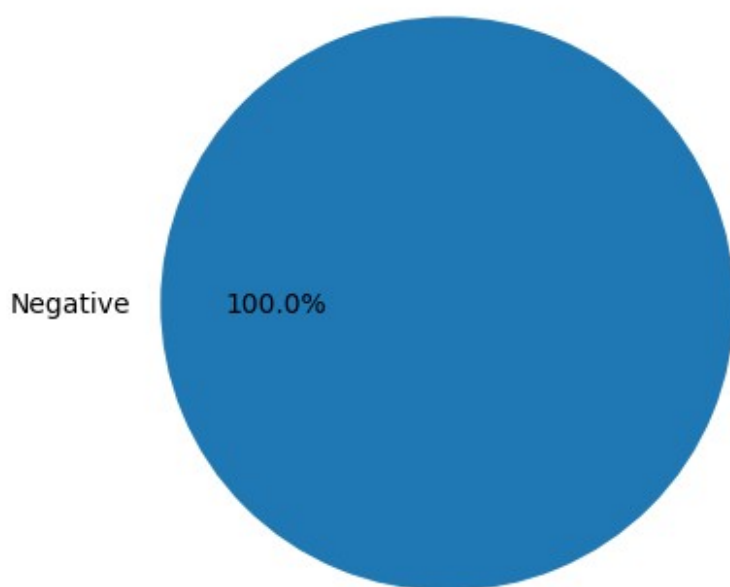
Word Cloud for Marathi



```
# sentiment labels for Punjabi test data
y_punjabi_pred = model_punjabi.predict_proba(X_punjabi_test)[: , 1]
sentiment_scores_punjabi = convert_to_sentiment_scores(y_punjabi_pred)

# pie chart for sentiment distribution in punjabi
sentiment_distribution_punjabi =
calculate_sentiment_distribution(sentiment_scores_punjabi)
plot_sentiment_distribution(sentiment_distribution_punjabi)
```

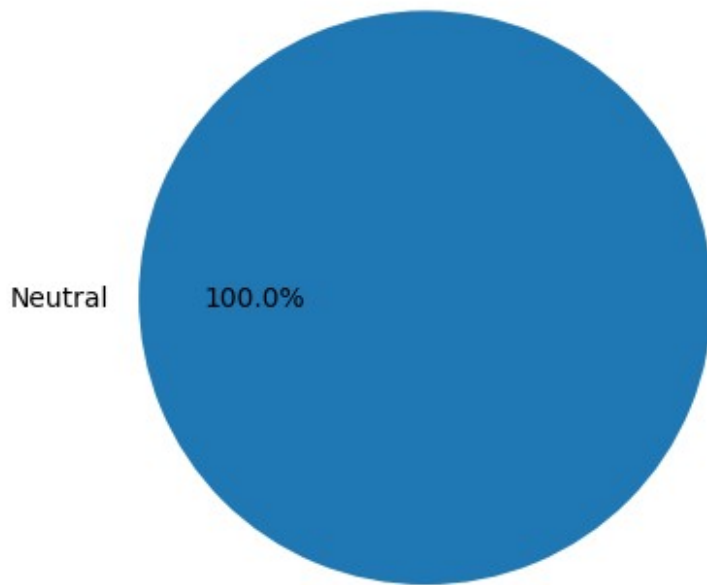
Sentiment Distribution



```
# sentiment labels for gujarati test data
y_gujarati_pred = model_gujrati.predict_proba(X_gujrati_test)[: , 1]
sentiment_scores_gujarati =
convert_to_sentiment_scores(y_gujarati_pred)

sentiment_distribution_gujarati =
calculate_sentiment_distribution(sentiment_scores_gujarati)
plot_sentiment_distribution(sentiment_distribution_gujarati)
```


Sentiment Distribution



```
dataframe_gujarati = pd.read_csv("/content/gujarati_data.csv")
gujarati_text = ' '.join(dataframe_gujarati['Sentence'])
wordcloud_gujarati = WordCloud(width=800, height=400,
background_color='white',
font_path='/content/Aakar.ttf').generate(gujarati_text)
plt.figure(figsize=(10, 8))
plt.imshow(wordcloud_gujarati, interpolation='bilinear')
plt.title('Word Cloud for Gujarati')
plt.axis('off')
plt.show()
```

Word Cloud for Gujarati

