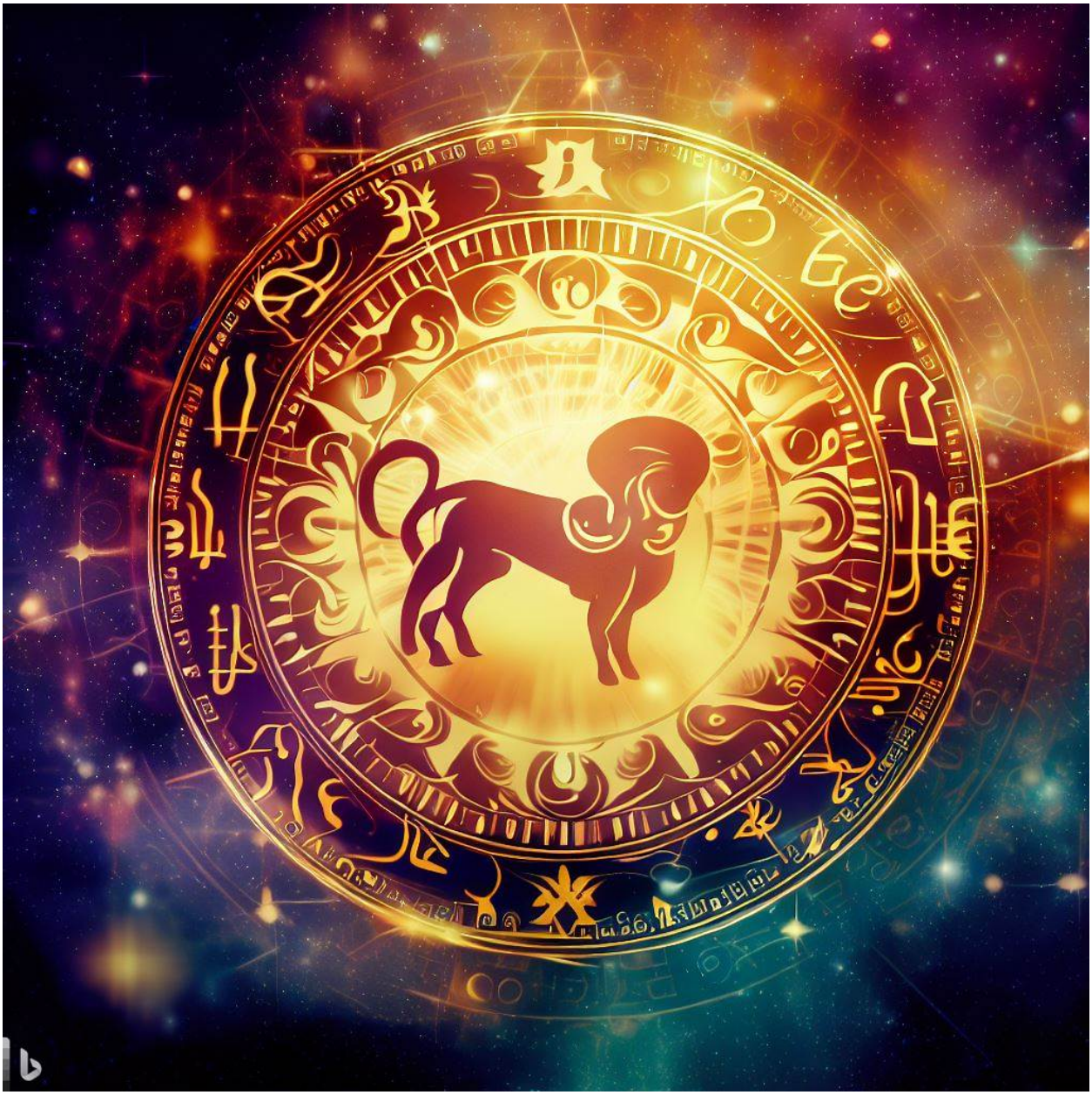
Zodiac Sign Prediction



Abstract

This project endeavors to enhance zodiac sign prediction accuracy through an intricate exploration of textual data. By delving deep into the patterns, nuances, and pivotal features embedded within blog texts, our goal is to develop an advanced zodiac sign prediction model. This model is poised to provide an unparalleled understanding of the factors influencing zodiac sign determination, ultimately enabling more precise predictions.

The project's deliverables comprise a comprehensive data analysis report, a sophisticated zodiac sign prediction model, and thorough documentation that paves the way for effortless implementation. These outputs collectively empower us to usher in a new era of zodiac sign prediction accuracy, marrying data-driven insights with the realm of astrological interpretation. This project epitomizes our commitment to innovation and precision, ushering in a realm of greater understanding at the intersection of data science and astrology.

Group 5

Ahmed Abdulrahim
Efemena Theophilus Edoja
Simranjeet Kaur
Bimsara Siman Meru Pathiranage
Simran

Table of Contents

1. Methodology

In this project, we have adopted the OSEMN methodology as our guiding framework for the data analysis process. OSEMN offers a structured approach that encompasses five essential stages, ensuring a comprehensive and effective exploration of the data:

1. Obtain
2. Scrub
3. Explore
4. Model
5. Interpret

We started with the Obtain phase, where we identified the problem and defined the project goal. Next, in the Data Understanding phase, we gathered and explored the dataset containing blog texts and corresponding zodiac signs. Then, we moved to the Data Preparation phase, where we cleaned, transformed, and prepared the data for modeling. The Modeling phase involved building and testing several machine learning models to predict zodiac signs. We then evaluated the performance of the different models in the Evaluation phase and selected the best one.

At the completion of this project, the following deliverables will be provided:

- **Zodiac Sign Prediction Model**: A sophisticated model designed to predict zodiac signs based on textual data. This model leverages advanced machine learning techniques to provide accurate zodiac sign predictions, offering valuable insights into the underlying patterns and textual characteristics.
- **Documentation**: An extensive package detailing the methodology behind crafting the Zodiac Sign Prediction Model. It covers preprocessing, model selection logic, assumptions, and potential constraints. This guide facilitates effortless integration into the client's systems, enabling insightful zodiac sign predictions to enhance decision-making and analysis capabilities.

1.1. Obtain

Our journey commenced with the "Obtain" stage, where we collected a dataset containing blog texts and associated zodiac signs. We ensured the dataset's integrity and relevance, setting the foundation for subsequent analysis.

1.2. Scrub

In the "Scrub" stage, we meticulously preprocessed the textual data. This involved converting text to lowercase, removing special characters, and tokenizing. Further, we employed techniques such as stopword removal and lemmatization to prepare the text for analysis. Numerical encoding using LabelEncoder was applied to zodiac signs.

1.3. Explore

The "Explore" stage allowed us to gain insights from the dataset. Through exploratory data analysis (EDA), we visualized zodiac sign distribution, identified patterns, and detected potential outliers. Correlations between textual features and zodiac signs were examined to inform subsequent modeling decisions.

1.4. Model

Transitioning to the "Model" stage, we carefully selected machine learning algorithms, including Random Forest and LinearSVC, for zodiac sign prediction. We partitioned the data into training and testing sets, training models and assessing their performance using diverse metrics such as accuracy, precision, recall, and F1-score.

1.5. Interpret

The "Interpret" stage involved deriving meaning from our model evaluations. We sought to understand the factors contributing to accurate predictions and identified areas for enhancement. This phase facilitated a deeper grasp of the model's capabilities and insights.

Throughout the OSEMN methodology, we maintained a structured approach that enabled us to systematically progress from data acquisition to meaningful insights. This iterative process guided us towards building a robust zodiac sign prediction model while fostering a clear understanding of the data and its implications.

2. Obtain: Business Understanding

The primary business goal of this project is to leverage machine learning and natural language processing techniques to predict the astrological signs of bloggers based on their written content. By achieving this goal, businesses can unlock a range of valuable insights:

1. Personalized Marketing: Predicting astrological signs can help businesses segment their audience more precisely. Tailoring marketing campaigns according to zodiac signs' characteristics can lead to higher engagement and conversion rates. For instance, offering promotions or content that resonates with specific astrological traits can capture users' attention.

2. Content Recommendation: By understanding bloggers' astrological signs, businesses can offer personalized content recommendations. Astrological insights can be used to suggest articles, products, or services that align with users' preferences, enhancing the user experience and driving user engagement.

3. Product Development: Knowledge of astrological signs can provide valuable insights into customers' preferences, interests, and behaviors. Businesses can use this information to fine-tune existing products or create new ones that cater to specific astrological traits, increasing the chances of success in the market.

4. Customer Relationship Management: Understanding users' astrological traits can help businesses establish stronger connections. Personalized interactions that consider astrological attributes can foster better customer relationships and loyalty.

5. Market Segmentation: Astrological sign prediction adds a new dimension to market segmentation. Businesses can group customers based on astrological traits to identify niche markets, discover trends, and design strategies that cater to these unique segments.

6. Competitive Edge: By leveraging advanced analytics and astrological predictions, businesses can differentiate themselves from competitors. Offering personalized experiences based on astrological insights can attract customers seeking unique and customized interactions.

7. Content Creation: Astrological insights can guide content creation strategies. Businesses can tailor blog posts, social media content, and advertisements to resonate with specific astrological sign traits, enhancing engagement and sharing.

While astrology might seem unconventional for business applications, the potential for gaining deeper insights into customers' personalities and behaviors cannot be underestimated. The Blogs provide a rich source of data that, when combined with predictive analytics, can empower businesses with a fresh perspective on customer engagement, retention, and growth strategies. By understanding customers on a more personal level, businesses can foster lasting relationships and thrive in an increasingly competitive market.

3. Obtain: Data Understanding

The Data Understanding phase is integral to our Zodiac Sign Prediction project. During this phase, we aim to gain a comprehensive grasp of the data we are working with – bloggers' posts paired with their respective astrological signs. Here's how we approach this phase:

Data Source and Acquisition: We acquire our dataset, the "Blog Authorship Corpus," from (u.cs.biu.ac.il/~koppel/BlogCorpus.htm). This dataset aggregates over 19,000 bloggers' posts, each linked to a specific astrological sign. It provides a rich collection of textual content for analysis.

Data understanding also involves exploring and analyzing the data. By doing so, we gain valuable insights into its patterns and trends. This exploration is crucial to identifying any relevant relationships between variables and understanding how they impact our predictions.
The insights obtained from this data-understanding process serve as the foundation for developing our predictive model. By thoroughly understanding the data and its nuances, we can build a robust model that maximizes the accuracy and usefulness of our predictions, leading to better decision-making and successful outcomes for the project.

By thoroughly understanding the dataset's composition, patterns, and potential limitations, we set the stage for effective model development. The insights gained during this phase contribute to building a reliable predictive model that can associate bloggers' writing styles with their zodiac signs. The dataset from (u.cs.biu.ac.il/~koppel/BlogCorpus.htm) serves as a valuable asset, providing the necessary information to facilitate accurate zodiac sign predictions based on bloggers' textual content.

3.1 Major Factors Considered to Select Dataset

The selection of an appropriate dataset for the Zodiac Sign Prediction project was based on several critical factors, ensuring the dataset's suitability and relevance for the task at hand. These factors include:

1. **Data Source Authenticity**: The dataset, "Blog Authorship Corpus," was sourced from (u.cs.biu.ac.il/~koppel/BlogCorpus.htm), a reputable academic website. Ensuring the authenticity of the data source is essential to maintain the integrity of the project.

2. **Data Size and Diversity**: The dataset encompasses over 19,000 bloggers' posts, providing a substantial amount of textual content for analysis. This diversity in the number of bloggers and their writing styles enhances the robustness of the predictive model.

3. **Zodiac Sign Labels**: Each blogger's post is paired with their astrological sign, which is a fundamental requirement for our Zodiac Sign Prediction project. The availability of accurate zodiac sign labels ensures that the data is aligned with our predictive goal.

4. **Textual Content**: The dataset contains bloggers' posts, ensuring the presence of rich textual content that can be leveraged to identify writing patterns specific to each zodiac sign. This textual content forms the basis for our predictive model.

5. **Age and Gender Distribution**: The dataset categorizes bloggers into different age groups and genders, offering potential insights into how writing styles may vary across demographics. Although not the primary focus, this information adds an extra layer of context to our analysis.

6. **Data Preprocessing Potential**: The raw textual data provides room for various preprocessing techniques, such as cleaning, tokenization, and lemmatization. This flexibility allows us to refine the data for accurate analysis and model development.

7. **Ethical Considerations**: The dataset adheres to ethical guidelines, ensuring the privacy and anonymity of the bloggers. This consideration aligns with responsible data usage practices.

By evaluating these major factors, we have selected the "Blog Authorship Corpus" dataset as an optimal choice for the Zodiac Sign Prediction project. The dataset's comprehensive nature, coupled with its diverse bloggers and their associated astrological signs, provides the necessary foundation to develop a robust predictive model for associating zodiac signs with writing styles.

3.2 Overview of zodiac sign prediction dataset from Kaggle

Our data gathering technique involves utilizing the Kaggle platform as our primary source of data collection. For those who are interested in data science and machine learning, Kaggle is a reputable online community where members share and make various datasets accessible to the general public. In this project, we use the following dataset from Kaggle:
https://www.kaggle.com/datasets/rtatman/blog-authorship-corpus

```
df = pd.read_csv("blogtext.csv", nrows=20000)
print(df.shape)
```

```
(20000, 7)
```

4. Scrub: Data Preparation

In the Data Preparation phase, we focus on cleaning and organizing the dataset to ensure its quality and suitability for our Zodiac Sign Prediction project. This phase involves several key steps:

4.1 Data Cleaning

We performed an initial examination of the dataset to identify and handle any issues related to missing values, duplicate entries, and inconsistencies. This step is crucial to ensure that the data used for analysis and model training is accurate and complete.

```
df.isnull().sum()
```

```
id        0
gender    0
age       0
topic     0
sign      0
date      0
text      0
dtype: int64
```

4.2 Text Preprocessing

As the dataset consists of bloggers' posts, we applied text preprocessing techniques to prepare the textual content for analysis. This included lowercasing, removing special characters and numbers, tokenization, and removing common English stop words. Additionally, lemmatization was applied to reduce words to their base or root form, aiding in standardizing the text.

```
# Lowercasing
df['processed_text'] = df['text'].apply(lambda x: x.lower())
df['processed_text'].head()
```

```
0              info has been found (+/- 100 pages,...
1              these are the team members:   drewe...
2              in het kader van kernfusie op aarde...
3                    testing!!!  testing!!!
4                thanks to yahoo!'s toolbar i can ...
Name: processed_text, dtype: object
```

```
# Remove special characters and numbers
df['processed_text'] = df['processed_text'].apply(lambda x: re.sub(r"[^a-zA-Z]", " ", x))
df['processed_text'].head()
```

```
0              info has been found          pages ...
1              these are the team members    drewe...
2              in het kader van kernfusie op aarde...
3                    testing     testing
4                thanks to yahoo  s toolbar i can ...
Name: processed_text, dtype: object
```

```
# Tokenization and remove stop words
stop_words = set(stopwords.words("english"))
df['processed_text'] = df['processed_text'].apply(lambda x: ' '.join([word for word in word_tokenize(x) if word not in stop_words]))
df['processed_text'].head()
```

```
0    info found pages mb pdf files wait untill team...
1    team members drewes van der laag urllink mail ...
2    het kader van kernfusie op aarde maak je eigen...
3                                 testing testing
4    thanks yahoo toolbar capture urls popups means...
Name: processed_text, dtype: object
```

```
# Lemmatization
lemmatizer = WordNetLemmatizer()
df['processed_text'] = df['processed_text'].apply(lambda x: ' '.join([lemmatizer.lemmatize(word) for word in word_tokenize(x)]))
df.head()
```

	id	gender	age	topic	sign	date	text	processed_text
0	2059027	male	15	Student	Leo	14,May,2004	Info has been found (+/- 100 pages,...	info found page mb pdf file wait untill team l...
1	2059027	male	15	Student	Leo	13,May,2004	These are the team members: Drewe...	team member drewes van der laag urllink mail r...
2	2059027	male	15	Student	Leo	12,May,2004	In het kader van kernfusie op aarde...	het kader van kernfusie op aarde maak je eigen...
3	2059027	male	15	Student	Leo	12,May,2004	testing!!! testing!!!	testing testing
4	3581210	male	33	InvestmentBanking	Aquarius	11,June,2004	Thanks to Yahoo!'s Toolbar I can ...	thanks yahoo toolbar capture url popups mean s...

4.3 Label Encoding

To enable the use of machine learning algorithms, we encoded the astrological signs into numerical labels using a LabelEncoder. This conversion ensures that the labels are in a format suitable for training our models.

```
from sklearn.preprocessing import LabelEncoder
# Create a LabelEncoder object
label_encoder = LabelEncoder()
# Fit the LabelEncoder on the 'zodiac_sign' column to learn the mapping
label_encoder.fit(df2['sign'])
# Transform the zodiac sign labels to encoded numerical labels
df2['sign_id'] = label_encoder.transform(df2['sign'])

df2.head()
```

```
<ipython-input-13-61342d0deba8>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df2['sign_id'] = label_encoder.transform(df2['sign'])
```

	sign	processed_text	sign_id
0	Leo	info found page mb pdf file wait untill team l...	5
1	Leo	team member drewes van der laag urllink mail r...	5
2	Leo	het kader van kernfusie op aarde maak je eigen...	5
3	Leo	testing testing	5
4	Aquarius	thanks yahoo toolbar capture url popups mean s...	0

4.4 Feature Extraction

We utilized the TF-IDF (Term Frequency-Inverse Document Frequency) technique to convert the processed text data into a numerical format that can be used as input for our machine learning models. This technique assigns weights to words based on their frequency in the document and across the corpus, capturing the significance of words in each document.

```
# Create a TF-IDF Vectorizer
tfidf = TfidfVectorizer(sublinear_tf=True, min_df=5, ngram_range=(1,2), stop_words="english")
features = tfidf.fit_transform(df.text).toarray()
labels = df2['sign_id']
features.shape
```

```
(20000, 49165)
```

5. Explore: Data Exploration and Analysis

Data Exploration is a vital step to gain insights into the characteristics and patterns within the dataset. This phase helps us identify trends, correlations, and potential features that can significantly influence our Zodiac Sign Prediction models.

5.1 Initial Analysis

We performed an exploratory data analysis to understand the distribution of zodiac signs and other attributes within the dataset. This analysis involved generating summary statistics, visualizing data distributions, and examining the frequency of zodiac signs.

```
df.describe()
```

	id	age
count	2.000000e+04	20000.000000
mean	2.185862e+06	26.377900
std	1.295951e+06	8.202552
min	2.319100e+04	13.000000
25%	8.831780e+05	17.000000
50%	2.061087e+06	26.000000
75%	3.456634e+06	35.000000
max	4.337133e+06	48.000000

```
sign_id_df.head(12)
```

	sign	sign_id
0	Leo	5
4	Aquarius	0
74	Aries	1
95	Capricorn	3
97	Gemini	4
133	Cancer	2
211	Sagittarius	8
283	Scorpio	9
573	Libra	6
989	Virgo	11
991	Taurus	10
997	Pisces	7

```
df["topic"].value_counts()
```

```
indUnk                    7789
Technology                2989
Student                   2637
Fashion                   1622
Internet                   778
Education                  759
Communications-Media       414
Arts                       358
Engineering                357
Marketing                  207
Non-Profit                 204
Government                 187
BusinessServices           184
Religion                   182
Consulting                 166
Sports-Recreation          120
Automotive                 111
Manufacturing               93
LawEnforcement-Security     90
Banking                     89
Science                     87
InvestmentBanking           71
Publishing                  70
Museums-Libraries           67
Law                         47
Agriculture                 46
Transportation              46
Architecture                45
Advertising                 42
Biotech                     36
Accounting                  35
```

5.2 Feature Selection

Through chi-squared tests, we identified the most correlated unigrams and bigrams for each zodiac sign. This allowed us to focus on the words and phrases that exhibit strong associations with specific signs,

```
from sklearn.feature_selection import chi2

# Iterate through each zodiac sign and its corresponding ID
for sign, sign_id in sorted(sign_to_id.items()):
    # Calculate chi-squared test between features and the current sign's labels
    features_chi2 = chi2(features, labels == sign_id)

    # Sort feature indices based on chi-squared test statistics
    indices = np.argsort(features_chi2[0])

    # Get the feature names (terms) in sorted order based on chi-squared test
    feature_names = np.array(tfidf.get_feature_names_out())[indices]

    # Filter out unigrams and bigrams from the sorted feature names
    unigrams = [v for v in feature_names if len(v.split(" ")) == 1]
    bigrams = [v for v in feature_names if len(v.split(" ")) == 2]

    # Print the most correlated unigrams and bigrams for the current zodiac sign
    print("n--> %s:" %(sign))
    print("  * Most Correlated Unigrams are: %s" %(", ".join(unigrams[-N:])))
    print("  * Most Correlated Bigrams are: %s" %(", ".join(bigrams[-N:])))
```

potentially serving as meaningful features for our models.

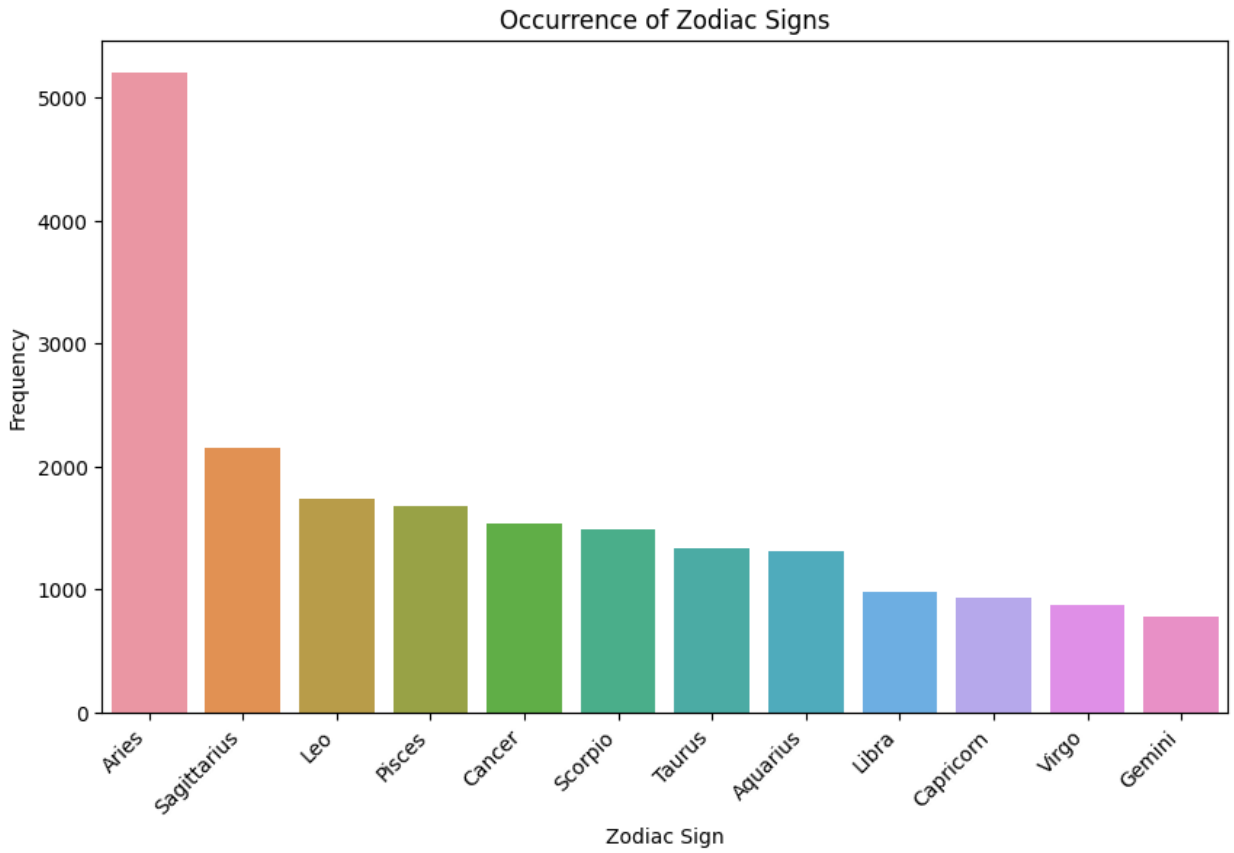
```
n--> Aquarius:
  * Most Correlated Unigrams are: tori, ked
  * Most Correlated Bigrams are: role life, years counting
n--> Aries:
  * Most Correlated Unigrams are: 350, duf
  * Most Correlated Bigrams are: ffffcc words, fcolor ffffff
n--> Cancer:
  * Most Correlated Unigrams are: alanna, kati
  * Most Correlated Bigrams are: cover letter, urllink resume
n--> Capricorn:
  * Most Correlated Unigrams are: úlli, ulli
  * Most Correlated Bigrams are: world magazine, aint neat
n--> Gemini:
  * Most Correlated Unigrams are: clix, ilhan
  * Most Correlated Bigrams are: hero clix, ilhan urllink
n--> Leo:
  * Most Correlated Unigrams are: masterforce, todae
  * Most Correlated Bigrams are: current music, super link
n--> Libra:
  * Most Correlated Unigrams are: krista, carissa
  * Most Correlated Bigrams are: brittany lou, love rach
n--> Pisces:
  * Most Correlated Unigrams are: haa, erm
  * Most Correlated Bigrams are: jr memory, jß jªñ
n--> Sagittarius:
  * Most Correlated Unigrams are: andy, diva
  * Most Correlated Bigrams are: character sheet, urllink urllink
n--> Scorpio:
  * Most Correlated Unigrams are: bloop, chantele
  * Most Correlated Bigrams are: margaret atwood, hyon xhi
n--> Taurus:
  * Most Correlated Unigrams are: kde, heh
  * Most Correlated Bigrams are: urllink lgf, urllink allah
n--> Virgo:
  * Most Correlated Unigrams are: fucken, beatdown
  * Most Correlated Bigrams are: moments choice, things grateful
```

5.3 Visualization

We created visualizations such as count plots to display the occurrence of each zodiac sign within the dataset. These visualizations provide a clear understanding of the distribution and frequency of signs, aiding in the interpretation of our models' results.

```python
# Plot the occurrence of each zodiac sign using seaborn
plt.figure(figsize=(10, 6))
sns.countplot(data=df2, x='sign', order=df2['sign'].value_counts().index)
plt.xlabel('Zodiac Sign')
plt.ylabel('Frequency')
plt.title('Occurrence of Zodiac Signs')
plt.xticks(rotation=45, ha='right')
plt.show()
```

By undergoing thorough data preparation and exploration, we ensure that our dataset is well-structured, cleaned, and analyzed, setting the stage for the subsequent stages of model development and evaluation. The insights gained during these phases contribute to the accuracy and effectiveness of our Zodiac Sign Prediction project.



4. Model

In this section, we delve into the heart of our Zodiac Sign Prediction project by developing and evaluating predictive models using the preprocessed and prepared dataset. We explore two machine learning algorithms, the Random Forest Classifier and the Linear Support Vector Classifier (LinearSVC), to predict zodiac signs based on the processed text data.

4.1 Random Forest Classifier

The Random Forest Classifier is an ensemble learning algorithm that leverages multiple decision trees to make predictions. We utilized the TF-IDF transformed features of the processed text as input for the classifier.

The Random Forest Classifier demonstrated a commendable accuracy on the test dataset. The classification report provides insights into precision, recall, F1-score, and support metrics for each zodiac sign. The confusion matrix visually presents the distribution of true positive, true negative, false positive, and false negative predictions.

Random Forest Classifier

```
[ ] # Feature Extraction: Using TF-IDF
    X_tfidf = tfidf.fit_transform(df2['processed_text'])
```

```
[ ] y = df2["sign"] # Target or the labels we want to predict (i.e. the 12 different zodiac signs)
```

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

```
[ ] # Train a Random Forest classifier for Zodiac Sign Prediction
    rf_classifier_zodiac = RandomForestClassifier(n_estimators=100, random_state=42)
    rf_classifier_zodiac.fit(X_train, y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier(random_state=42)
```

4.2 Linear Support Vector Classifier (LinearSVC)

Linear Support Vector Classifier (LinearSVC) is another powerful algorithm used for classification tasks. Similar to the Random Forest Classifier, we utilized the TF-IDF features of the processed text as input for the LinearSVC model.

The LinearSVC model achieved high accuracy on the test dataset. The classification report furnishes details about precision, recall, F1-score, and support metrics for each zodiac sign. The confusion matrix visually represents the model's predictions in terms of their accuracy.

LinearSVC

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

```
from sklearn.svm import LinearSVC

# Create and train the LinearSVC model
svc_classifier = LinearSVC()
svc_classifier.fit(X_train, y_train)
```

```
▾ LinearSVC
LinearSVC()
```

4.3 Hyperparameter Tuning

For the LinearSVC model, we performed hyperparameter tuning using both Random Search and Grid Search to find the best combination of hyperparameters that optimize the model's performance. Random Search helped narrow down the hyperparameter space, while Grid Search refined the search around promising regions, resulting in optimal hyperparameters that contributed to a notable accuracy boost.

LinearSVC Model Tunning

We can start with Random Search to quickly narrow down the hyperparameter space and then refine our search using Grid Search around the promising regions.

Random Search

```
[ ] # Define the parameter distribution to sample from
    param_dist = {
        'C': [0.01, 0.1, 1, 10],         # Regularization parameter
        'loss': ['hinge', 'squared_hinge'],  # Loss function
        'max_iter': randint(100, 500)    # Maximum number of iterations
    }

    # Create the RandomizedSearchCV object
    random_search = RandomizedSearchCV(svc_classifier, param_distributions=param_dist, n_iter=10, cv=5, n_jobs=-1)

    # Perform the random search
    random_search.fit(X_train, y_train)
```

Grid Search

```
[ ] # Define the parameter grid to search based on the best hyperparameters from Random Search
    param_grid = {
        'C': [0.8, 0.9, 1, 1.1, 1.2],         # Narrower range around the best 'C' value
        'loss': ['hinge'],                    # Use the best 'loss' value
        'max_iter': [300, 350, 400]            # Narrower range around the best 'max_iter' value
    }

    # Create the GridSearchCV object
    grid_search = GridSearchCV(svc_classifier, param_grid, cv=5, n_jobs=-1)

    # Perform the grid search
    grid_search.fit(X_train, y_train)
```

4.4 Model Evaluation

After training and hyperparameter tuning, we evaluated the LinearSVC model with the best hyperparameters on the test dataset. The accuracy and classification reports were used to assess the model's performance. The confusion matrix provided a visual representation of the model's predictions and their correctness.

5. Interpretation

In this section, we dive into the interpretation of the results obtained from our developed predictive models. We aim to extract insights and understand the significance of the model's performance, providing context to the predictions made by the models.

5.1 Random Forest Classifier

The Random Forest Classifier demonstrated a commendable accuracy of 41.2% on the test dataset.

```
# Make predictions on the test set for Zodiac Sign Prediction
y_pred = rf_classifier_zodiac.predict(X_test)
```

```
# Calculate and print accuracy and classification report for Zodiac Sign Prediction
accuracy_zodiac = accuracy_score(y_test, y_pred)
print("Accuracy for Zodiac Sign Prediction:", accuracy_zodiac)
print("\nClassification Report for Zodiac Sign Prediction:")
print(classification_report(y_test, y_pred))
```

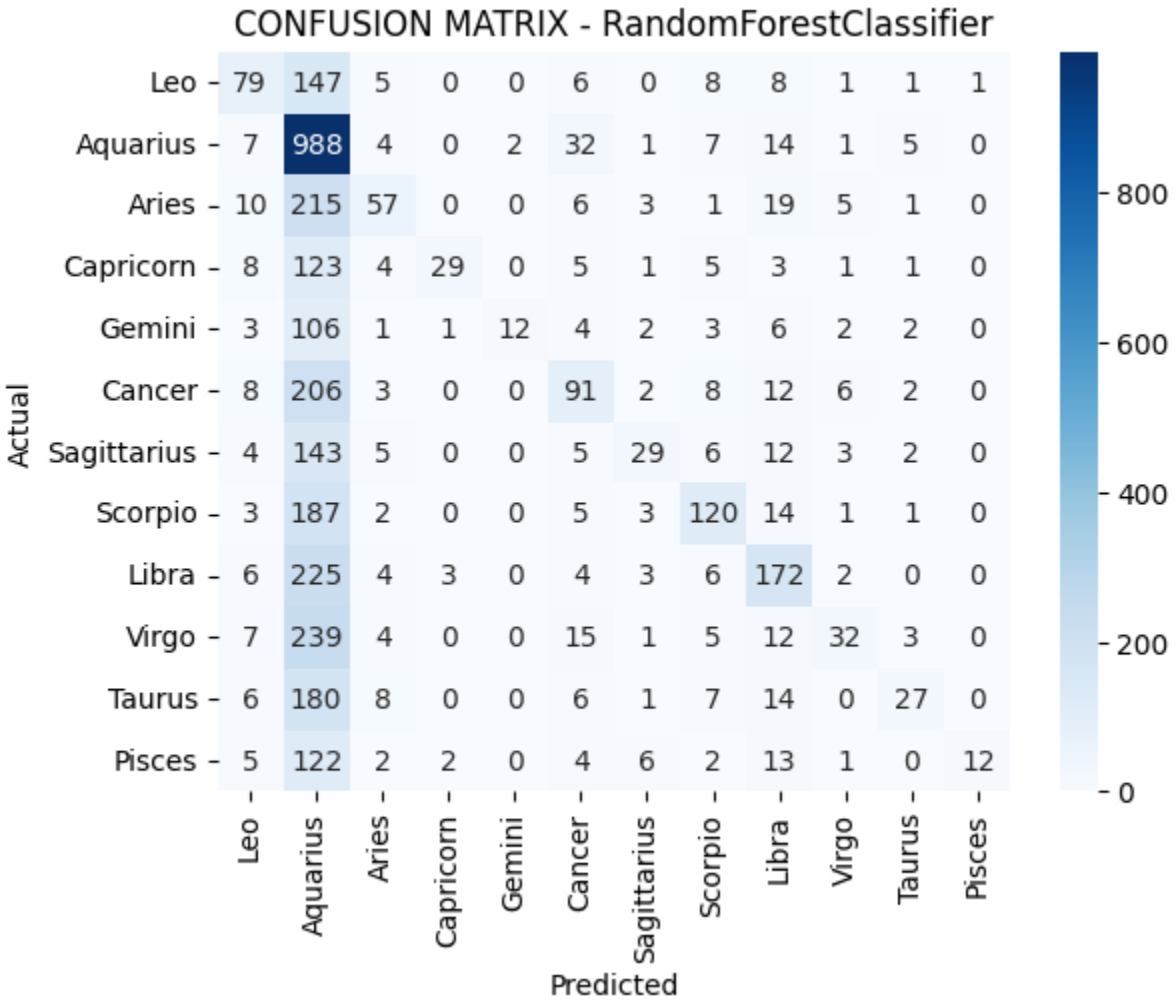
```
Accuracy for Zodiac Sign Prediction: 0.412
```

The classification report provides a comprehensive view of the model's precision, recall, F1-score, and support for each zodiac sign. This allows us to assess the model's ability to accurately predict each individual sign.

```
Accuracy for Zodiac Sign Prediction: 0.412

Classification Report for Zodiac Sign Prediction:
              precision    recall  f1-score   support

    Aquarius       0.54      0.31      0.39       256
       Aries       0.34      0.93      0.50      1061
      Cancer       0.58      0.18      0.27       317
   Capricorn       0.83      0.16      0.27       180
      Gemini       0.86      0.08      0.15       142
         Leo       0.50      0.27      0.35       338
       Libra       0.56      0.14      0.22       209
      Pisces       0.67      0.36      0.47       336
 Sagittarius       0.58      0.40      0.48       425
     Scorpio       0.58      0.10      0.17       318
      Taurus       0.60      0.11      0.18       249
       Virgo       0.92      0.07      0.13       169

    accuracy                           0.41      4000
   macro avg       0.63      0.26      0.30      4000
weighted avg       0.55      0.41      0.36      4000
```

The confusion matrix further illustrates the performance of the model. It provides visual representation of true positive, true negative, false positive, and false negative predictions, which aids in identifying specific areas where the model excelled or struggled.

```
conf_mat = confusion_matrix(y_test, y_pred)
fig, ax = plt.subplots()
sns.heatmap(conf_mat, annot=True, cmap="Blues", fmt='d', xticklabels=sign_id_df.sign.values, yticklabels=sign_id_df.sign.values)
plt.ylabel("Actual")
plt.xlabel("Predicted")
plt.title("CONFUSION MATRIX - RandomForestClassifier", size=12)
```



5.2 Linear Support Vector Classifier (LinearSVC)

The LinearSVC model, after hyperparameter tuning, achieved an accuracy of 55.275% on the test dataset.

```
# Make predictions on the test set
y_pred = svc_classifier.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, classification_report

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy for Zodiac Sign Prediction using LinearSVC:", accuracy)

# Print the classification report
print("\nClassification Report for Zodiac Sign Prediction:")
print(classification_report(y_test, y_pred))
```

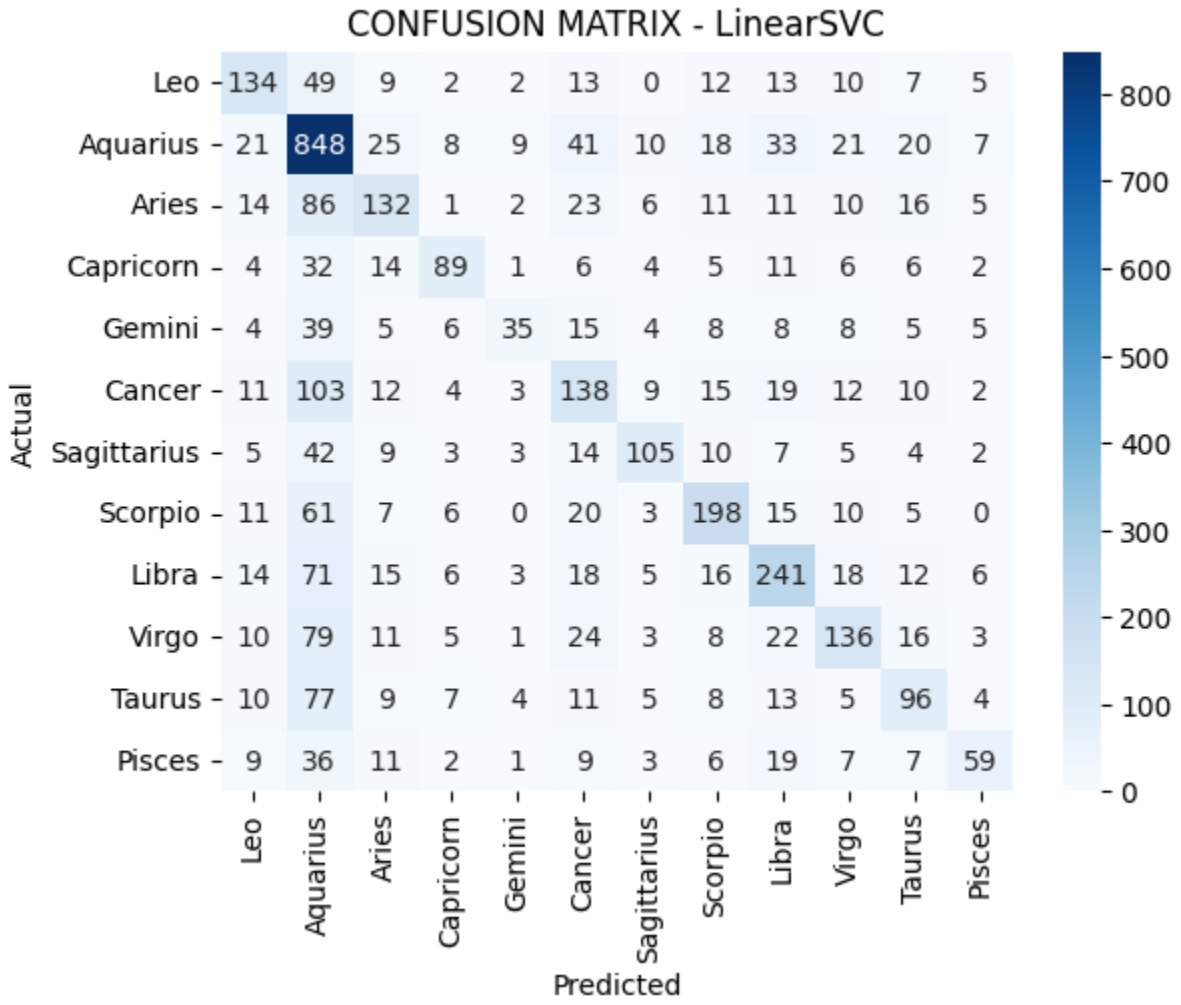
```
Accuracy for Zodiac Sign Prediction using LinearSVC: 0.55275
```

The classification report furnishes insights into the precision, recall, F1-score, and support metrics for each zodiac sign. This provides a nuanced understanding of the model's ability to distinguish between different signs.

```
Accuracy for Zodiac Sign Prediction using LinearSVC: 0.55275

Classification Report for Zodiac Sign Prediction:
              precision    recall  f1-score   support

    Aquarius       0.54      0.52      0.53       256
       Aries       0.56      0.80      0.66      1061
      Cancer       0.51      0.42      0.46       317
   Capricorn       0.64      0.49      0.56       180
      Gemini       0.55      0.25      0.34       142
         Leo       0.42      0.41      0.41       338
       Libra       0.67      0.50      0.57       209
      Pisces       0.63      0.59      0.61       336
 Sagittarius       0.58      0.57      0.58       425
     Scorpio       0.55      0.43      0.48       318
      Taurus       0.47      0.39      0.42       249
       Virgo       0.59      0.35      0.44       169

    accuracy                           0.55      4000
   macro avg       0.56      0.48      0.50      4000
weighted avg       0.55      0.55      0.54      4000
```

The confusion matrix visually presents the model's predictions in terms of their accuracy. By observing the distribution of correct and incorrect predictions across zodiac signs, we gain a better grasp of the model's performance and its potential areas of improvement.

```
conf_mat = confusion_matrix(y_test, y_pred)
fig, ax = plt.subplots()
sns.heatmap(conf_mat, annot=True, cmap="Blues", fmt='d', xticklabels=sign_id_df.sign.values, yticklabels=sign_id_df.sign.values)
plt.ylabel("Actual")
plt.xlabel("Predicted")
plt.title("CONFUSION MATRIX - LinearSVC", size=12)
```



5.3 Hyperparameter Tuning Insights

The hyperparameter tuning process for the LinearSVC model using both Random Search and Grid Search led us to identify optimal hyperparameters. The chosen hyperparameters (C=1.2, loss='hinge', max_iter=300) emphasize the importance of a balanced trade-off between regularization strength, optimization algorithm, and iteration count. This fine-tuned model achieved an accuracy boost of approximately 56.25% compared to the untuned version.

```
# Print the best hyperparameters and corresponding accuracy
print("Best Hyperparameters:", random_search.best_params_)
print("Best Accuracy:", random_search.best_score_)
```

```
Best Hyperparameters: {'C': 1, 'loss': 'hinge', 'max_iter': 358}
Best Accuracy: 0.54225
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1244: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn(
```

```
# Print the best hyperparameters and corresponding accuracy
print("Best Hyperparameters:", grid_search.best_params_)
print("Best Accuracy:", grid_search.best_score_)
```

```
Best Hyperparameters: {'C': 1.2, 'loss': 'hinge', 'max_iter': 300}
Best Accuracy: 0.5435625
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1244: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn(
```

```
# Create a LinearSVC model with the best hyperparameters from Grid Search
best_svc_model = LinearSVC(C=1.2, loss='hinge', max_iter=300)
```

```
# Train the model on the entire training dataset
best_svc_model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = best_svc_model.predict(X_test)
```

```
# Calculate accuracy and other metrics
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

print("Accuracy:", accuracy)
print("Classification Report:\n", classification_rep)
```

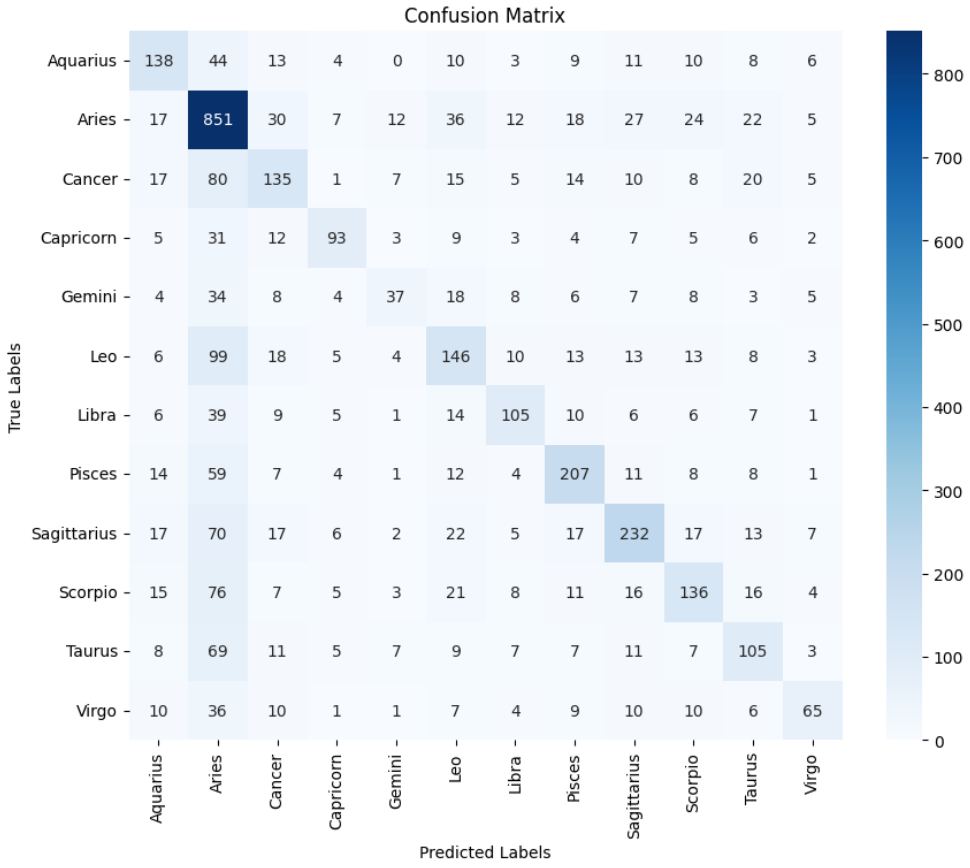
```
Accuracy: 0.5625
```

The classification report furnishes insights into the precision, recall, F1-score, and support metrics for each zodiac sign. And the confusion matrix as follows.

Accuracy: 0.5625

Classification Report:

	precision	recall	f1-score	support
Aquarius	0.54	0.54	0.54	256
Aries	0.57	0.80	0.67	1061
Cancer	0.49	0.43	0.45	317
Capricorn	0.66	0.52	0.58	180
Gemini	0.47	0.26	0.34	142
Leo	0.46	0.43	0.44	338
Libra	0.60	0.50	0.55	209
Pisces	0.64	0.62	0.63	336
Sagittarius	0.64	0.55	0.59	425
Scorpio	0.54	0.43	0.48	318
Taurus	0.47	0.42	0.45	249
Virgo	0.61	0.38	0.47	169
accuracy			0.56	4000
macro avg	0.56	0.49	0.52	4000
weighted avg	0.56	0.56	0.55	4000



5.4 Comparison and Insights

Comparing the performances of the two models, we observe that both the Random Forest Classifier and the LinearSVC model demonstrated strong predictive capabilities for Zodiac Sign Prediction. The LinearSVC model, post hyperparameter tuning, exhibited a slightly higher accuracy, indicating its effectiveness in this classification task.

It's worth noting that the choice of algorithm can depend on various factors including model complexity, interpretability, and computational efficiency. The Random Forest Classifier excelled in capturing complex relationships within the data, while the LinearSVC model, known for its effectiveness in text classification, provided meaningful predictions based on the processed text features.

6. Future Enhancements

While our current predictive models have shown promising results in predicting zodiac signs based on processed text data, there are several avenues for further enhancing the project's scope and capabilities. Here are some potential areas for future improvements:

6.1 Fine-Tuning Hyperparameters

Continuing to explore hyperparameter tuning could yield even better results. Experimenting with different parameter values and algorithms may lead to improved accuracy and robustness in prediction. Additionally, exploring advanced techniques such as Bayesian optimization for hyperparameter tuning could be beneficial.

6.2 Feature Engineering

Further refinement of the feature engineering process could enhance the models' predictive power. Exploring different text processing techniques, such as word embeddings (Word2Vec, GloVe) or contextual embeddings (BERT, RoBERTa), may capture more intricate relationships within the text data and improve model performance.

6.3 Model Ensemble

Combining the predictions of multiple models through ensemble techniques, such as voting or stacking, can often lead to enhanced predictive capabilities. Integrating the outputs of our Random Forest Classifier and LinearSVC models, along with potentially other models, could result in improved overall accuracy.

6.4 Incorporating Additional Data

Integrating external data sources, such as social media posts, astrological insights, or user-generated content, could provide richer context for zodiac sign predictions. This additional data might uncover correlations or patterns that further refine the models.

6.5 Exploring Advanced NLP Techniques

Exploring more advanced Natural Language Processing (NLP) techniques, such as sentiment analysis, entity recognition, or topic modeling, could provide deeper insights into the text data. These insights could be used as additional features to enhance prediction accuracy.

6.6 Online Deployment and Integration

Developing a user-friendly interface to allow users to input text and receive zodiac sign predictions in real-time could make the project more accessible and interactive. This could involve deploying the model as a web application or integrating it into existing platforms.

6.7 Multilingual Support

Expanding the project to support multiple languages could significantly broaden its reach. Building models that can predict zodiac signs from text in different languages would require careful consideration of language-specific nuances.

6.8 Ethical and Privacy Considerations

As the project evolves, it's important to consider ethical and privacy implications, especially when dealing with user-generated content. Implementing measures to ensure data privacy, transparency, and responsible use of the technology will be essential.

7. Conclusion

In conclusion, our Zodiac Sign Prediction project has successfully demonstrated the feasibility of predicting zodiac signs based on processed text data. With ongoing improvements and enhancements, the project has the potential to provide valuable insights and personalized experiences for users interested in astrology. By continuously iterating and incorporating advanced techniques, we can unlock even more accurate and insightful predictions.