Enhancing Music Genre Classification Through PCA and Logistic Regression: A Comprehensive Approach to Data Preprocessing, Dimensionality Reduction, and Model Evaluation

Introduction to Music Classification

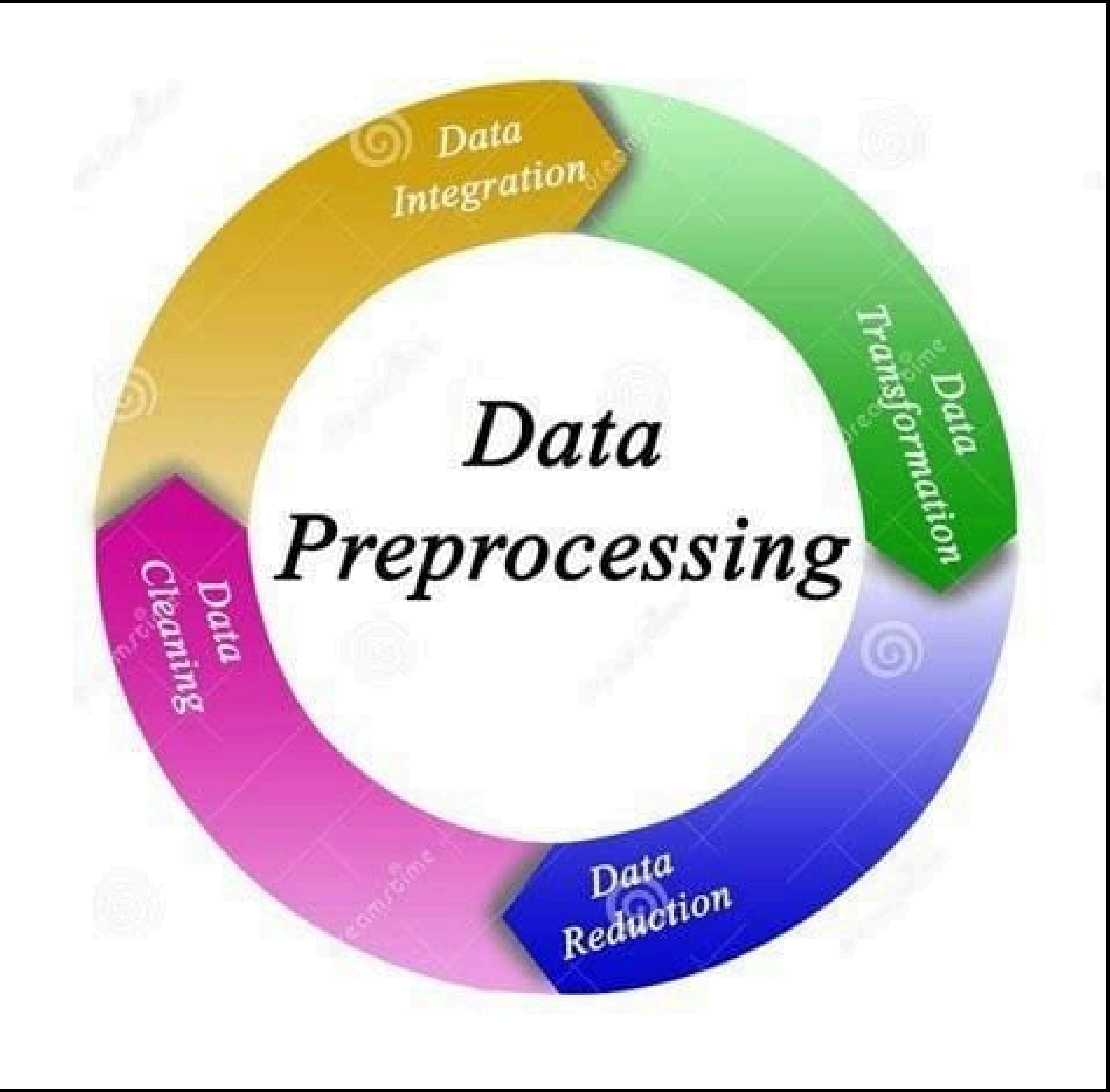
Music genre classification is a vital task in the field of audio processing. This presentation explores how **PCA (Principal Component Analysis)** and **Logistic Regression** can enhance this classification process. We will discuss data preprocessing, dimensionality reduction, and model evaluation techniques that lead to improved accuracy.

```python
data = pd.read_csv('/content/music_dataset_mod.csv')
data.head()
```

	Tempo	Dynamics Range	Vocal Presence	Percussion Strength	String Instrument Detection	Electronic Element Presence	Rhythm Complexity	Drums Influence	Distorted Guitar	Metal Frequencies	Ambient Sound Influence	Instrumental Overlaps	Genre
0	114.618354	57.976367	53.251766	99.061840	14.686768	17.628630	46.545522	75.839434	79.378892	71.753088	96.439665	53.771763	Country
1	116.672803	69.387087	95.787280	90.831033	47.280419	-15.618194	85.421085	100.455908	0.713015	0.000000	17.327295	15.017146	Classical
2	128.328121	52.930677	65.701187	104.439247	5.984994	50.467388	18.006722	77.642913	80.652946	87.692110	95.125207	25.308020	Rock
3	128.511337	25.494755	14.095374	40.106130	47.715584	87.335201	68.603329	63.536557	74.888346	76.239108	97.016998	96.893109	Hip-hop
4	135.474190	45.174876	101.469872	70.002203	108.177637	25.865590	31.295163	81.121030	36.178193	23.381542	53.753793	30.142986	Country

```python
data.columns
```

```
Index(['Tempo', 'Dynamics Range', 'Vocal Presence', 'Percussion Strength',
       'String Instrument Detection', 'Electronic Element Presence',
       'Rhythm Complexity', 'Drums Influence', 'Distorted Guitar',
       'Metal Frequencies', 'Ambient Sound Influence', 'Instrumental Overlaps',
       'Genre', 'Genre_Encoded'],
      dtype='object')
```



Data Preprocessing Techniques

Effective **data preprocessing** is crucial for successful classification. This involves steps like **normalization**, **feature extraction**, and **noise reduction**. By ensuring that the data is clean and well-prepared, we set a solid foundation for further analysis and modeling.

```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 13 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Tempo                         1000 non-null   float64
 1   Dynamics Range                1000 non-null   float64
 2   Vocal Presence                1000 non-null   float64
 3   Percussion Strength           1000 non-null   float64
 4   String Instrument Detection   1000 non-null   float64
 5   Electronic Element Presence   1000 non-null   float64
 6   Rhythm Complexity             1000 non-null   float64
 7   Drums Influence               1000 non-null   float64
 8   Distorted Guitar              1000 non-null   float64
 9   Metal Frequencies             1000 non-null   float64
 10  Ambient Sound Influence       1000 non-null   float64
 11  Instrumental Overlaps         1000 non-null   float64
 12  Genre                         890 non-null    object
dtypes: float64(12), object(1)
memory usage: 101.7+ KB
```

[5]
```python
data['Genre'].unique()
```

```
array(['Country', 'Classical', 'Rock', 'Hip-hop', nan, 'Jazz'],
      dtype=object)
```

```python
# Checking for missing values
data.isnull().sum()
```

	0
Tempo	0
Dynamics Range	0
Vocal Presence	0
Percussion Strength	0
String Instrument Detection	0
Electronic Element Presence	0
Rhythm Complexity	0
Drums Influence	0
Distorted Guitar	0
Metal Frequencies	0
Ambient Sound Influence	0
Instrumental Overlaps	0
Genre	110

```
[7] label_encoder = LabelEncoder()
    data['Genre_Encoded'] = label_encoder.fit_transform(data['Genre'])
```

```
# Select numerical features (assuming all except 'Genre' and 'Genre_Encoded')
numerical_features = data.drop(['Genre','Genre_Encoded'],axis=1)

#Scale numerical faetures
scaler = StandardScaler()
scaled_features = scaler.fit_transform(numerical_features)

# Convert a new Dataframe with scaled features
scaled_data = pd.DataFrame(scaled_features, columns=numerical_features.columns)
scaled_data['Genre_Encoded'] = data['Genre_Encoded']
```

Understanding PCA

Principal Component Analysis (PCA) is a technique used for **dimensionality reduction**. It transforms high-dimensional data into a lower-dimensional format while retaining most of the variance. This slide will cover how PCA helps in simplifying the dataset for better model performance.

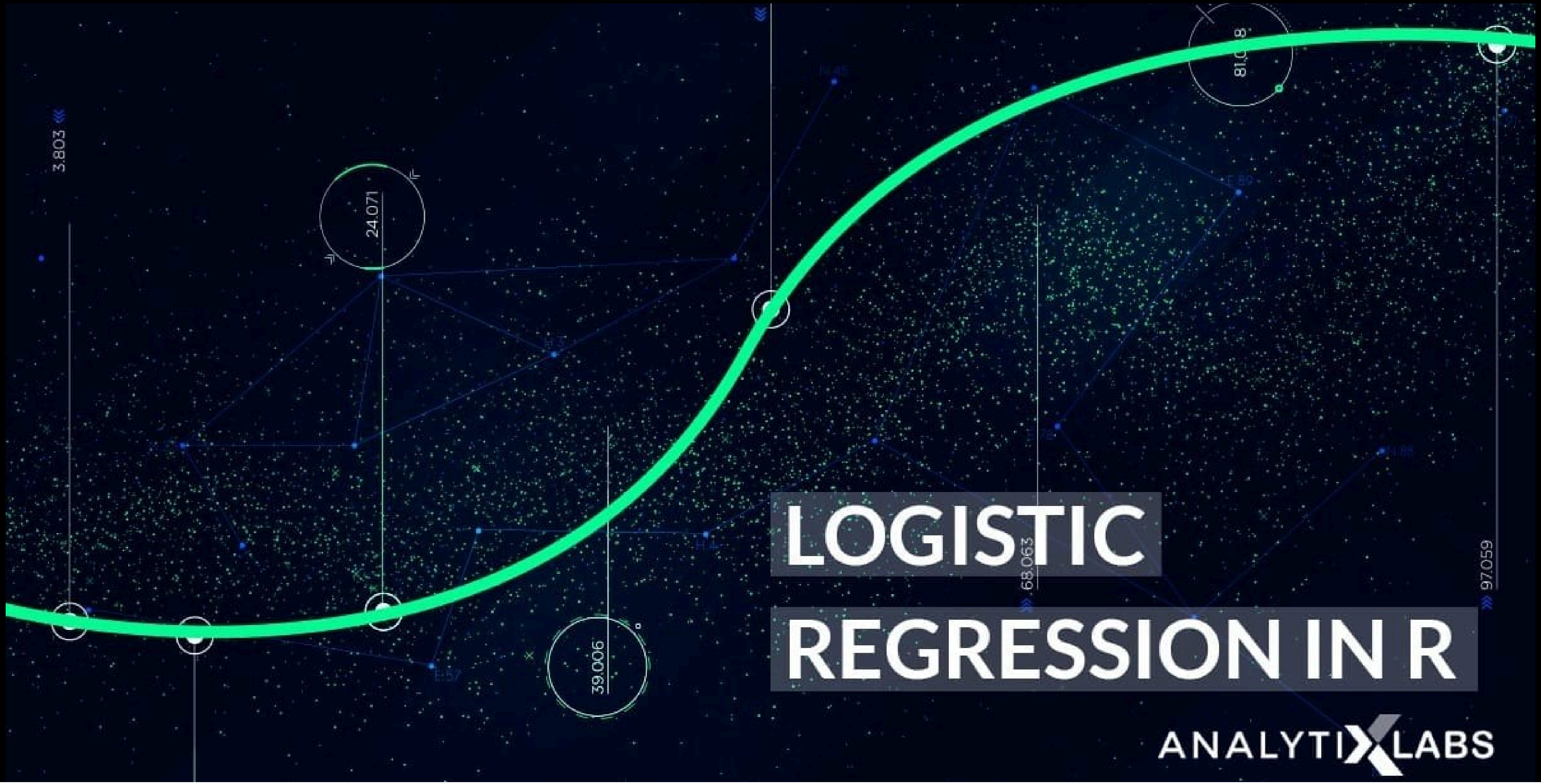
```python
# Apply PCA
pca = PCA(n_components=0.95)  # Explain 95% of variance
pca_features = pca.fit_transform(scaled_data.drop(['Genre_Encoded'], axis=1))

# Create a DataFrame with PCA features
pca_data = pd.DataFrame(pca_features, index=data.index)
pca_data['Genre_Encoded'] = data['Genre_Encoded']  # Add back the target variable
```

```python
X_train, X_test, y_train, y_test = train_test_split(
    pca_data.drop(['Genre_Encoded'], axis=1),
    pca_data['Genre_Encoded'],
    test_size=0.2,
    random_state=42
)
```

```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

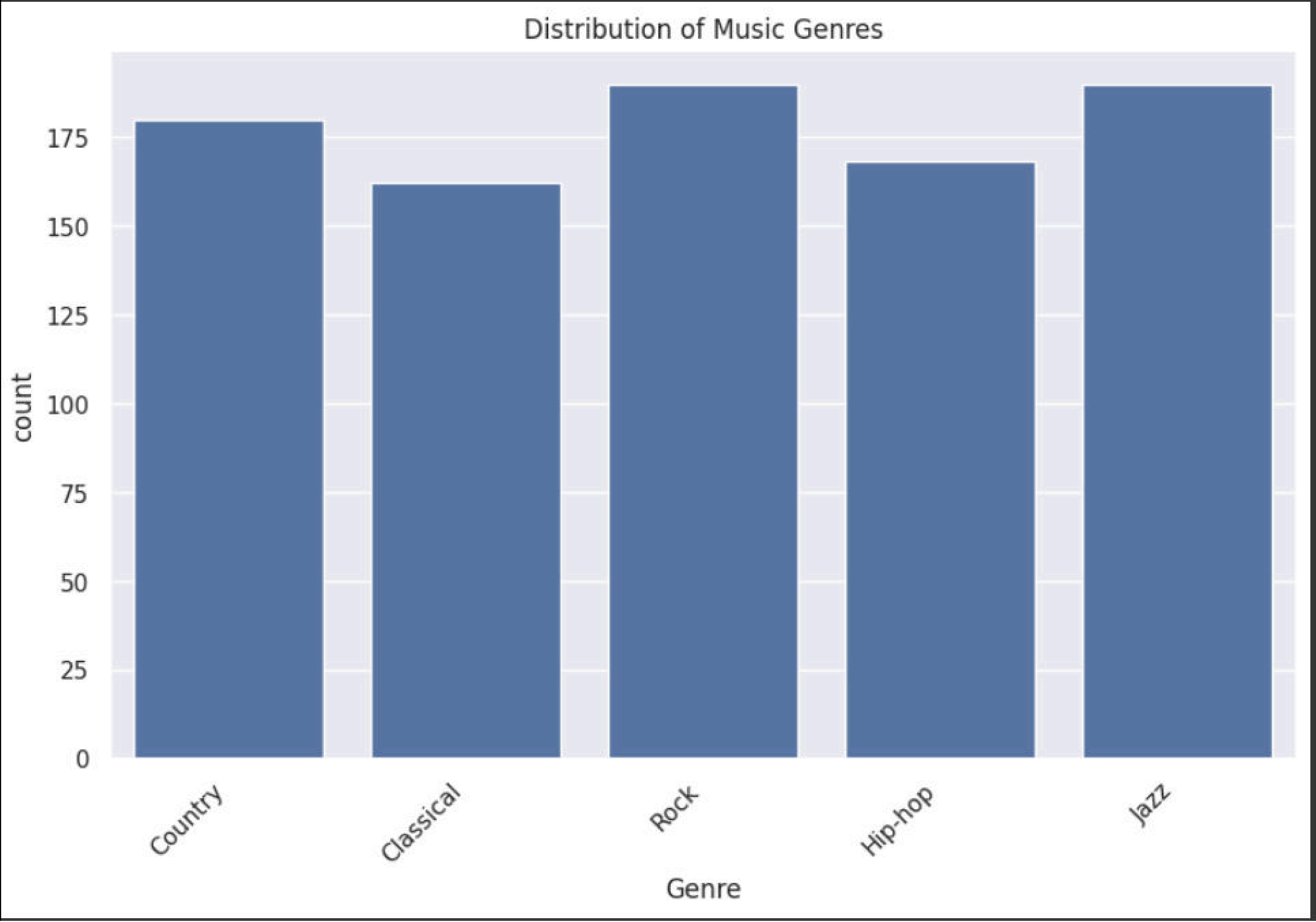
```
LogisticRegression
LogisticRegression(max_iter=1000)
```

```
12] y_pred = model.predict(X_test)
```



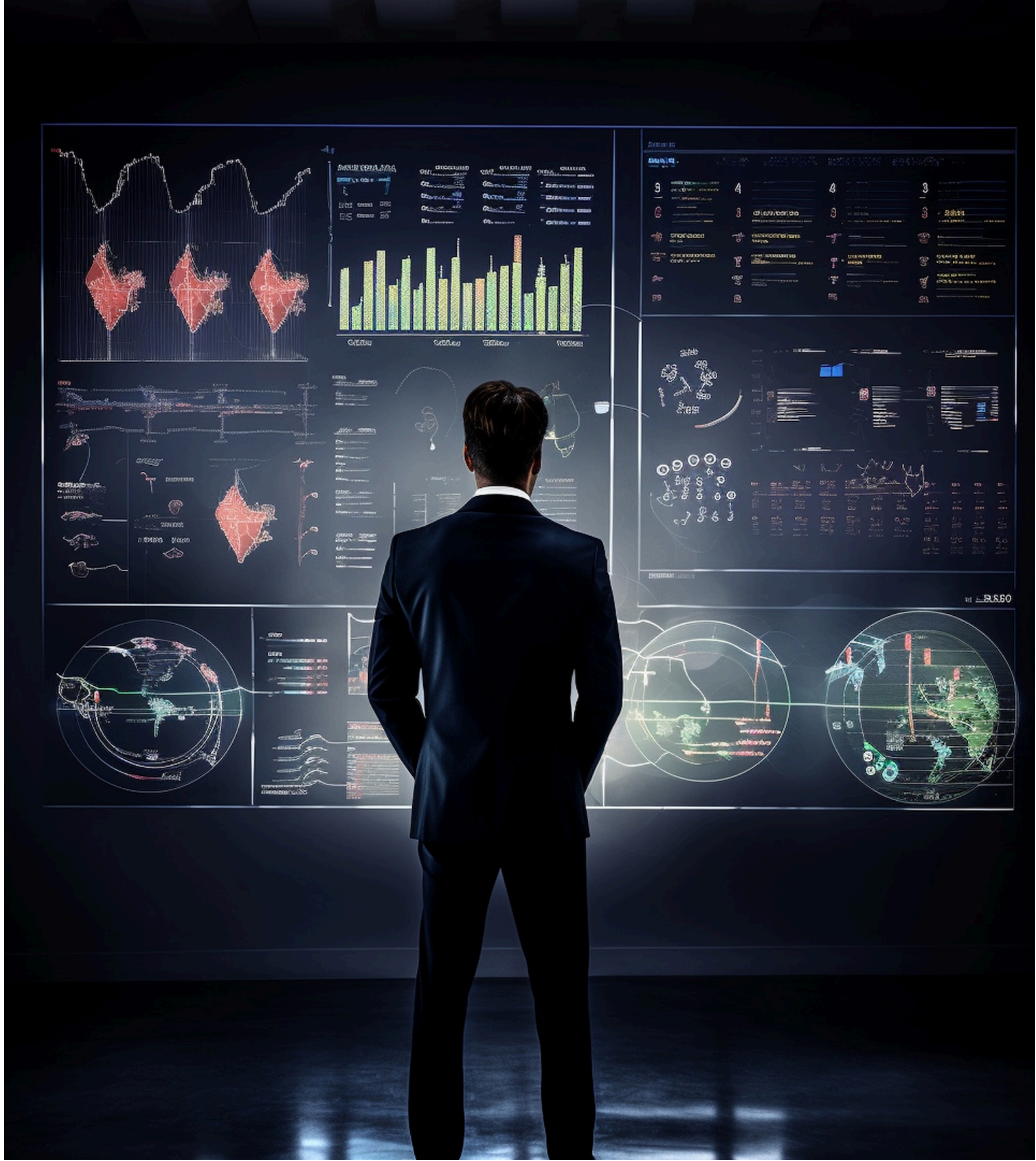
Logistic Regression Overview

Logistic Regression is a statistical method for predicting binary outcomes. In the context of music genre classification, it helps in understanding the relationship between features and genre labels. We will discuss its application and effectiveness in this domain.


Distribution of Music Genres
count
175
150
125
100
75
50
25
0
Country
Classical
Rock
Hip-hop
Jazz
Genre

Model Evaluation Metrics

Evaluating the performance of our model is essential. We will focus on key **evaluation metrics** such as **accuracy**, **precision**, and **recall**. Understanding these metrics allows us to assess the effectiveness of our classification approach accurately.



```python
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")

# Print classification report for detailed evaluation
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.515
              precision    recall  f1-score   support

           0       0.73      0.95      0.83        40
           1       0.46      0.36      0.41        36
           2       0.36      0.39      0.38        31
           3       0.51      0.62      0.56        39
           4       0.44      0.55      0.49        29
           5       0.00      0.00      0.00        25

    accuracy                           0.52       200
   macro avg       0.42      0.48      0.44       200
weighted avg       0.45      0.52      0.48       200
```

Conclusion and Future Work

In conclusion, combining **PCA** and **Logistic Regression** significantly enhances music genre classification accuracy. Future work may involve exploring **deep learning** techniques and other advanced algorithms to further improve classification results and explore new genres.

The weighted average f1-score is 0.52, indicating the model has some classification capability but may benefit from further tuning.

To improve performance, we could try the following:

- Adjusting PCA components to optimize dimensionality reduction.
- Hyperparameter tuning of the logistic regression model.
- Exploring additional classifiers (e.g., SVM, Random Forest) for comparison.

Thanks!

By Pranav Mishra
(Data Analyst Fellow)