Exploratory Data Analysis (EDA) Report on Iris Dataset

Project Overview

Project ID: #CC69856

Project Title: Predicting Employee Attrition

Internship Domain: Data Science Intern

Project Level: - Intermediate Level

Assigned By: CodeClause Internship

Name: Omar Samir Mohamed

1. Introduction

Aim

Develop a model to predict the likelihood of employee attrition in a company.

Description

Utilize HR data to build a classification model that predicts whether an employee is likely to leave the company.

2. Data Overview

Describe the dataset used for analysis and modeling.

Dataset Description:

- `MMM-YY`: The month and year of the record.
- `Emp_ID`: Employee ID.
- `Age`: Age of the employee.
- `Gender`: Gender of the employee.
- `City`: City where the employee is located.
- `Education_Level`: Education level of the employee.
- `Salary`: Employee's salary.
- `Dateofjoining`: Date the employee joined the company.
- `LastWorkingDate`: The last working date of the employee.
- `Joining Designation`: Designation when the employee joined.
- `Designation`: Current designation.
- `Total Business Value`: Business value associated with the employee.
- `Quarterly Rating`: Performance rating for the quarter.

3. Data Preprocessing

- **Handling Missing Values**:
  Last Working Date filled with '2100-01-01'.

```
# Handle missing values

company['LastWorkingDate'].fillna('2100-01-01', inplace=True)
```

- **Encoding Categorical Variables**:
  - Gender, City, and Education Level encoded using LabelEncoder.

```
# Encode categorical variables

label_encoder = LabelEncoder()

company['Gender'] = label_encoder.fit_transform(company['Gender'])

company['City'] = label_encoder.fit_transform(company['City'])

company['Education_Level'] = label_encoder.fit_transform(company['Education_Level'])
```

- **Feature Engineering**:
  - Created a target variable 'Attrition' based on Last Working Date.

```
# Create a target column for attrition

company['Attrition'] = company['LastWorkingDate'] != pd.to_datetime('2100-01-01')
```

- **Feature Selection**:
  - Dropped unnecessary columns: MMM-YY, Emp_ID, Date of Joining, Last Working Date.

```
# Drop unnecessary columns

company.drop(columns=['MMM-YY', 'Emp_ID', 'Dateofjoining', 'LastWorkingDate'], inplace=True)
```

4. Exploratory Data Analysis (Optional)

- Summarize key insights and visualizations from exploring the dataset. Include any notable trends or patterns observed.

5. Model Training and Evaluation

- **Splitting Data**:
  - Split dataset into training (80%) and testing (20%) sets.

```
# Define features and target

X = company.drop(columns=['Attrition'])

y = company['Attrition']

# 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)
```

- **Model Selection**:
  - Chose RandomForestClassifier with 100 estimators.

```
# Initialize and train the model (example: Random Forest)

model = RandomForestClassifier(n_estimators=100)

model.fit(X_train, y_train)
```

- **Model Evaluation**:
  - Evaluated model performance on the test set using:
    - Classification Report
    - ROC-AUC Score

```
# Evaluate the model

y_pred = model.predict(X_test)

y_pred_prob = model.predict_proba(X_test)[:, 1]

print('Model Performance:')

print(classification_report(y_test, y_pred))

print(f'ROC-AUC: {roc_auc_score(y_test, y_pred_prob)}')
```

6. Model Performance

Classification Report

[Insert Classification Report Output]

ROC Curve

- **ROC-AUC Score**: [ROC-AUC Score]

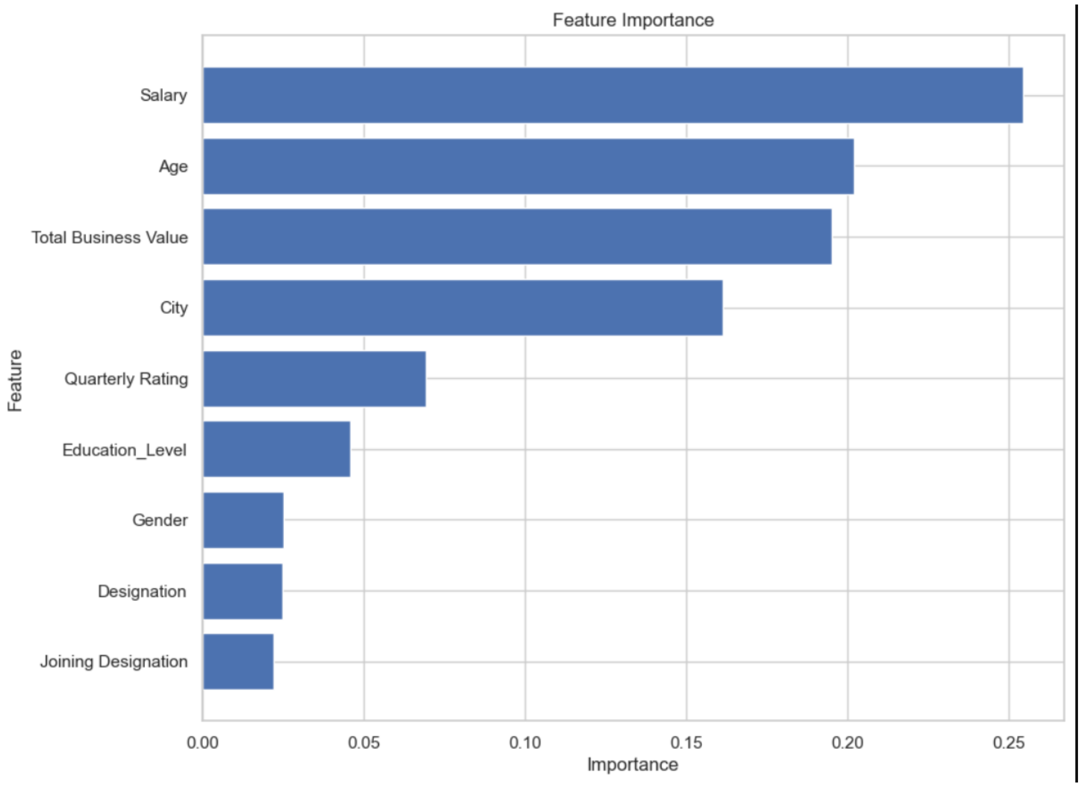
Feature Importance

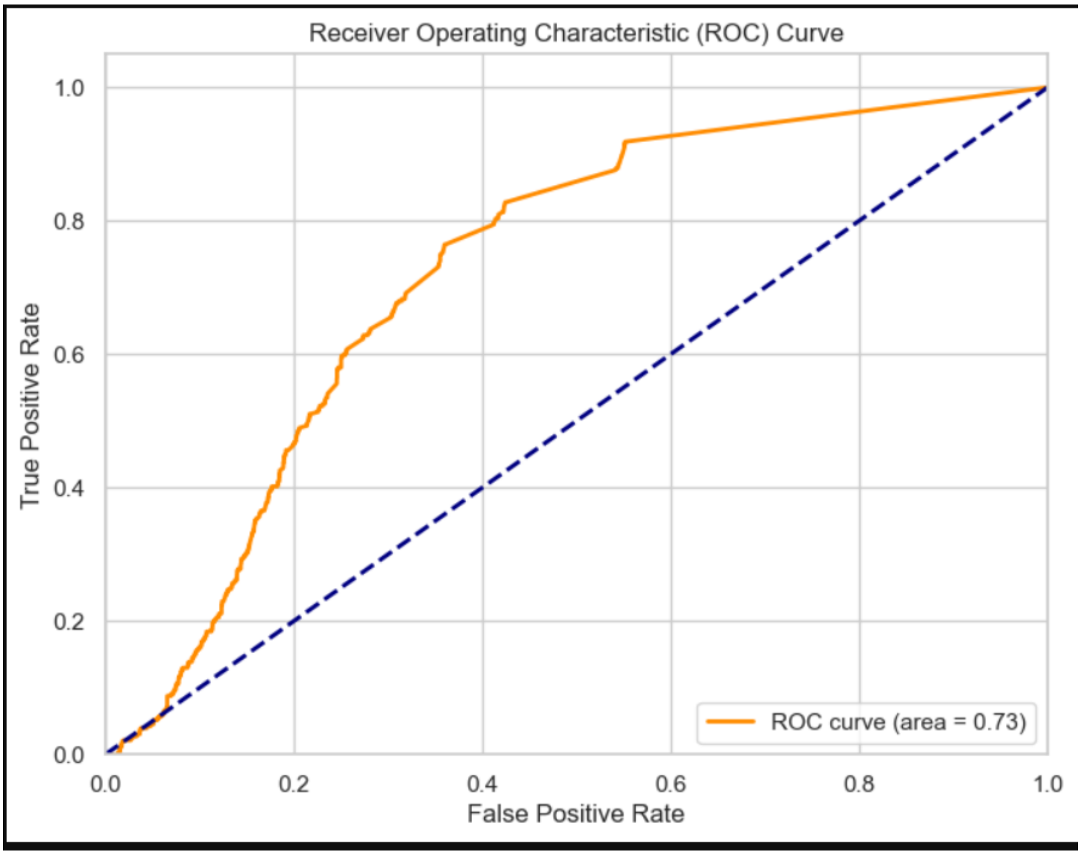
- **Key Insights**:
  - [Provide insights from feature importance plot]

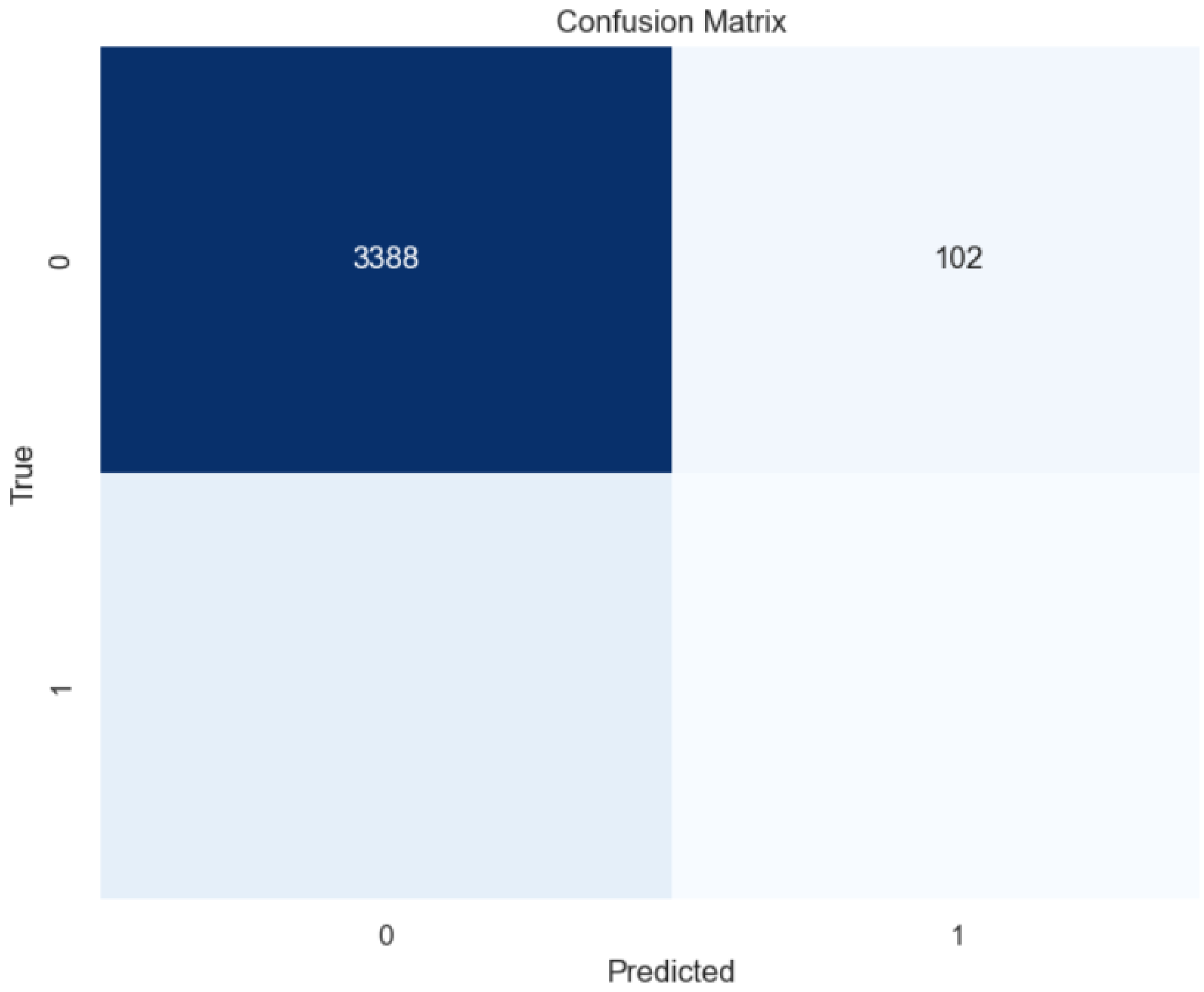
7. Predictions

- **Prediction Results**:
  - Predicted attrition for all employees in the dataset.

```
# Predict attrition for all employees in the dataset
company_scaled = scaler.transform(X)
predictions = model.predict(company_scaled)
prediction_probs = model.predict_proba(company_scaled)[:, 1]

# Add predictions to the dataset
company['Attrition Prediction'] = ['Leave' if pred else 'Stay' for pred in predictions]
company['Probability of Leaving'] = prediction_probs * 100  # Convert to percentage

# Save the results to a new CSV file
company.to_csv('company_attrition_predictions.csv', index=False)

print("Predictions saved to 'company_attrition_predictions.csv'")
```


Feature Importance
Salary
Age
Total Business Value
City
Quarterly Rating
Education_Level
Gender
Designation
Joining Designation
Feature
0.00
0.05
0.10
0.15
0.20
0.25
Importance


Receiver Operating Characteristic (ROC) Curve
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve (area = 0.73)



8. Conclusion

In this project, we aimed to predict employee attrition within the company using a machine learning approach. The steps taken included data preprocessing, model training, evaluation, and generating predictions for all employees. Here’s a summary of the key steps and findings: