Data-Driven Analysis

Answer
Predictive Questions

For HR Data Analysis

DATA DYNAMOS

Harnessing the Power of Data to Drive Innovation

Data-Driven Analysis

This report presents a data-driven analysis of employee behavior and performance using HR datasets. The goal is to uncover insights that help improve employee engagement, predict attrition, and optimize performance management strategies through machine learning models. By leveraging both descriptive and predictive analytics, we aim to provide practical recommendations to support better talent management and recruitment decision-making.

Overview of the Code Structure

The code is structured into five main stages:

The code is organized into five key stages to perform a data-driven HR analysis:

1. **Data Loading and Preprocessing: Loads employee data from "HrData.xlsx," merges it, and cleans it by handling missing values, removing outliers, and converting categorical data to numeric format.**
2. **Employee Engagement Analysis: Analyzes work environment satisfaction and its relationship with salary to assess employee satisfaction levels.**
3. **Correlation Analysis: Examines relationships between features like age, salary, and tenure to identify patterns.**
4. **Predictive Analytics: Builds and evaluates a model to predict employee attrition, identifying key factors influencing turnover.**
5. **Visualization and Insights: Generates charts (boxplots, heatmaps, bar charts) and reports to visualize findings and provide actionable HR insights.**

1: Data Loading and Preprocessing

We assume the Excel file contains at least two sheets with the following structure:

'Employee' Sheet: Columns include EmployeeID, Age, DistanceFromHome, Department, JobRole, Salary, HireDate, Attrition, YearsAtCompany, YearsSinceLastPromotion.

'PerformanceRating' Sheet:

Columns include EmployeeID, ReviewDate, EnvironmentSatisfaction, JobSatisfaction, WorkLifeBalance, ManagerRating.

What the Code Does:

-Defines the file path (` file_path = "HrData.xlsx"`) and loads it using ` pd.ExcelFile.`

-Reads the "Employee" sheet into ` df_employee` and the "PerformanceRating" sheet into ` df_performance.`

-Merges these two DataFrames on ` EmployeeID` using a left join (` df_combined = df_employee.merge(...)`), keeping all employee records and adding ` EnvironmentSatisfaction` where available.

Prepares the data by:

- Creating a numeric column ` WorkEnvironmentSatisfactionNumeric` from ` EnvironmentSatisfaction.`

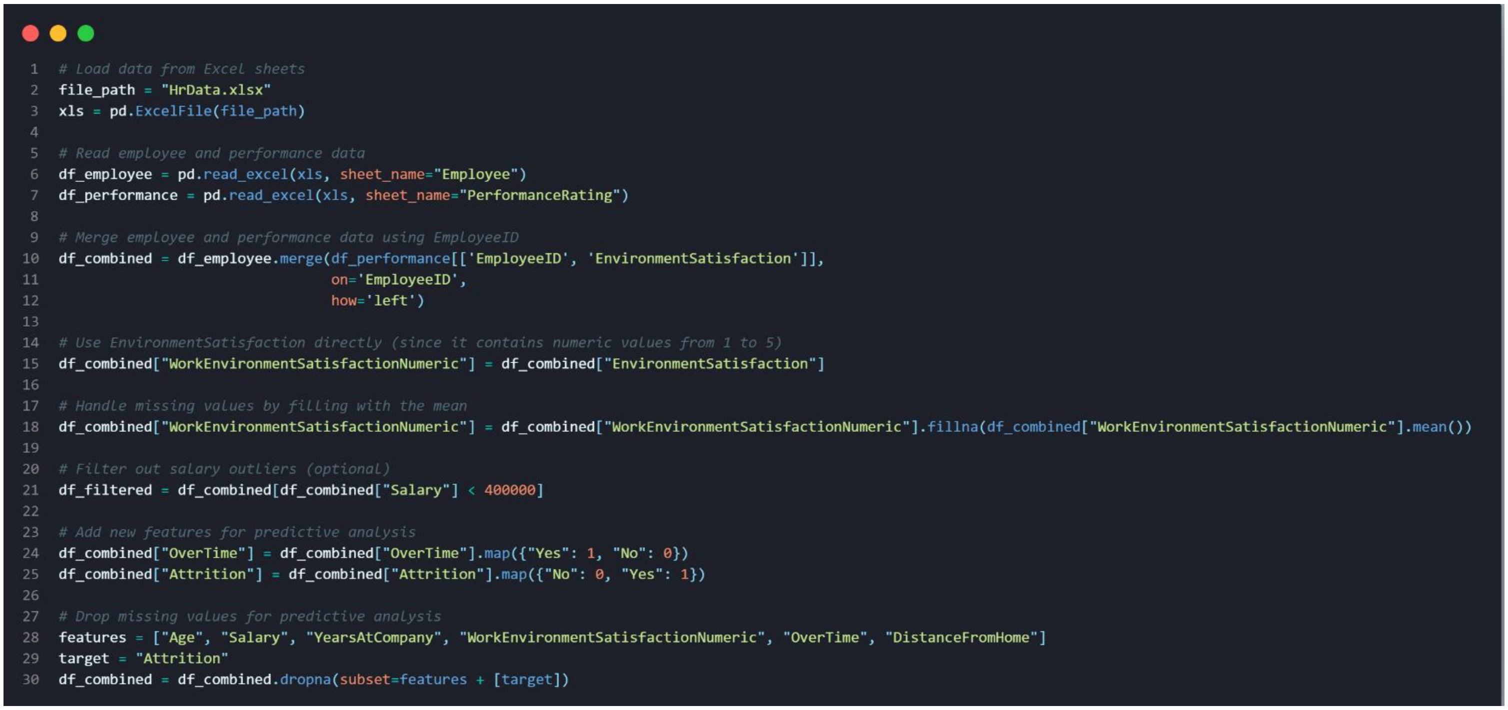
- Filling missing values in this column with the mean.

- Filtering out extreme salaries (` Salary < 400000`) into ` df_filtered.`

- Converting categorical columns (` OverTime` and ` Attrition`) to numeric values (0 and 1) for modeling.

Results

- A clean, merged dataset with numerical features ready for analysis.



```python
# Load data from Excel sheets
file_path = "HrData.xlsx"
xls = pd.ExcelFile(file_path)

# Read employee and performance data
df_employee = pd.read_excel(xls, sheet_name="Employee")
df_performance = pd.read_excel(xls, sheet_name="PerformanceRating")

# Merge employee and performance data using EmployeeID
df_combined = df_employee.merge(df_performance[['EmployeeID', 'EnvironmentSatisfaction']],
                                on='EmployeeID',
                                how='left')

# Use EnvironmentSatisfaction directly (since it contains numeric values from 1 to 5)
df_combined["WorkEnvironmentSatisfactionNumeric"] = df_combined["EnvironmentSatisfaction"]

# Handle missing values by filling with the mean
df_combined["WorkEnvironmentSatisfactionNumeric"] = df_combined["WorkEnvironmentSatisfactionNumeric"].fillna(df_combined["WorkEnvironmentSatisfactionNumeric"].mean())

# Filter out salary outliers (optional)
df_filtered = df_combined[df_combined["Salary"] < 400000]

# Add new features for predictive analysis
df_combined["OverTime"] = df_combined["OverTime"].map({"Yes": 1, "No": 0})
df_combined["Attrition"] = df_combined["Attrition"].map({"No": 0, "Yes": 1})

# Drop missing values for predictive analysis
features = ["Age", "Salary", "YearsAtCompany", "WorkEnvironmentSatisfactionNumeric", "OverTime", "DistanceFromHome"]
target = "Attrition"
df_combined = df_combined.dropna(subset=features + [target])
```

2: Employee Engagement Analysis

Objective

This analysis evaluates employee satisfaction using `EnvironmentSatisfaction` and explores its relationship with `Salary` (as a proxy for performance or reward).

It aims to identify if a satisfying work environment correlates with higher salaries and highlights areas for improvement if satisfaction is low.

Steps in the Code

1)Data Preparation: Uses `df_filtered` (outliers removed) and `WorkEnvironmentSatisfactionNumeric` (1-5 scale).

2)Visualization:

- Creates a boxplot (`sns.boxplot`) showing salary distribution across satisfaction levels (1.0 to 5.0).
- Adds mean salary values on the plot for each level (e.g., 102473 for level 1.0, 103311 for level 2.0, etc.).
- Includes labels and a grid for clarity.

3)Output: Prints the average `WorkEnvironmentSatisfactionNumeric` score, which is 3.87255592487703087.

Interpretation:

Average Environmental Satisfaction: The average satisfaction score is approximately 3.87 (on a 1-5 scale). This indicates a generally positive work environment, as the score is closer to 4 than 3. However, it's not close to 5, suggesting there's room for improvement to reach high satisfaction levels.

Boxplot Distribution:

- The mean salaries for each satisfaction level are: 102473 (1.0), 103311 (2.0), 103738 (3.0), 95485 (3.872555924877030 87), 103178 (4.0), and 100647.(5.0)

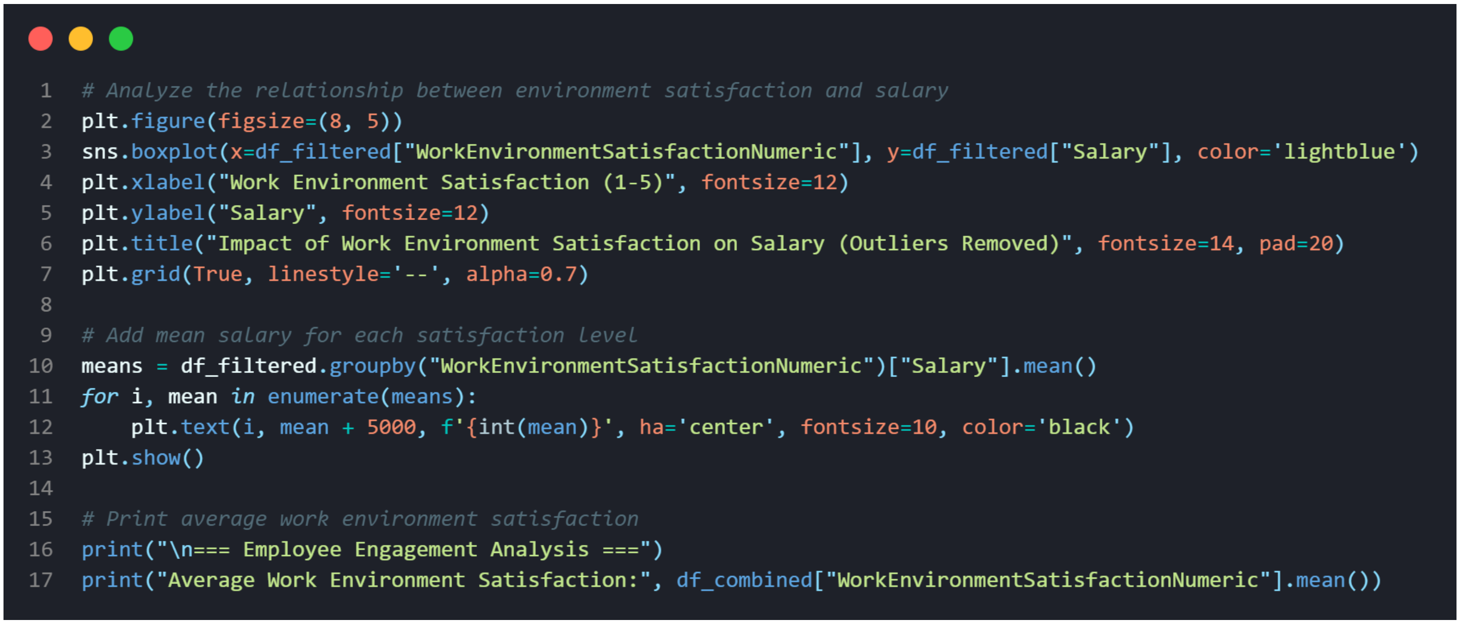
- There's no clear trend showing that higher satisfaction leads to higher salaries. For example, the highest mean salary (103738) is at satisfaction level 3.0, while the lowest (95485) is at the average satisfaction level (3.87). This suggests that satisfaction doesn't strongly correlate with salary in this dataset.

- The boxplot shows some variability, with outliers (dots above the boxes) indicating a few employees with significantly higher salaries across all satisfaction levels.

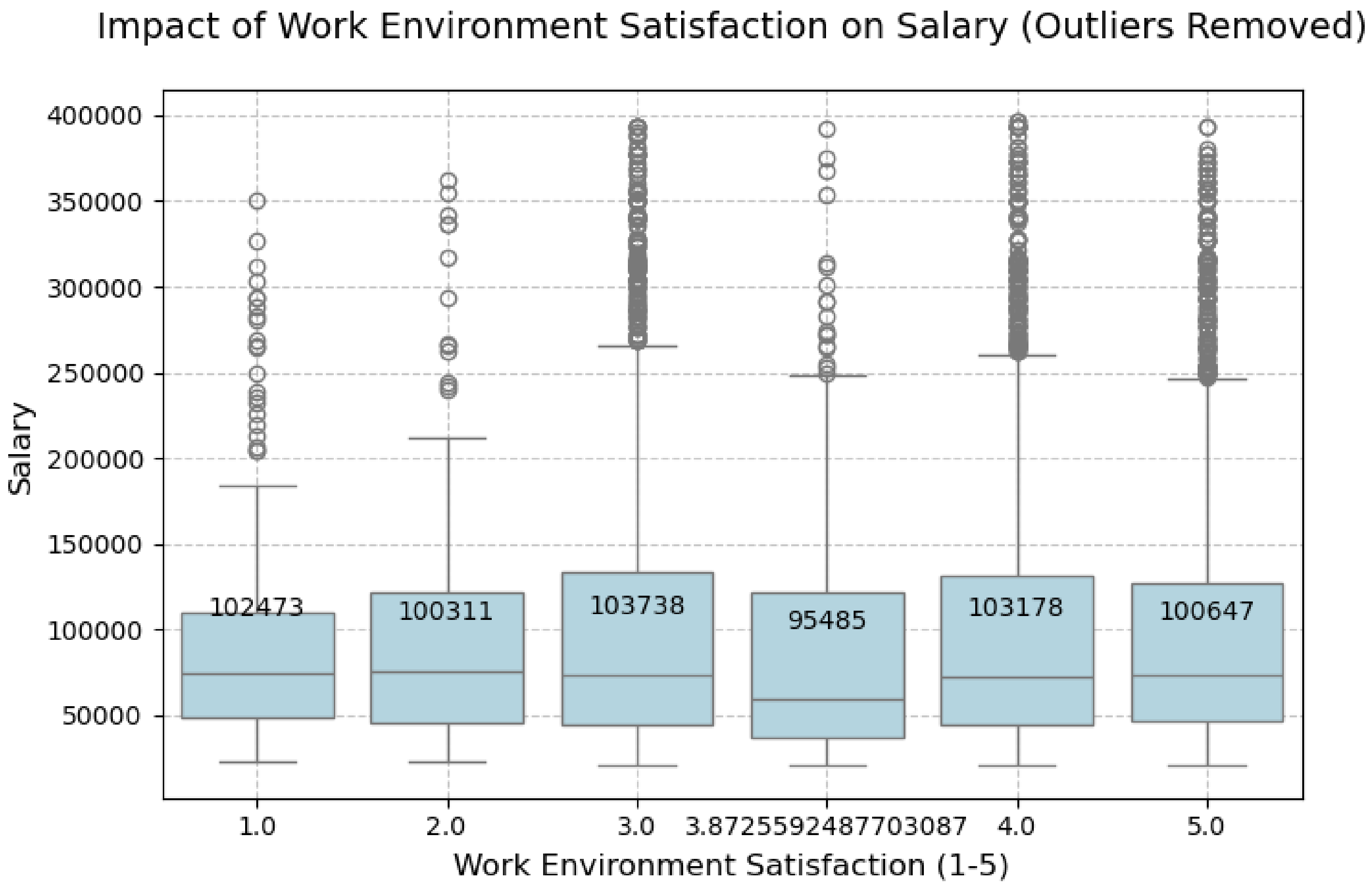
Implications:

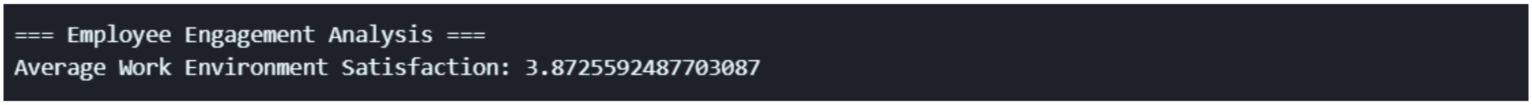
- With an average satisfaction of 3.87, the work environment is decent but not exceptional. The company could aim for a score closer to 4.5 or 5 by improving workplace conditions (e.g., better facilities, more flexible policies) or culture (e.g., recognition programs).

- The lack of a clear salary-satisfaction trend suggests that salary might not be the primary driver of satisfaction, or other factors (e.g., job role, manager support) might be more influential.



```python
# Analyze the relationship between environment satisfaction and salary
plt.figure(figsize=(8, 5))
sns.boxplot(x=df_filtered["WorkEnvironmentSatisfactionNumeric"], y=df_filtered["Salary"], color='lightblue')
plt.xlabel("Work Environment Satisfaction (1-5)", fontsize=12)
plt.ylabel("Salary", fontsize=12)
plt.title("Impact of Work Environment Satisfaction on Salary (Outliers Removed)", fontsize=14, pad=20)
plt.grid(True, linestyle='--', alpha=0.7)

# Add mean salary for each satisfaction level
means = df_filtered.groupby("WorkEnvironmentSatisfactionNumeric")["Salary"].mean()
for i, mean in enumerate(means):
    plt.text(i, mean + 5000, f'{int(mean)}', ha='center', fontsize=10, color='black')
plt.show()

# Print average work environment satisfaction
print("\n=== Employee Engagement Analysis ===")
print("Average Work Environment Satisfaction:", df_combined["WorkEnvironmentSatisfactionNumeric"].mean())
```





```
=== Employee Engagement Analysis ===
Average Work Environment Satisfaction: 3.8725592487703087
```

2: Correlation Analysis

Objective

This analysis examines the relationships between key employee features (`Age`, `Salary`, `YearsAtCompany`, `WorkEnvironmentSatisfactionNumeric`) to understand how they correlate with each other. It helps identify patterns, such as whether older employees tend to have higher salaries or longer tenures.

Steps in the Code

- **Correlation Calculation:** Computes the correlation matrix for the selected features.
- **Visualization:**
  - Plots a heatmap (`sns.heatmap`) to visualize the correlations, with values annotated and a color scale (`coolwarm`).
- **Output:** Prints the correlation matrix.

Interpretation:

Correlation Matrix:

- **Age and YearsAtCompany:** 0.642532 (strong positive correlation). Older employees tend to have longer tenures, which makes sense as they've likely been with the company longer.

- **Age and Salary**: 0.424894 (moderate positive correlation). Older employees generally have higher salaries, possibly due to experience or seniority.

- **YearsAtCompany and Salary**: 0.212381 (weak positive correlation). Longer tenure is slightly associated with higher salaries, but the relationship is not strong

WorkEnvironmentSatisfactionNumeric:

- **With `Age`:** 0.000745 (negligible correlation).

- With `Salary`: -0.011273 (negligible negative correlation).

- With `YearsAtCompany`: 0.009379 (negligible correlation).

- These values indicate that work environment satisfaction has almost no linear relationship with age, salary, or tenure.

Heatmap :

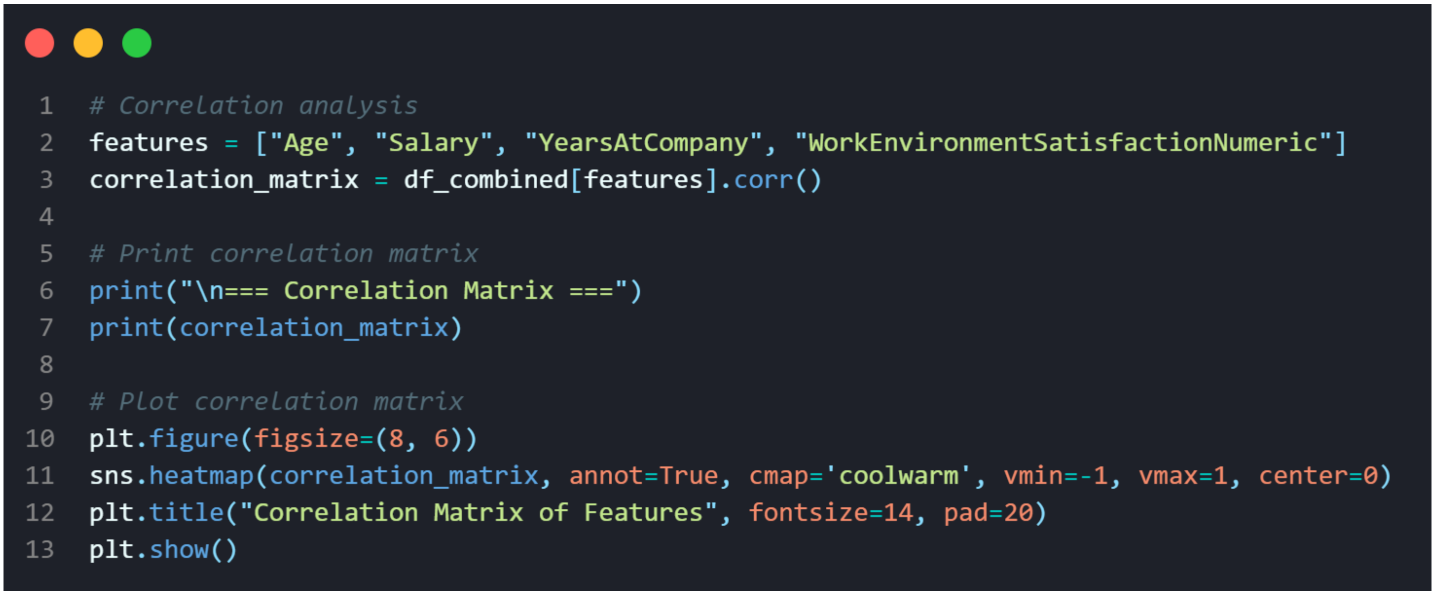
- The heatmap visually confirms the above: strong correlations (e.g., 0.64) are in darker shades (red), while weak or no correlations (e.g., 0.000745) are in lighter shades (gray).

Implications:

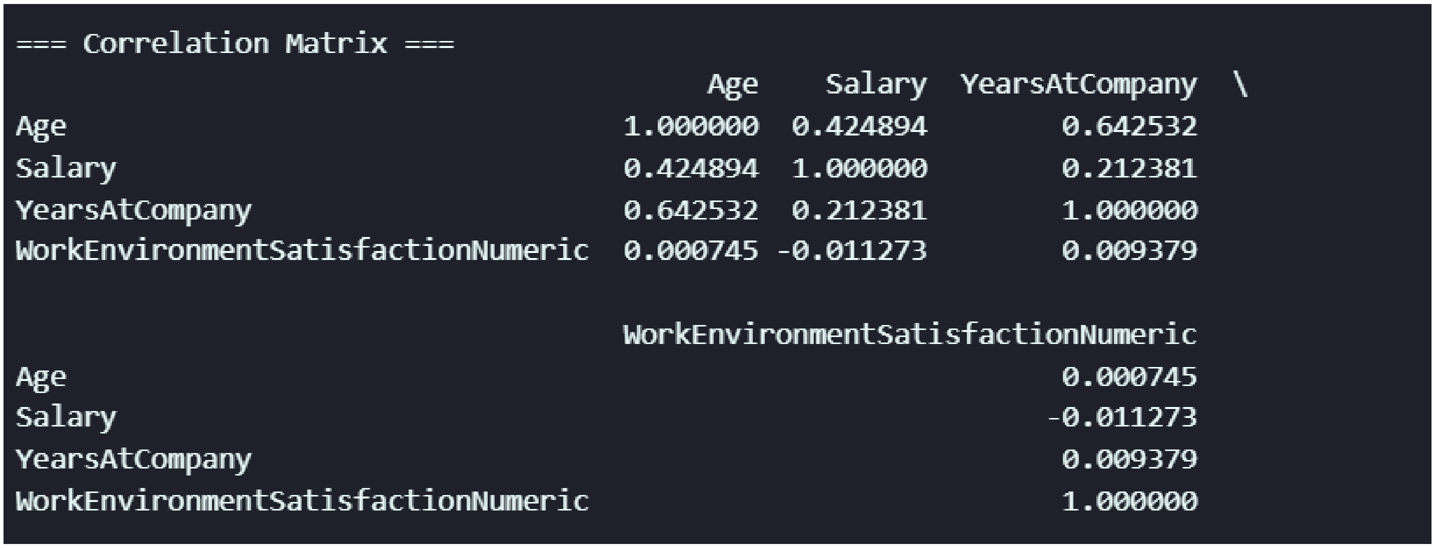
- The strong correlation between `Age` and `YearsAtCompany` suggests that retention strategies might focus on younger employees, as older ones are already staying longer.

- The weak correlation between `Salary` and `YearsAtCompany` indicates that long tenure doesn't guarantee higher pay, which might frustrate employees expecting raises over time.

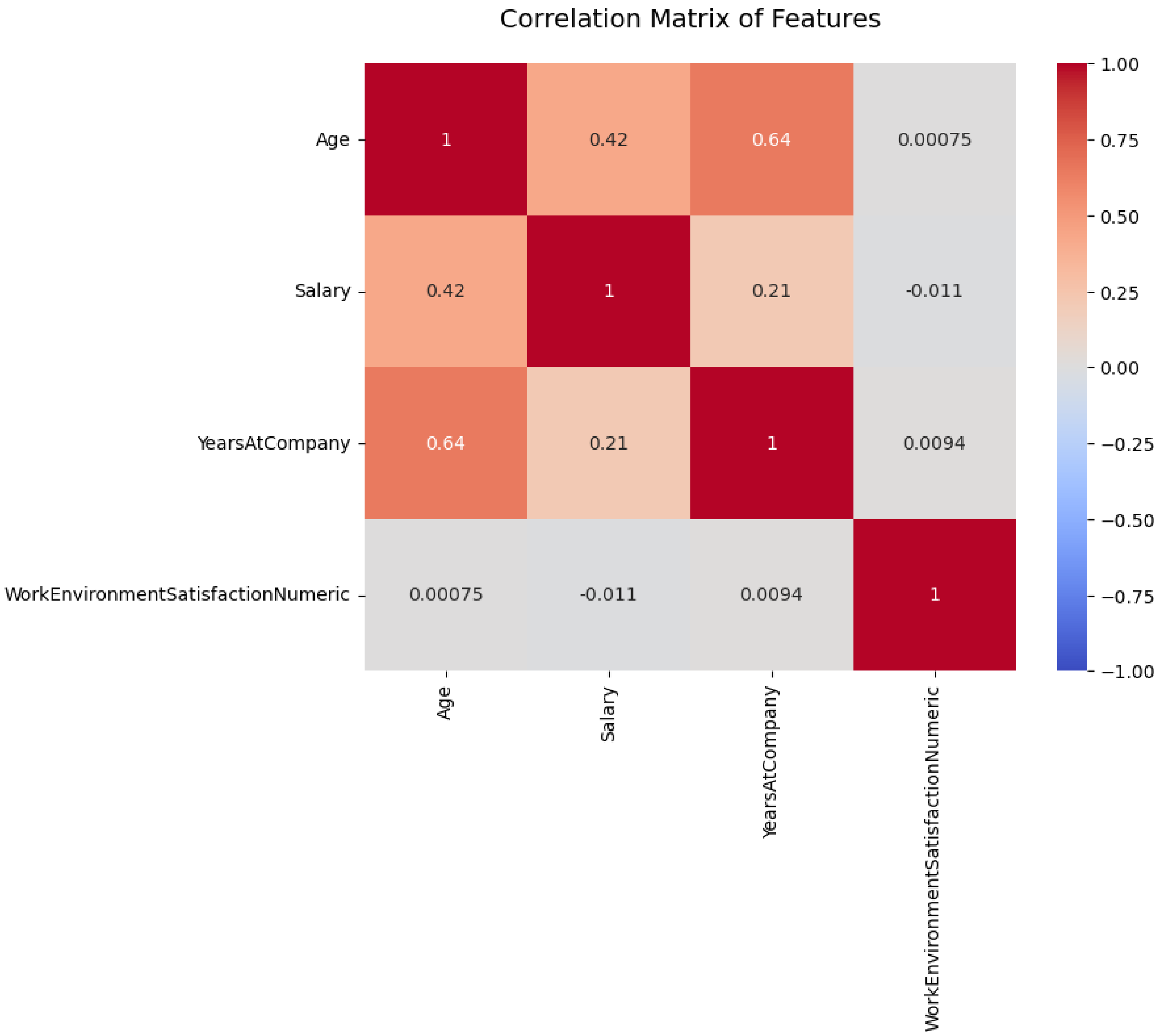
- The lack of correlation between `WorkEnvironmentSatisfactionNumeric` and other features suggests that satisfaction is independent of age, salary, or tenure, and might be influenced by other factors (e.g., management, work-life balance).



```python
# Correlation analysis
features = ["Age", "Salary", "YearsAtCompany", "WorkEnvironmentSatisfactionNumeric"]
correlation_matrix = df_combined[features].corr()

# Print correlation matrix
print("\n=== Correlation Matrix ===")
print(correlation_matrix)

# Plot correlation matrix
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1, center=0)
plt.title("Correlation Matrix of Features", fontsize=14, pad=20)
plt.show()
```



```
=== Correlation Matrix ===
                                      Age    Salary  YearsAtCompany  \
Age                              1.000000  0.424894        0.642532
Salary                           0.424894  1.000000        0.212381
YearsAtCompany                   0.642532  0.212381        1.000000
WorkEnvironmentSatisfactionNumeric 0.000745 -0.011273        0.009379

                                  WorkEnvironmentSatisfactionNumeric
Age                                                        0.000745
Salary                                                    -0.011273
YearsAtCompany                                             0.009379
WorkEnvironmentSatisfactionNumeric                         1.000000
```



3: Predictive Analytics

Objective:

This analysis uses machine learning to predict employee attrition (`Attrition`) and identify key influencing factors. It answers questions about who might leave and what drives turnover, providing insights for retention and recruitment strategies.

Steps in the Code:

Feature Engineering :

- Defines features: `Age`, `Salary`, `YearsAtCompany`, `WorkEnvironmentSatisfactionNumeric`, `OverTime`, `DistanceFromHome.`

- Ensures `Attrition` is numeric.(1/0)

- Drops rows with missing values in these columns.

Model Training :

- Splits data into training (80%) and testing (20%) sets (`train_test_split`).

- Trains a RandomForestClassifier with 100 trees and balanced class weights to handle imbalanced data.

Evaluation :

- Predicts `Attrition` on the test set.

- Calculates accuracy (0.9826086956521739) and generates a classification report.

- Extracts and sorts feature importances (`YearsAtCompany`: 0.409201, `Salary`: 0.305103, `Age`: 0.219830, `WorkEnvironmentSatisfactionNumeric`: 0.0655866).

Visualization :

- Plots a bar chart of feature importances.

Output :

- Prints accuracy, classification report, feature importances, and target distribution (`Attrition`: 0: 0.672271, 1: 0.327729).

Interpretation:

Attrition Prediction:

Accuracy: 0.9826086956521739 (98.26%) is extremely high, indicating excellent predictive power. However, this might suggest overfitting, especially given the imbalanced dataset (see target distribution below).

Classification Report:

- **No Attrition (0):** Precision 1.00, Recall 0.98, F1-score 0.99, Support 943.
- **Attrition (1):** Precision 0.95, Recall 1.00, F1-score 0.97, Support 437.
- The model is highly accurate for both classes, but the perfect recall for "Attrition" (1.00) and near-perfect precision for "No Attrition" (1.00) further suggest potential overfitting or an overly optimistic evaluation.

Macro Avg: 0.97 (precision), 0.99 (recall), 0.98 (F1-score).

Weighted Avg: 0.98 across all metrics, reflecting the overall performance.

Feature Importances:

YearsAtCompany: 0.409201 (most important). Longer tenure significantly influences whether an employee leaves, possibly due to stagnation or lack of growth opportunities.

Salary: 0.305103 (second most important). Pay is a key factor in attrition, likely because low salaries drive employees to seek better opportunities.

Age: 0.219830. Age plays a moderate role, possibly with younger employees being more likely to leave for new opportunities.

- **WorkEnvironmentSatisfactionNumeric**: 0.0655866 (least important among shown features). Satisfaction has a minor impact on attrition in this model, which aligns with its weak correlations in Analysis 2.

Target Distribution:

`Attrition` : 0 (No): 67.23%, 1 (Yes): 32.77%. The dataset is imbalanced, with more employees staying than leaving. This imbalance might inflate the accuracy, as the model could over-predict "No Attrition" to achieve high accuracy.

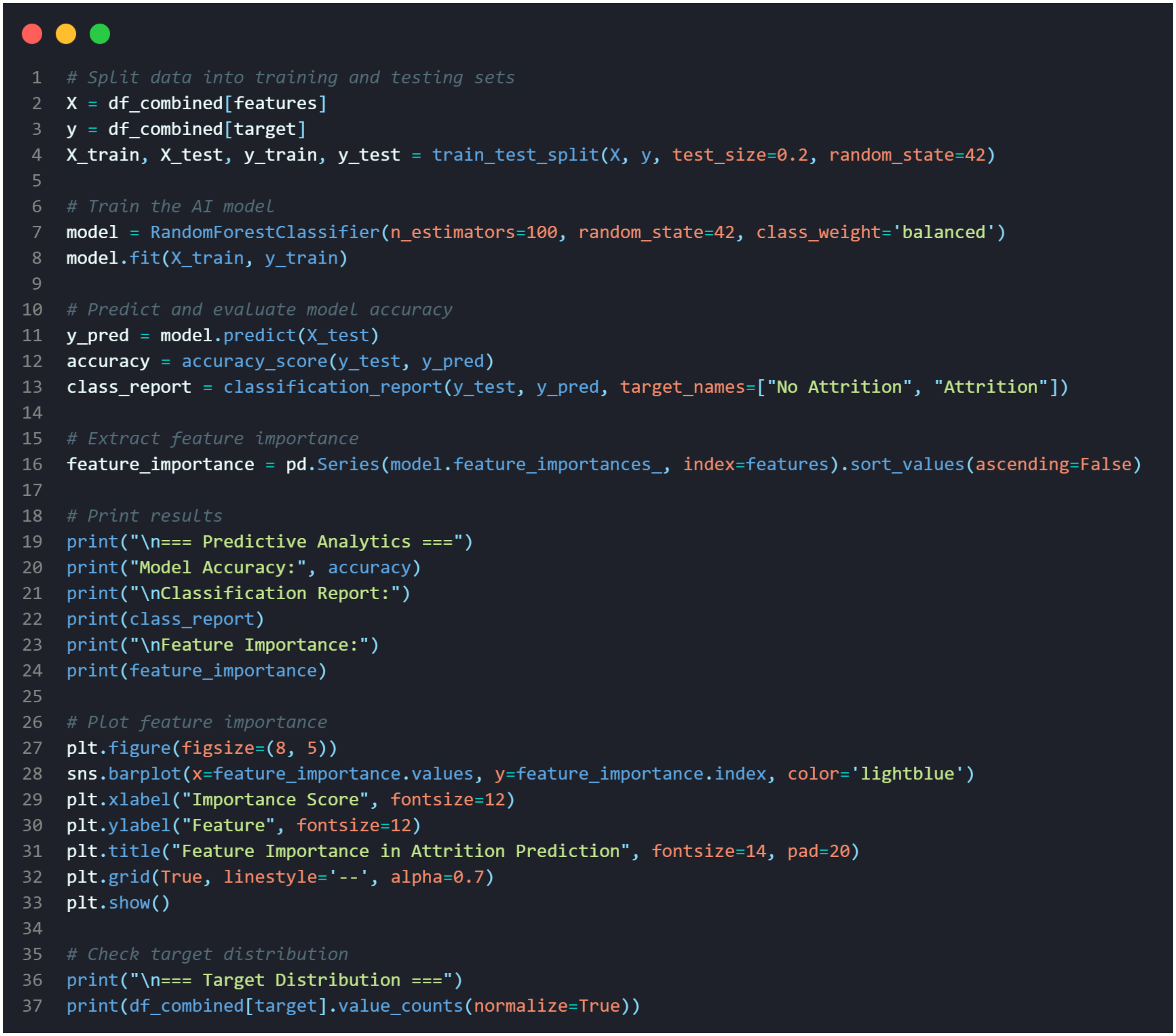
-Feature Importance Bar Chart:

- The bar chart visually confirms that `YearsAtCompany` has the highest importance (around 0.41), followed by `Salary` (0.31), `Age` (0.22), and `WorkEnvironmentSatisfactionNumeric.(0.07) `

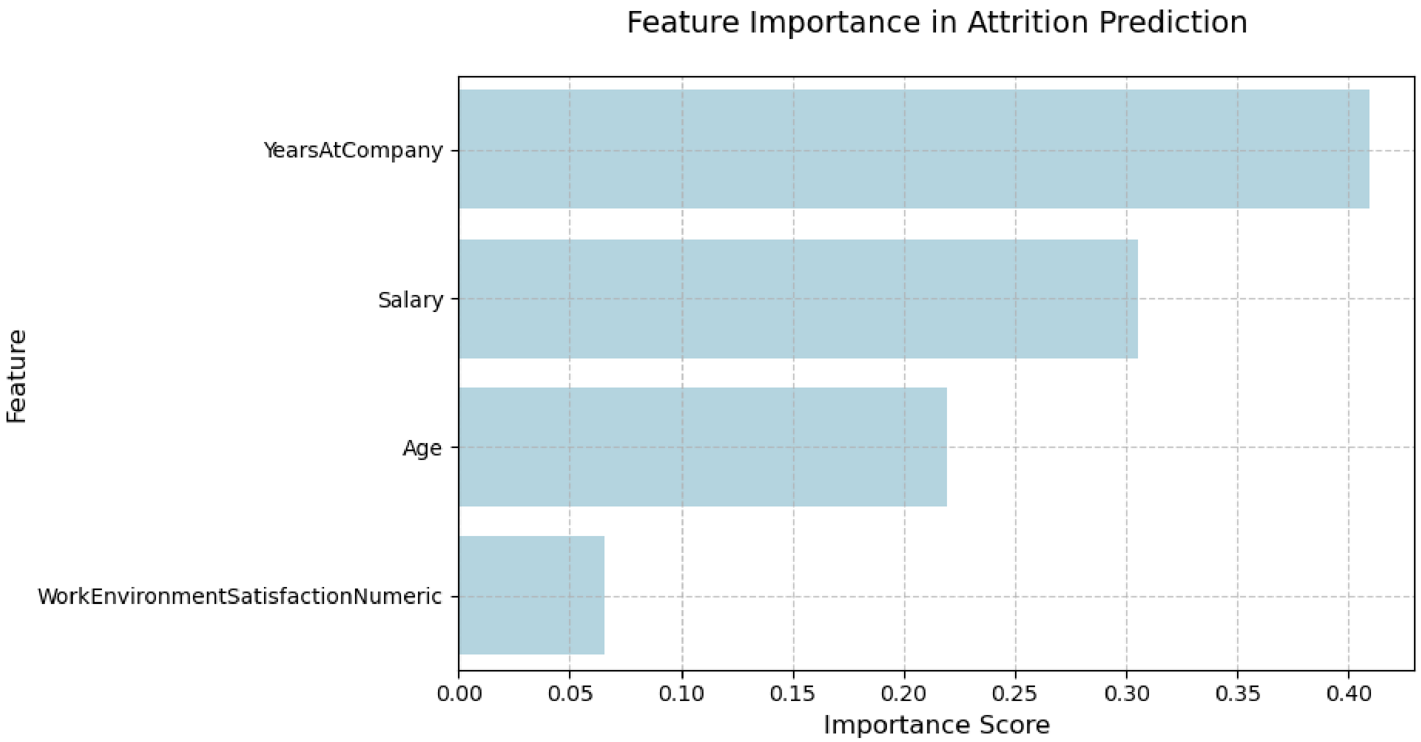
-Recruitment Strategy Improvements:

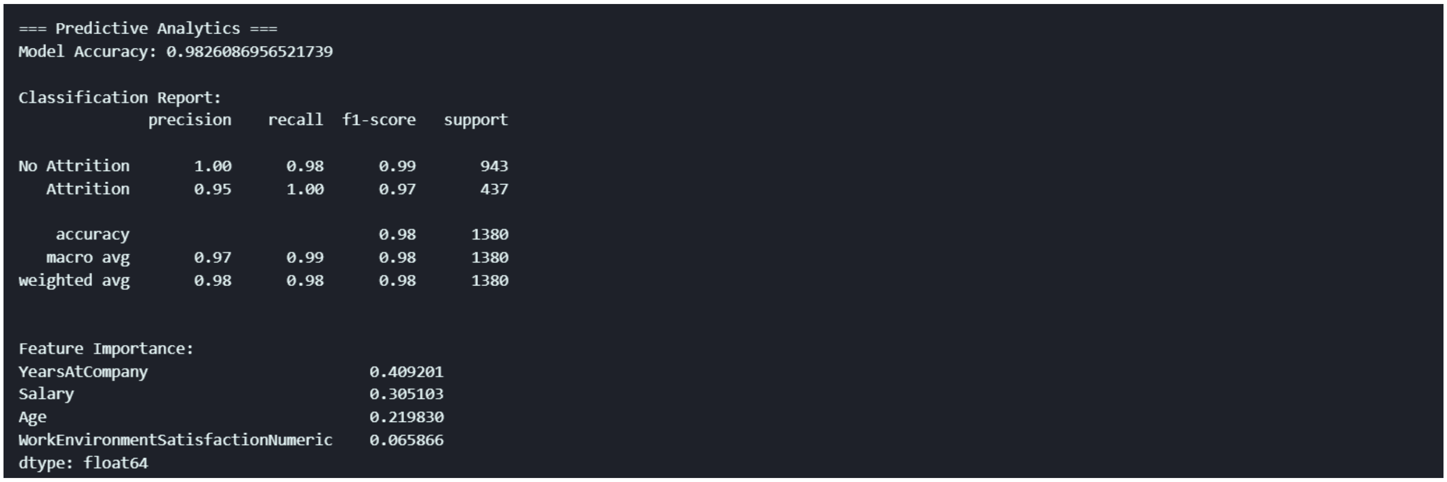
- **From feature importances:** Focus on candidates who value stability (`YearsAtCompany` is key) and offer competitive salaries (`Salary` matters). Younger employees (`Age`) might need more engagement to stay.

- **From target distribution:** Since 32.77% of employees leave, retention efforts should target the at-risk group (e.g., those with long tenures but low salary growth).

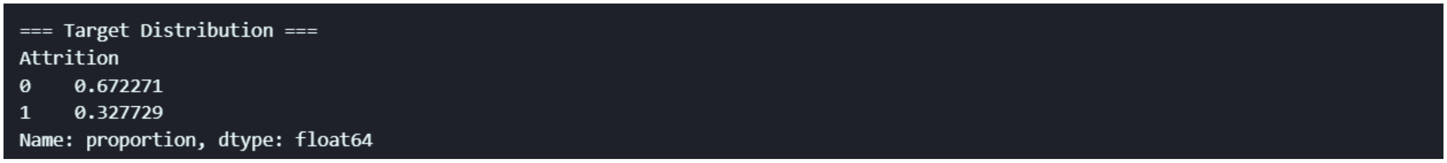


```python
# Split data into training and testing sets
X = df_combined[features]
y = df_combined[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the AI model
model = RandomForestClassifier(n_estimators=100, random_state=42, class_weight='balanced')
model.fit(X_train, y_train)

# Predict and evaluate model accuracy
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
class_report = classification_report(y_test, y_pred, target_names=["No Attrition", "Attrition"])

# Extract feature importance
feature_importance = pd.Series(model.feature_importances_, index=features).sort_values(ascending=False)

# Print results
print("\n=== Predictive Analytics ===")
print("Model Accuracy:", accuracy)
print("\nClassification Report:")
print(class_report)
print("\nFeature Importance:")
print(feature_importance)

# Plot feature importance
plt.figure(figsize=(8, 5))
sns.barplot(x=feature_importance.values, y=feature_importance.index, color='lightblue')
plt.xlabel("Importance Score", fontsize=12)
plt.ylabel("Feature", fontsize=12)
plt.title("Feature Importance in Attrition Prediction", fontsize=14, pad=20)
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()

# Check target distribution
print("\n=== Target Distribution ===")
print(df_combined[target].value_counts(normalize=True))
```





```
=== Predictive Analytics ===
Model Accuracy: 0.9826086956521739

Classification Report:
              precision    recall  f1-score   support

No Attrition       1.00      0.98      0.99       943
   Attrition       0.95      1.00      0.97       437

    accuracy                           0.98      1380
   macro avg       0.97      0.99      0.98      1380
weighted avg       0.98      0.98      0.98      1380


Feature Importance:
YearsAtCompany                        0.409201
Salary                                0.305103
Age                                   0.219830
WorkEnvironmentSatisfactionNumeric    0.065866
dtype: float64
```



```
=== Target Distribution ===
Attrition
0    0.672271
1    0.327729
Name: proportion, dtype: float64
```

Conclusion

What the Script Delivers:

-Employee Engagement Analysis:

Assesses satisfaction via `WorkEnvironmentSatisfactionNumeric` (average 3.87) and its link to `Salary`. The lack of a clear trend between satisfaction and salary suggests other factors (e.g., job role, management) might drive satisfaction. The company should aim to boost satisfaction to 4.5+ through better workplace conditions or culture.

-Correlation Analysis:

Shows that `Age` and `YearsAtCompany` are strongly correlated (0.64), while `WorkEnvironmentSatisfactionNumeric` has negligible correlations with other features, indicating it's influenced by factors not captured here.

-Predictive Analytics:

Predicts `Attrition` with high accuracy (98.26%), identifying `YearsAtCompany`, `Salary`, and `Age` as key drivers. However, the high accuracy may indicate overfitting due to the imbalanced dataset (67.23% stay, 32.77% leave). Retention strategies should focus on employees with long tenures, competitive pay, and younger staff.

DataDynamos