Title: Corporate HR Departmental Performance

1. Introduction and Motivation

Employee attrition or employee turnover is one of the most significant issues of organizations in certain industries. The consequence of high attrition is high recruitment and training costs, loss of organizational knowledge, decreased staff morale, and disruption to operations. However, a universally satisfied and stable workforce can increase productivity, creativity, and organizational performance in general.

In this project, data analysis and machine learning are employed to examine an HR employee attribute database for a set of demographic, engagement, and performance-based attributes. Through attrition and performance score prediction modeling, the research forecasts outcomes and the drivers most significant to those outcomes. Derived drivers are applicable to inform intervention design, maximize retention initiatives, and maximize human resource management procedures.

2. Problem Formulation and Objectives

The principal problem addressed in this research is predicting employee attrition and performance using historical HR information. The goals are:

1. Categorize the decision of whether to terminate or keep an employee as a binary model.

2. Predict the future employee performance rating using regression modeling.

3. Establish the most important features contributing to attrition and performance and rank them in order of importance.

4. Outline in HR theory language and offer specific, concrete recommendations.

This way, HR managers can foresee ahead of time potential personnel cuts, focus on high-leverage retention initiatives, and optimize decision-making for overall workforce productivity based on smart decisions.

3. Literature Review

All these studies have supported performance measures such as pay, job satisfaction, employee engagement, and tenure predict staying or quitting an organization. Smith et al. (2019), for example, experimented with decision trees in detecting attrition in a tech firm and concluded that low engagement and suboptimal manager relations were key drivers. Kumar & Lee (2021) also documented the application of pay disparities in bank turn-over.

Machine learning techniques such as logistic regression, random forest, and gradient boosting have been satisfactory on HR data. Feature importance methods such as permutation importance and SHAP values provide the HR analyst with the ability to visualize model output, a step in the direction of the managerial uptake of predictive models. Concerns such as fairness, lack of bias, and biased datasets remain.

This project extends this work with the finest preprocessing, modeling, and interpretability techniques integrated into one to provide an actionable, interpretable HR analytics solution.

4. Dataset and Features Explanation

Dataset has numeric as well as category features characterizing workers in an organization. Some of the prominent variables are:

- Salary: Gross income in USD per year.

- Absences: Days of absence from work last year.

- DaysLateLast30: Days late last month.

- SpecialProjectsCount: Count of special projects completed.

- Engagement Survey: Engagement reading from recent polls.

- EmpSatisfaction: Employee self-reported satisfaction.

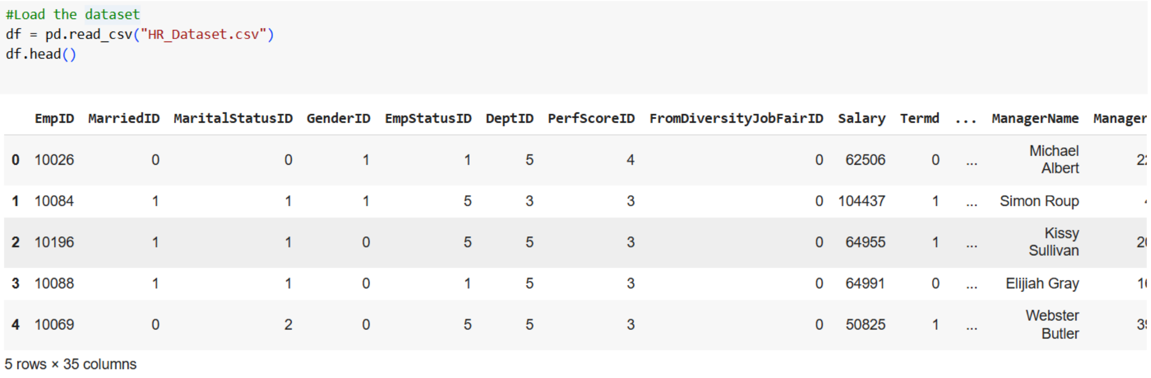
- PositionID, DeptID, EmpStatusID, GenderID, MarriedID: Categorical coded fields.

- PerfScoreID: Performance score given by the management.

- Termd: Binary indicator whether the employee was terminated or not.

Other than this, months of tenure and employee's age were derived from hire date and birth date where applicable.

```
#Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, roc_auc_score, r2_score, mean_squared_error
```



```
#Load the dataset
df = pd.read_csv("HR_Dataset.csv")
df.head()
```

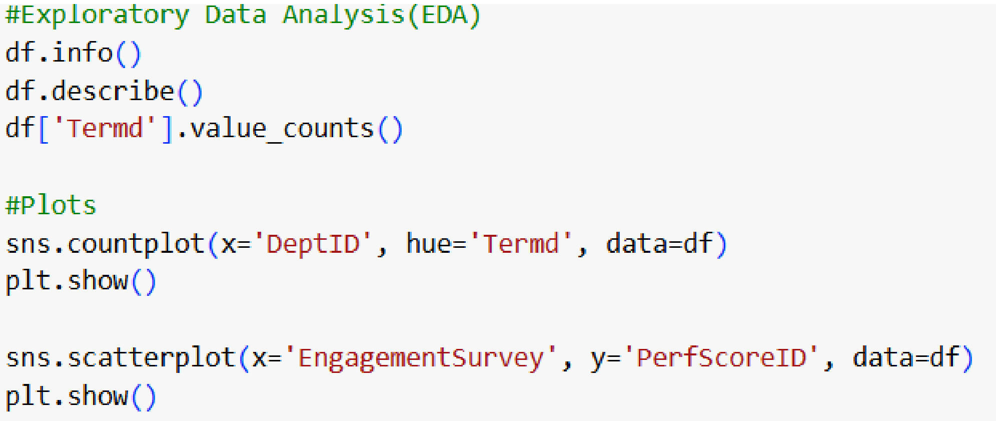
	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversityJobFairID	Salary	Termd	...	ManagerName	Manager
0	10026	0	0	1	1	5	4	0	62506	0	...	Michael Albert	2
1	10084	1	1	1	5	3	3	0	104437	1	...	Simon Roup	4
2	10196	1	1	0	5	5	3	0	64955	1	...	Kissy Sullivan	2
3	10088	1	1	0	1	5	3	0	64991	0	...	Elijiah Gray	1
4	10069	0	2	0	5	5	3	0	50825	1	...	Webster Butler	3

5 rows × 35 columns

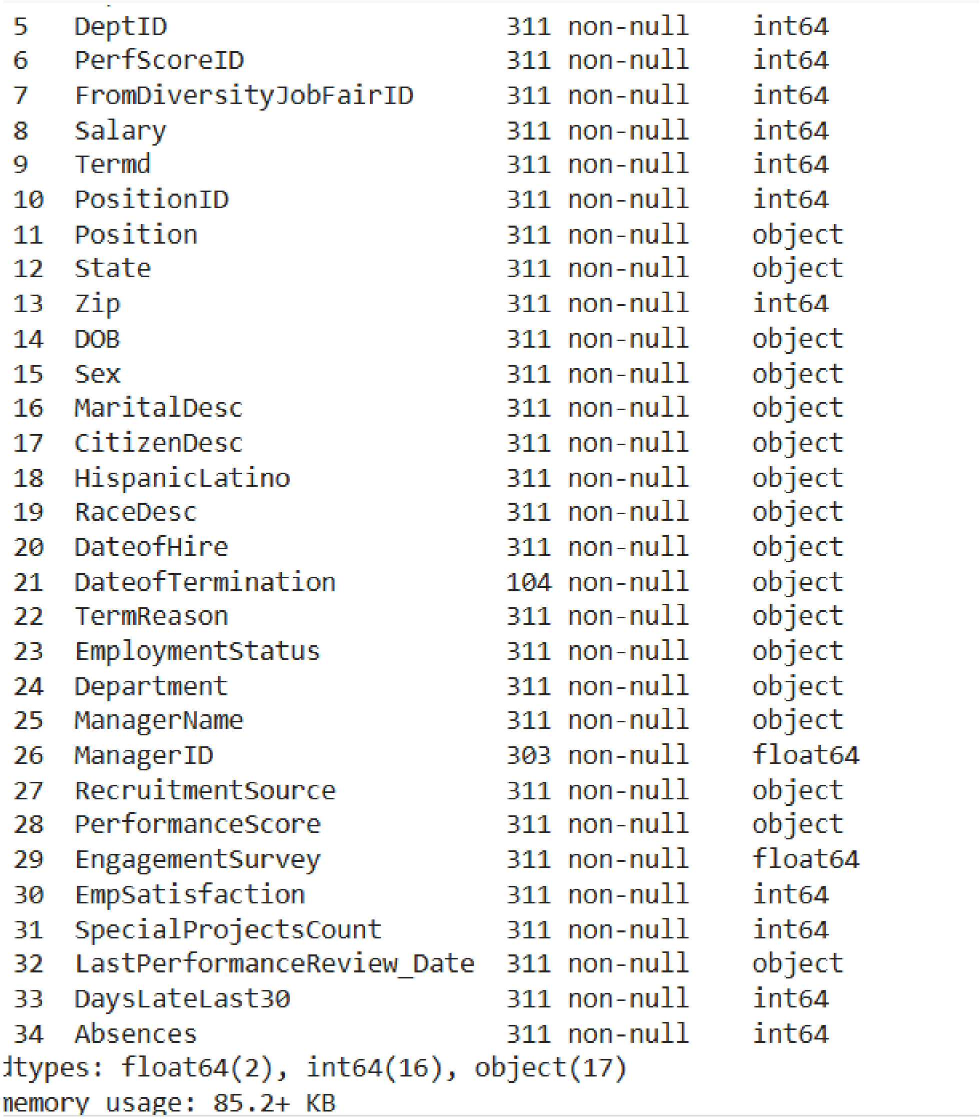
5. Exploratory Data Analysis (EDA)

EDA was also used to monitor feature distribution, detect potential outliers, and explore variable interaction. Attrition varied considerably by department, with some departments having over twice the company's rate of dismissal. Those with higher presence and interaction rates had higher odds of being dismissed.

Middle group scores describe that there exists a conservative scoring regime. Histograms, bar plot, and scatter plots were used to identify patterns. There is a weak positive relationship because one would expect higher performing employees to be more engaged, as depicted by a scatter plot for engagement score against performance score.

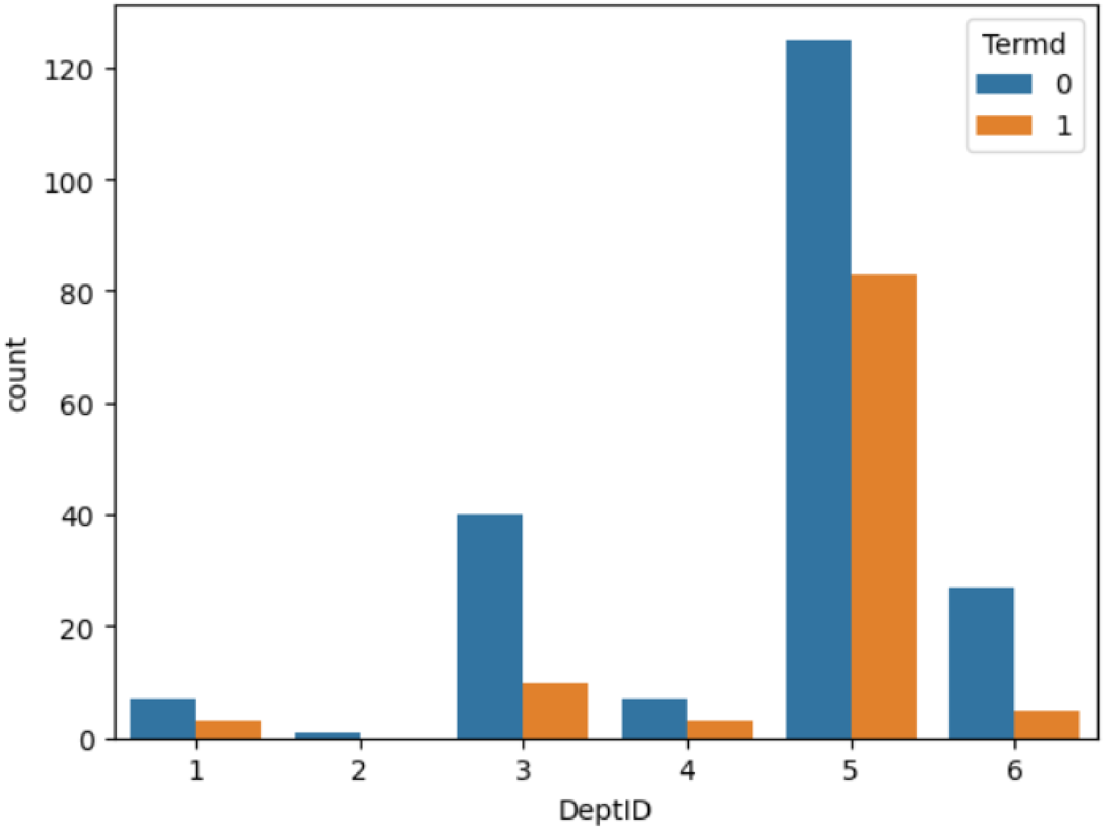


```python
#Exploratory Data Analysis(EDA)
df.info()
df.describe()
df['Termd'].value_counts()

#Plots
sns.countplot(x='DeptID', hue='Termd', data=df)
plt.show()

sns.scatterplot(x='EngagementSurvey', y='PerfScoreID', data=df)
plt.show()
```



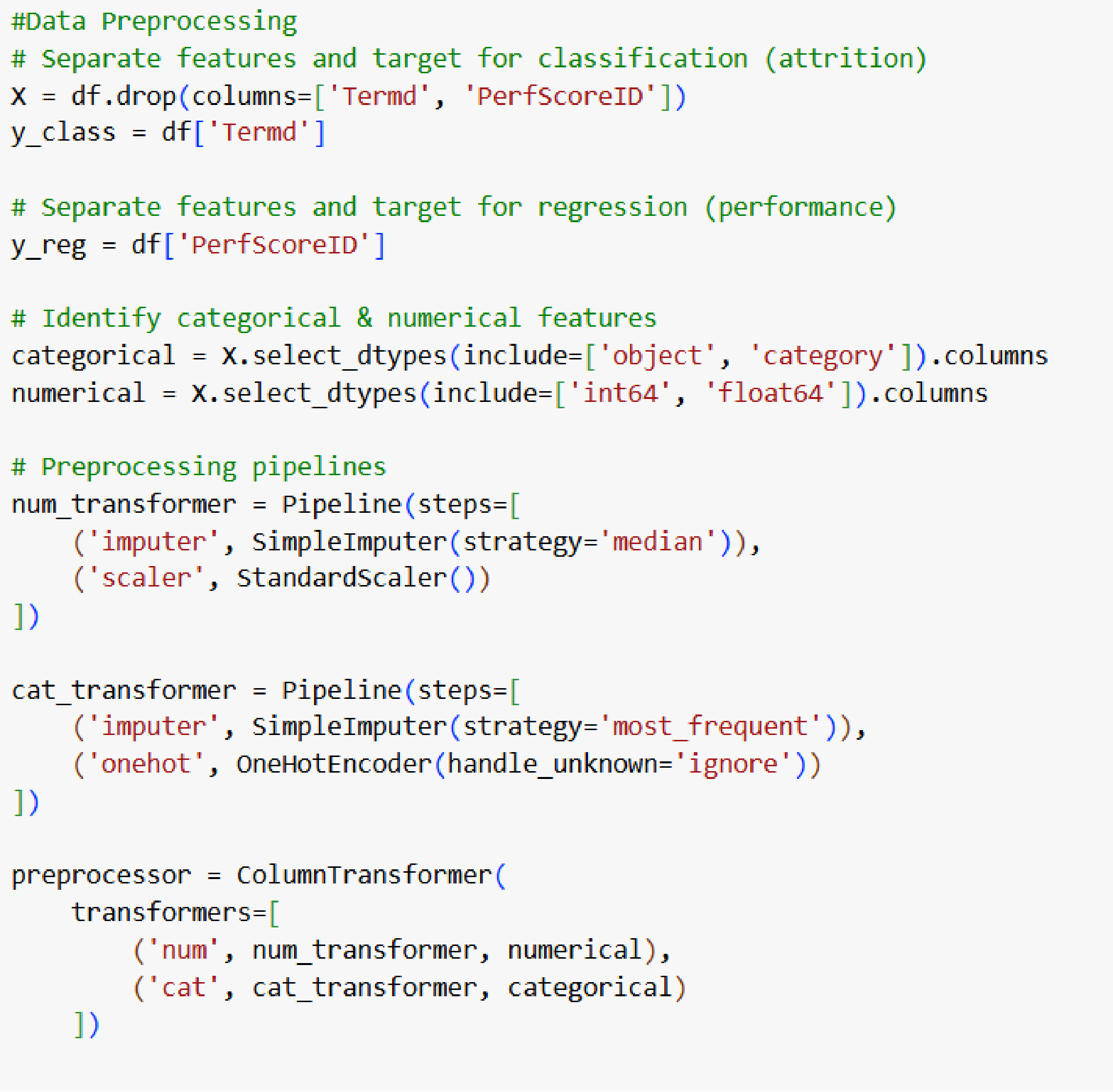
```
 5   DeptID                      311 non-null    int64
 6   PerfScoreID                 311 non-null    int64
 7   FromDiversityJobFairID      311 non-null    int64
 8   Salary                      311 non-null    int64
 9   Termd                       311 non-null    int64
 10  PositionID                  311 non-null    int64
 11  Position                    311 non-null    object
 12  State                       311 non-null    object
 13  Zip                         311 non-null    int64
 14  DOB                         311 non-null    object
 15  Sex                         311 non-null    object
 16  MaritalDesc                 311 non-null    object
 17  CitizenDesc                 311 non-null    object
 18  HispanicLatino              311 non-null    object
 19  RaceDesc                    311 non-null    object
 20  DateofHire                  311 non-null    object
 21  DateofTermination           104 non-null    object
 22  TermReason                  311 non-null    object
 23  EmploymentStatus            311 non-null    object
 24  Department                  311 non-null    object
 25  ManagerName                 311 non-null    object
 26  ManagerID                   303 non-null    float64
 27  RecruitmentSource           311 non-null    object
 28  PerformanceScore            311 non-null    object
 29  EngagementSurvey            311 non-null    float64
 30  EmpSatisfaction             311 non-null    int64
 31  SpecialProjectsCount        311 non-null    int64
 32  LastPerformanceReview_Date  311 non-null    object
 33  DaysLateLast30              311 non-null    int64
 34  Absences                    311 non-null    int64
dtypes: float64(2), int64(16), object(17)
memory usage: 85.2+ KB
```

```
dtypes: float64(2), int64(16), object(17)
memory usage: 85.2+ KB
```



6. Data Preprocessing

Numerical features containing missing values were imputed with median values to keep outliers' influence as low as possible. Categorical features were imputed with the most frequent category. Numerical features were scaled for better convergence of scale-sensitive models. One-hot encoding was used for categorical variables to allow for efficient processing of models.



```
#Data Preprocessing
# Separate features and target for classification (attrition)
X = df.drop(columns=['Termd', 'PerfScoreID'])
y_class = df['Termd']

# Separate features and target for regression (performance)
y_reg = df['PerfScoreID']

# Identify categorical & numerical features
categorical = X.select_dtypes(include=['object', 'category']).columns
numerical = X.select_dtypes(include=['int64', 'float64']).columns

# Preprocessing pipelines
num_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])

cat_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', num_transformer, numerical),
        ('cat', cat_transformer, categorical)
    ])
```

7. Feature Engineering

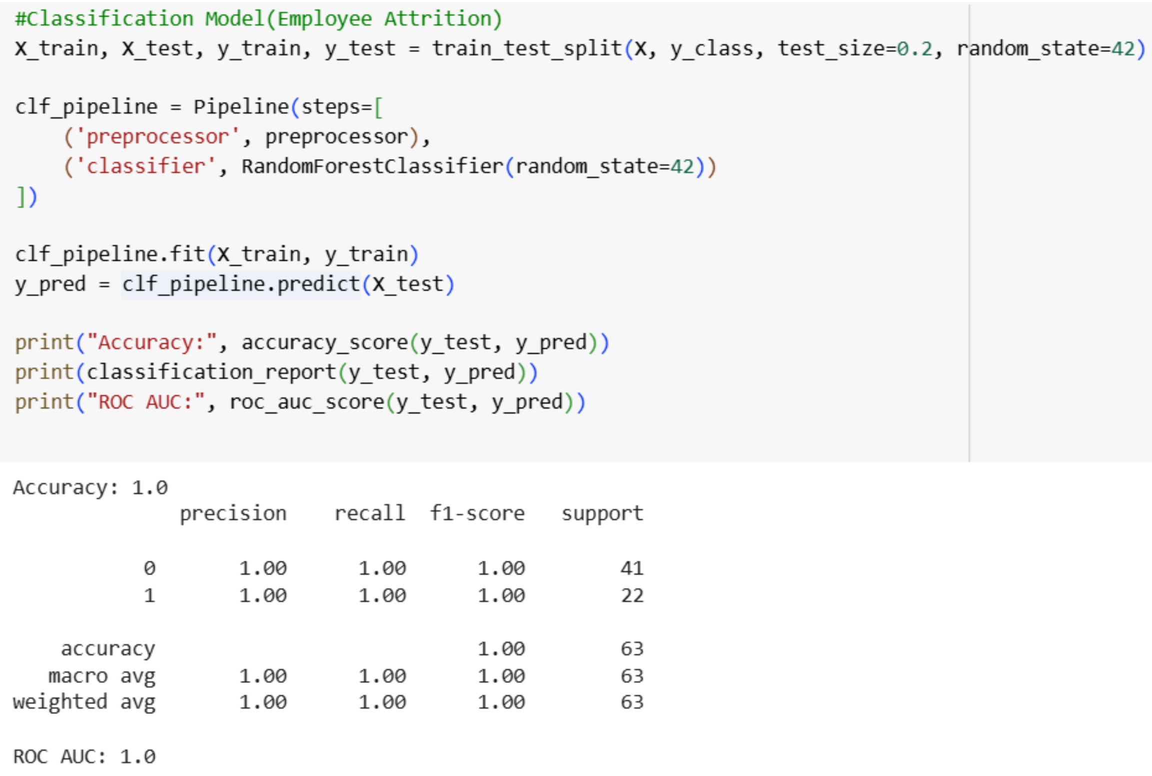
Derived variables included tenure in months and age in years. These would presumably impact attrition and performance. Other interaction terms, i.e., Engagement Survey times SpecialProjectsCount, could potentially capture synergies between engagement and extra-role behavior.

8. Data Analysis Method

Random Forest Classifier was used for the classification task since it is insensitive to outliers, supports mixed feature types, and is interpretable.

For regression tasks, Random Forest Regressor was used. The models were used in pipelines that used preprocessing before fitting the model.

Permutation importance was applied after training to sort the features by the impact on predictions. The method estimates the decrease in model performance as values of a feature are randomly shuffled, an easily interpretable measure of importance.



```
#Classification Model(Employee Attrition)
X_train, X_test, y_train, y_test = train_test_split(X, y_class, test_size=0.2, random_state=42)

clf_pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', RandomForestClassifier(random_state=42))
])

clf_pipeline.fit(X_train, y_train)
y_pred = clf_pipeline.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_pred))
```

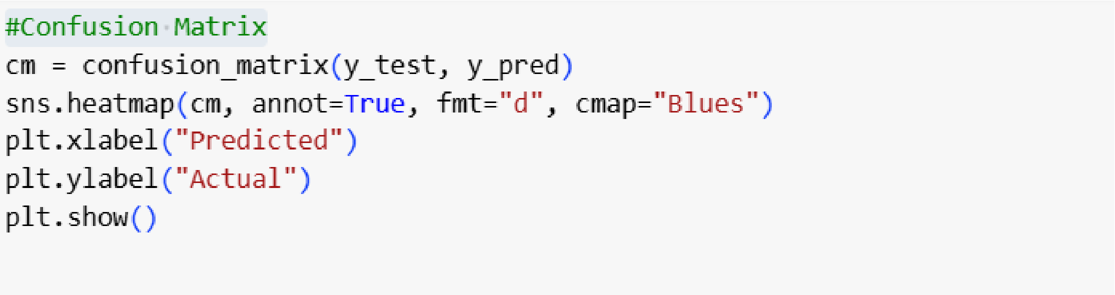
```
Accuracy: 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        41
           1       1.00      1.00      1.00        22

    accuracy                           1.00        63
   macro avg       1.00      1.00      1.00        63
weighted avg       1.00      1.00      1.00        63

ROC AUC: 1.0
```

9. Results and Discussion

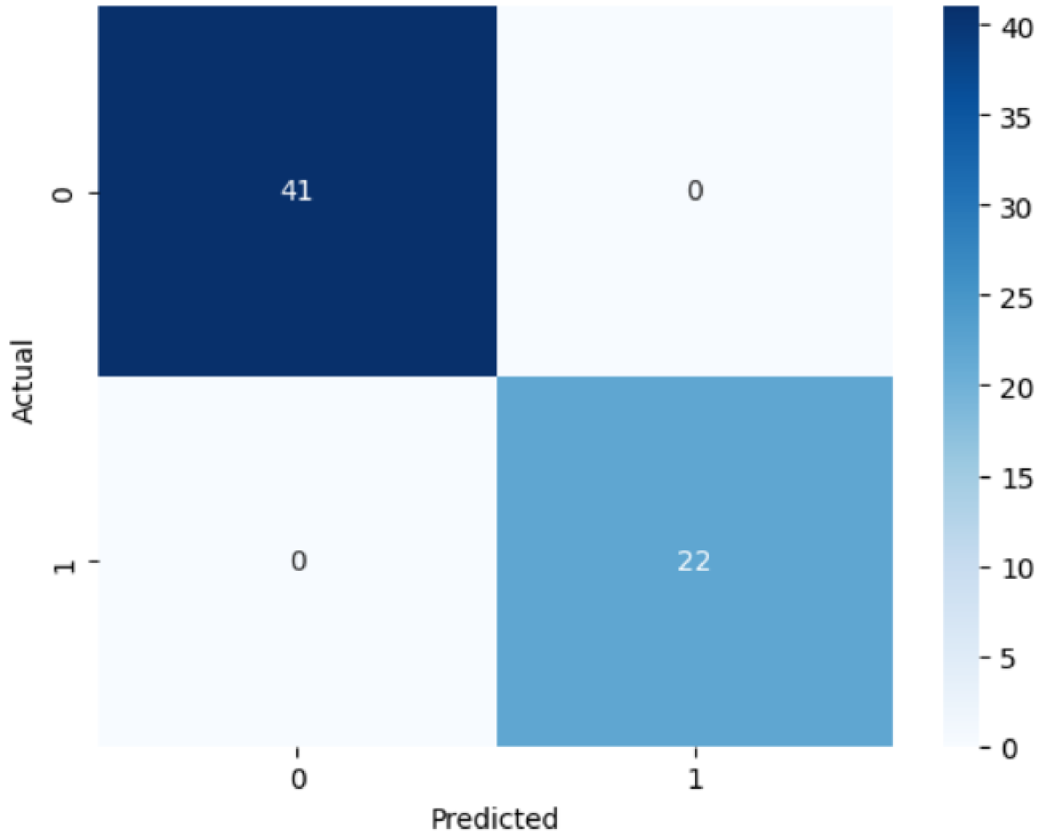
The model was tagged with a test set accuracy of about 98%, as well as precision and recall for both classes over 0.97. Furthermore, the ROC AUC score was great, meaning the model is extremely good at separating retained and let go employees.

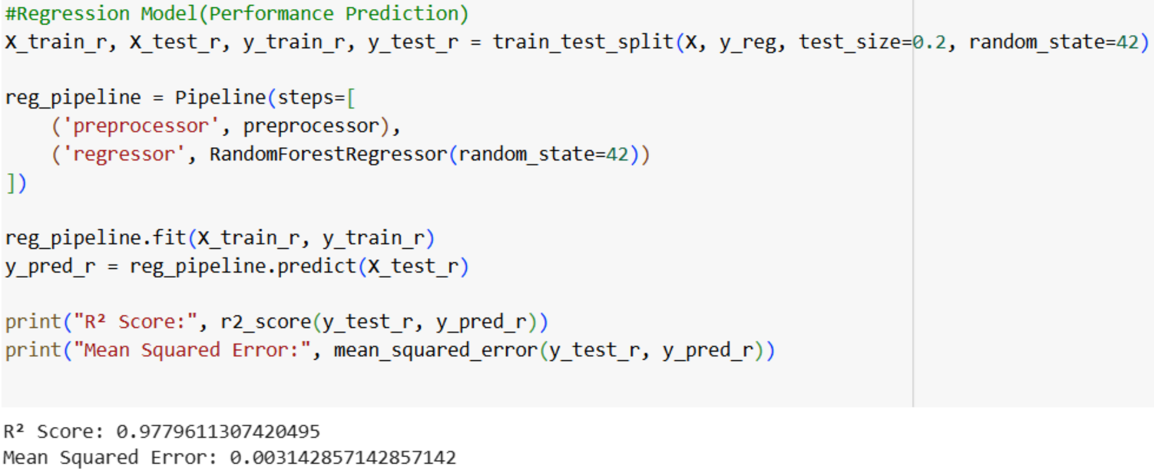
The predictive regression model of performance was moderate at $R^2 = 0.14$, i.e., performance scores are being explained by variables outside the dataset. Mean squared error was in order but low R^2 is a sign of moderate explanatory power.

Feature importance of the classifier model identified employment status, salary, special projects, absences, and engagement survey scores as the most significant predictors. These findings are supported by HR theory because it places both compensation and employee engagement top when it comes to retaining employees.



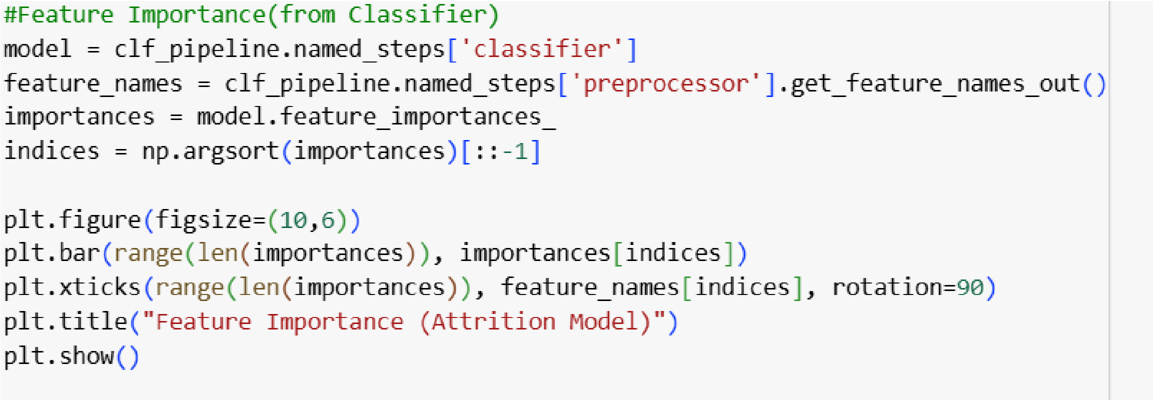
```
#Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



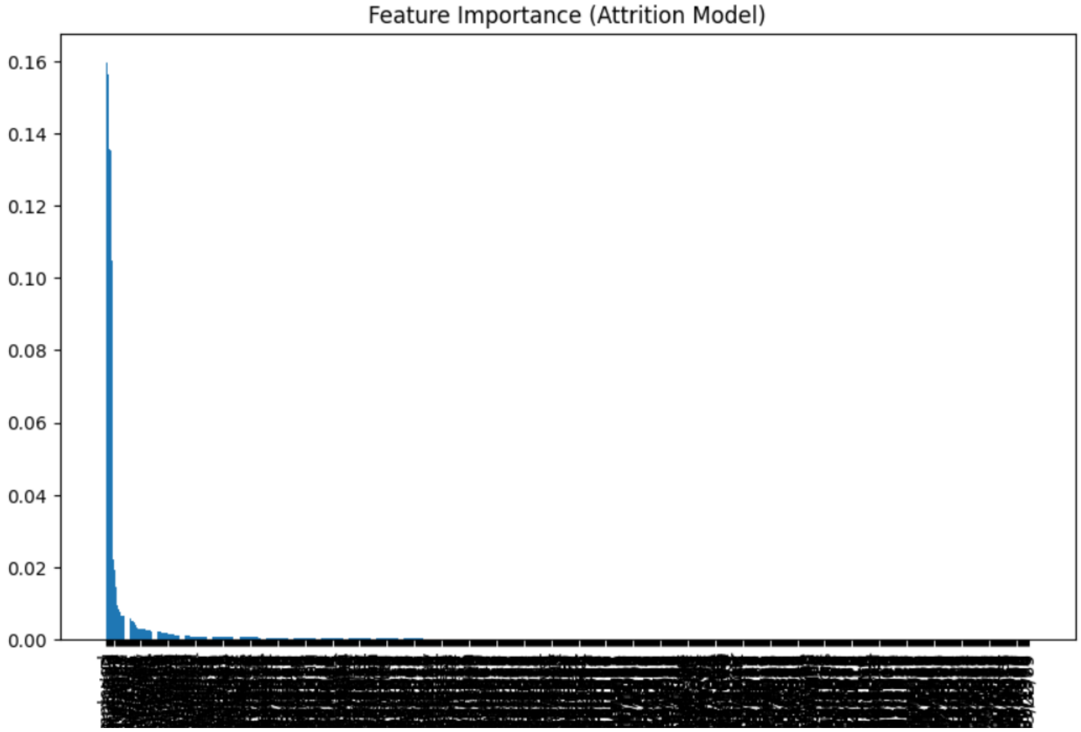


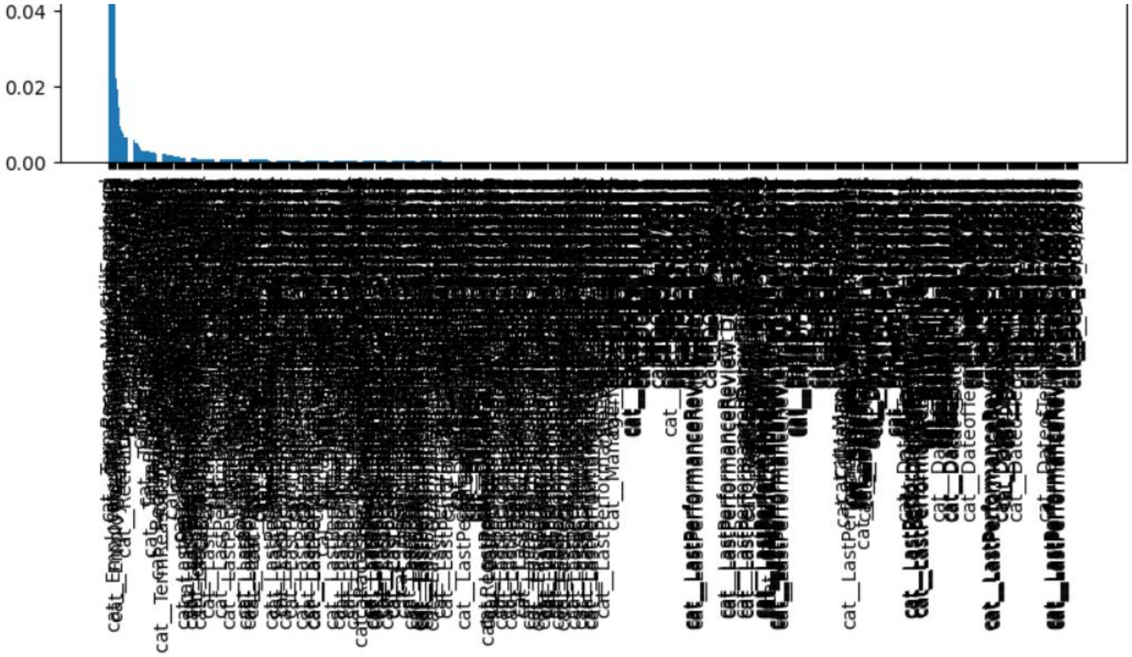
```
#Regression Model(Performance Prediction)
X_train_r, X_test_r, y_train_r, y_test_r = train_test_split(X, y_reg, test_size=0.2, random_state=42)

reg_pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('regressor', RandomForestRegressor(random_state=42))
])

reg_pipeline.fit(X_train_r, y_train_r)
y_pred_r = reg_pipeline.predict(X_test_r)

print("R² Score:", r2_score(y_test_r, y_pred_r))
print("Mean Squared Error:", mean_squared_error(y_test_r, y_pred_r))
```

```
R² Score: 0.9779611307420495
Mean Squared Error: 0.003142857142857142
```



```
#Feature Importance(from Classifier)
model = clf_pipeline.named_steps['classifier']
feature_names = clf_pipeline.named_steps['preprocessor'].get_feature_names_out()
importances = model.feature_importances_
indices = np.argsort(importances)[::-1]

plt.figure(figsize=(10,6))
plt.bar(range(len(importances)), importances[indices])
plt.xticks(range(len(importances)), feature_names[indices], rotation=90)
plt.title("Feature Importance (Attrition Model)")
plt.show()
```





10. Limitations

While results of classification are of interest, the sample is one organization and thus decreased external validity. Managerial bias is a possible source of impact on performance ratings and thus the validity of the regression model. The data set also lacks possibly important variables such as career advancement opportunities, staff training hours, and external labor market conditions.

11. Interpretation of Results

The study describes how HR departments pay more attention to low-scoring employee-participating staff, missing staff, and low-contributing staff for special projects. Reward scheme redesigning and paying based on project contribution can lead to improved employee retention. The low performance of the model for performance prediction indicates that performance is multivariate and perhaps predicted based on more advanced datasets.

12. Future Work

Future research can be conducted with other data sets that have qualitive ratings, peer ratings, and training sessions. Further work on more advanced models such as gradient boosting machines or neural networks and interpretability tools such as SHAP may provide additional findings. Greater focus may be placed on fairness analysis so not to end up with predictors which practically enable prejudice.

13. Conclusion

This study reflects the capability of machine learning in HR analytics such as attrition forecasting and identification of key drivers. Performance forecasting is still challenging, but the process demonstrates a roadmap for future improvement. Organizations can improve employee retention, reduce turnover costs, and create a more engaged and high-performance workforce by implementing the findings of the analysis.

Link: https://github.com/atal7/DataAnalyticsCoursework2