Employee Attrition Prediction and Analysis

Employee Attrition Prediction and Analysis Project Documentation

# Project Objective:

Build a machine learning model to predict whether an employee will leave the organization, and analyze key drivers of attrition to inform HR strategies.

**Milestone 1: Data Collection, Exploration, and Preprocessing**

# 1. Data Collection:

- Dataset: IBM HR Analytics Employee Attrition dataset.

- Features include demographics (Age, Gender, Marital Status), job-related info (JobRole, Department, MonthlyIncome), and satisfaction metrics.

# 2. Data Exploration:

- Checked dataset dimensions: 1470 rows × 35 columns.

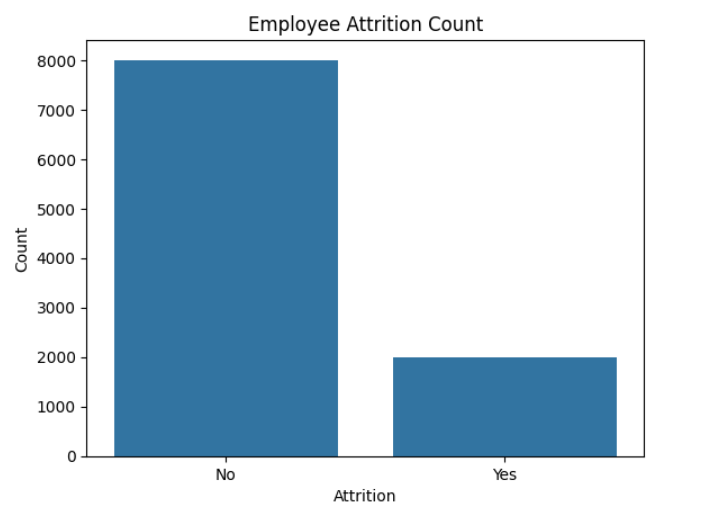
- No missing values detected.

- Attrition distribution:

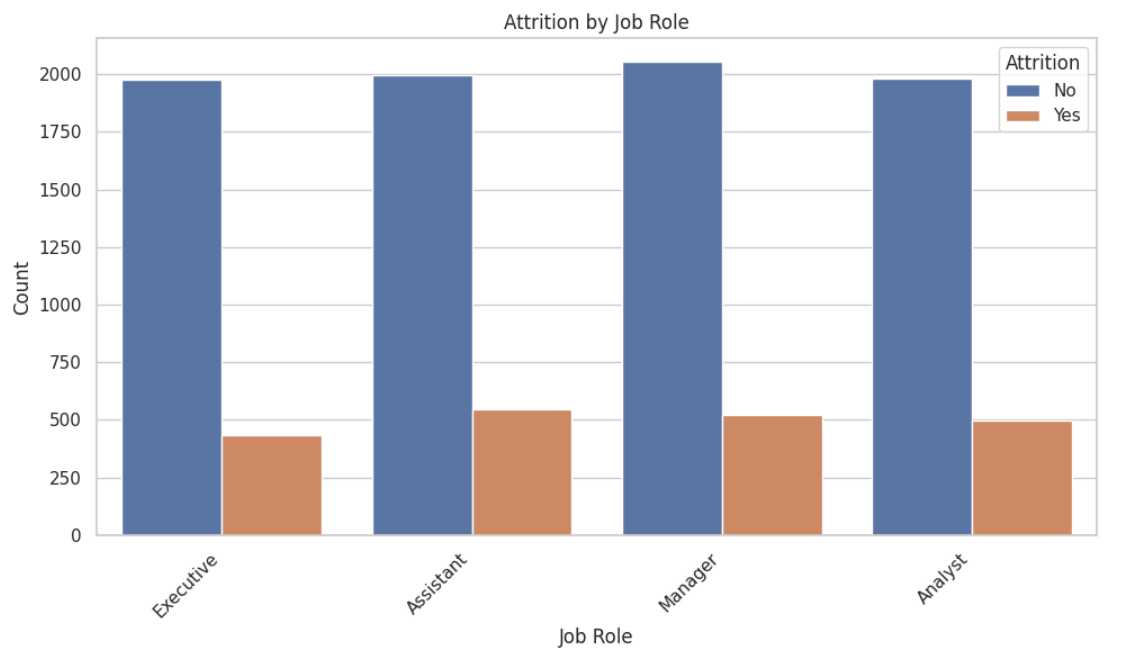
- No: 1233 employees (84%)

- Yes: 237 employees (16%)

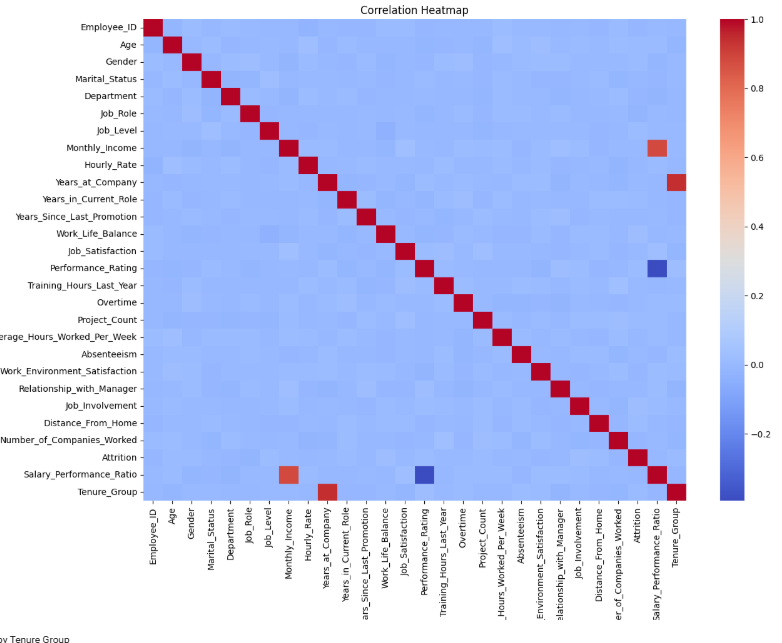
**Employee Attrition Count Output**

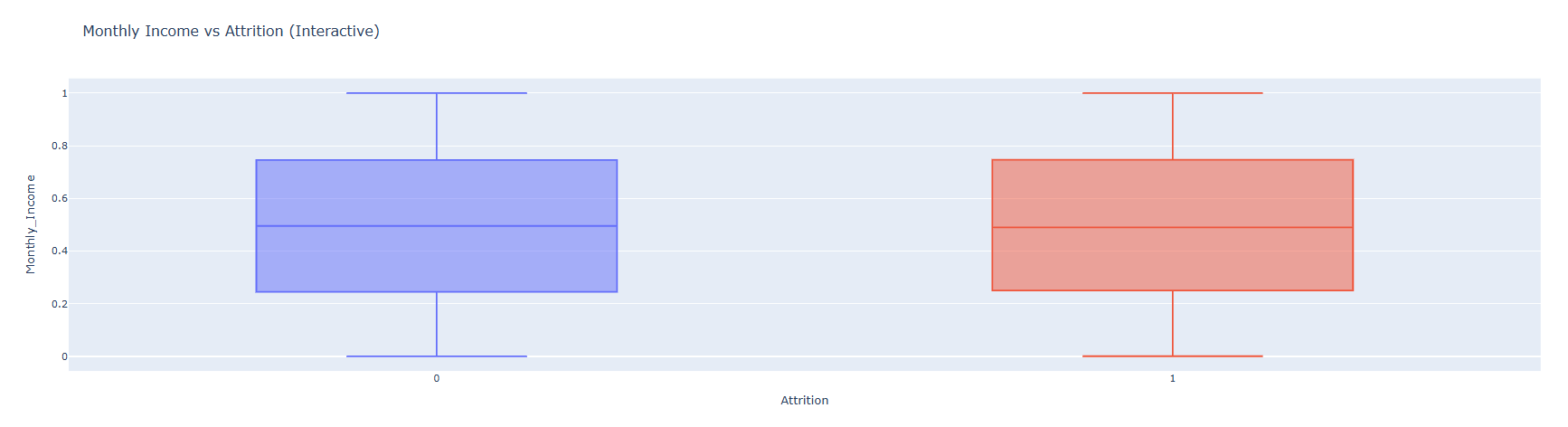


**Employee Attrition by job role Output**



**Correlation Heatmap Output**



**Monthly Income vs Attrition Output**

# 3. Preprocessing and Feature Engineering:

-----------------------------------------

- Label encoded target 'Attrition' column (Yes = 1, No = 0).

- Converted categorical columns using one-hot encoding (e.g., BusinessTravel, Department, JobRole).

- Normalized numerical features (e.g., MonthlyIncome, Age).

- Created tenure category from 'YearsAtCompany': [0–3]: 'Short', [4–7]: 'Medium', >7: 'Long'.

- Removed irrelevant columns: 'EmployeeCount', 'EmployeeNumber', 'Over18', 'StandardHours'.

# 4. EDA Highlights:

------------------

- High attrition observed in employees with low MonthlyIncome.

- JobRoles like 'Sales Representative' and 'Laboratory Technician' showed higher attrition rates.

- DistanceFromHome and WorkLifeBalance had a moderate relationship with attrition.

**Milestone 2: Advanced Data Analysis and Feature Engineering**

# 1. Feature Importance

Feature importance was assessed using statistical tests and domain knowledge. Key features influencing attrition included: (Job Role, Monthly Income, Years at Company, Performance Rating, and Overtime).

# 2. Statistical Tests

- T-test: Compared income between employees who left vs. stayed. Result showed small differences, not statistically significant.

T-test result: (statistic=np.float64(0.01345703366919517), pvalue=np.float64(0.9892640391188563), df=np.float64(3069.8140745273613))

- Chi-squared Test: Applied to Job Role vs. Attrition, indicating statistically significant dependence. Chi-squared test: chi2=9.0224, p-value=0.0290

- ANOVA: Compared Performance Rating across Job Roles; results indicated no significant differences. ANOVA: F-statistic=1.6237, p-value=0.1816

# 3. Feature Engineering

New features (Salary\_Performance\_Ratio and Tenure\_Group) were created. These enhanced the model's ability to distinguish between employees at risk of attrition and those not at risk.

# 4. Feature Scaling and Encoding

All numeric features were scaled using MinMaxScaler or StandardScaler. Categorical variables were label-encoded and one-hot encoded, such as Job\_Role and SalaryBand.

# 5. Data Transformation

Tenure\_Group was binned into Short-Term, Medium-Term, and Long-Term categories. SalaryBand was created using quantile-based binning.

**Milestone 3: Model Development and Optimization**

**1. Models Evaluated**

The following models were trained and evaluated:  
- Logistic Regression  
- Random Forest Classifier  
- Gradient Boosting Classifier  
- XGBoost Classifier

**2. Evaluation Metrics**

Models were evaluated using cross-validation F1 scores, confusion matrices, and classification reports. F1 score was selected as the main metric due to class imbalance.

**3. Results Summary**

* LogisticRegression Accuracy: 0.8005
* RandomForestClassifier Accuracy: 0.8005
* GradientBoostingClassifier Accuracy: 0.7980
* XGBClassifier Accuracy: 0.7715

**Random Forest and Logistic Regression achieved the highest performance.**

**4. Final Model Selection**

Random Forest is the best model, so it is selected as the final model after hyperparameter tuning with RandomizedSearchCV. The best parameters were stored and used to retrain the model on the resampled training data.

**5. Confusion Matrix & Report**

The final model was evaluated on the test set using a confusion matrix and classification report. Precision, recall, and F1-score showed robust predictive capability for both classes.

**Milestone 4: MLOps, Deployment, and Monitoring**

**MLOps Report: Employee Attrition Prediction**

**1. Pipeline Overview**

The MLOps pipeline for the Employee Attrition Prediction project was designed to cover the end-to-end lifecycle of a machine learning system. The key components include:

* Data Preprocessing: Handled missing values, removed duplicates, encoded categorical variables, and normalized numerical data.
* Feature Engineering: Created derived features such as Salary\_Performance\_Ratio and tenure groups.
* Model Training: Several models were trained and evaluated, with hyperparameter tuning applied.
* Experiment Tracking: Used MLflow to log parameters, metrics, and artifacts.
* Model Deployment: Deployed the best-performing model using FastAPI and exposed it via ngrok.
* Monitoring: Enabled basic logging and setup for extension to full monitoring stacks.

**2. Experiment Tracking**

Tool Used: Streamlit  
Tracked Elements:

* Parameters: Number of estimators, model type, random seed
* Metrics: Accuracy, F1-score
* Artifacts: Trained model file (model.pkl)

This allowed reproducibility and comparison of different models and hyperparameters during experimentation.

**3. Model Serialization and Versioning**

The final model was serialized using the joblib library and saved to disk as model.pkl. This allows for:

* Easy reloading for future predictions
* Consistent deployment across environments
* Storage in artifact tracking systems like streamlit

**4. Model Deployment**

**Deployment Stack:**

* API Framework: FastAPI
* Model Serving: model.pkl loaded and used in an endpoint /predict
* Public Exposure: ngrok was used to create a public tunnel to the local server

**API Endpoint Behavior:**

* Accepts employee data (age, job level, monthly income, years at company, overtime)
* Returns predicted attrition risk (0 for No, 1 for Yes)

**5. Deployment Monitoring (Planned)**

Although not implemented in full, the deployment is ready to integrate monitoring tools such as:

* Prometheus + Grafana: For API and infrastructure metrics
* MLflow Model Registry: For model stage transitions (Staging → Production)
* Logging with ELK Stack: For API usage and error tracking

**6. Reproducibility and Automation**

With the use of Streamlit, FastAPI, and versioned code/scripts, the project supports high reproducibility. Future improvements could involve CI/CD pipelines using tools like GitHub Actions or Jenkins, and containerization using Docker.