**Can We Predict Employee Attrition?**

A Machine Learning Approach Using the IBM HR Analytics Dataset

**Author:** Dimitrios Nanos  
**Program:** MSc in Data Science & Machine Learning  
**Date:** May 2025

Fig 1. Chat-GTP generated photo with the prompt “How do you understand employee attrition?”



**Abstract**

Employee attrition poses significant challenges to organizations, impacting productivity, team morale, and long-term strategic planning. In this project, I leverage the IBM HR Analytics fictional dataset to develop a machine learning model capable of predicting which employees are at risk of leaving the fictional company XYZ. After thorough data preprocessing, exploratory analysis, and model experimentation, the best-performing model was selected based on evaluation metrics such as Precision, Recall and F1 score. The final solution was deployed through a FastAPI application that allows HR professionals to input employee data and receive real-time predictions. Additionally, a sensitivity analysis was conducted to identify the most influential features affecting employee turnover. This project demonstrates how data-driven decision-making can support HR departments in identifying retention risks and proactively addressing employee concerns.

**1. Introduction**

Employee turnover remains a persistent issue across industries, with both direct and indirect costs for organizations. Identifying patterns and predicting employee attrition can help companies implement targeted interventions to retain valuable talent. In this context, machine learning offers a powerful approach to analyzing employee data and uncovering the factors most associated with voluntary departures.

The purpose of this project is to build a predictive model that estimates the likelihood of an employee leaving the company. The IBM HR Analytics dataset includes a range of features related to employee demographics, job roles, satisfaction levels, and performance metrics.

The objectives of the project are as follows:

- To preprocess and analyze the IBM HR dataset to extract meaningful insights

- To train and evaluate several machine learning models for predicting attrition

- To perform sensitivity analysis to understand the drivers of attrition

- To deploy the final model via a FastAPI application for HR usability

By the end of this study, I aim to provide both an analytical understanding of attrition drivers and a practical tool for HR professionals to use in daily decision-making.

# **2. Dataset Description**

This study uses the IBM HR Analytics Employee Attrition dataset, a publicly available ([IBM HR Analytics Employee Attrition & Performance](https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset)) fictional dataset, created by IBM’s Data Scientists for practitioners. It contains information on 1,470 employees. Each row represents a current or former employee, and the target variable, Attrition, indicates whether the employee has left the company (Yes) or not (No).

The dataset includes a diverse set of 35 features, covering multiple dimensions of employee data:

**Demographic data:** Age, Gender, MaritalStatus, Education, etc.

**Job-related data:** JobRole, Department, JobLevel, MonthlyIncome, YearsAtCompany, YearWithCurrManager, Ovetime etc.

**Performance & satisfaction:** JobSatisfaction, EnvironmentSatisfaction, PerformanceRating, etc.

**Behavioral indicators:** WorkLifeBalance, NumCompaniesWorked, etc.

The dataset has the following characteristics:

* **Total records:** 1,470
* **Binary classification problem:** The goal is to predict whether Attrition is Yes or No
* **No missing values:** The dataset is complete, though some features require transformation or encoding

Below is a **sample** of the features included in the dataset:

|  |  |
| --- | --- |
| **Table 1: Sample of Features** | |
| **Feature** | **Description** |
| Age | Age of the employee |
| BusinessTravel | Frequency of business travel |
| Department | Department the employee belongs to |
| DistanceFromHome | Distance from home to work (in km) |
| Education | Employee’s education level (1–5) |
| EnvironmentSatisfaction | Satisfaction with work environment (1–4) |
| JobRole | Employee’s job role |
| MonthlyIncome | Monthly income |
| NumCompaniesWorked | Number of previous employers |
| TotalWorkingYears | Total years of work experience |
| YearsAtCompany | Number of years at current company |
| Overtime | Whether the employee works overtime |
| Attrition (target) | Whether the employee has left the company or not |

The dataset is well-suited for building classification models and exploring relationships between various workplace factors and employee attrition.

# **3. Data Preprocessing**

Before building predictive models, the dataset underwent several preprocessing steps to ensure data quality and prepare it for machine learning algorithms.

**a. Missing Values**

A preliminary check showed that the dataset does not contain any missing values, which simplifies preprocessing.

**b. Feature Drops**

Certain columns were dropped due to having no variance or not being useful for prediction:

- EmployeeCount: Constant value for all records

- Over18: Constant value ("Y")

- StandardHours: Constant value (typically 40)

- EmployeeNumber: Serves as a unique identifier with no predictive power

**c. Target Variable Encoding**

The target variable “Attrition”, originally labeled as "Yes" or "No", was converted to binary format (1 = Attrition, 0 = No Attrition). (The same transformation applied to Overtime and Gender variables)

**d. Categorical Encoding**

* BusinessTravel was ordinal-encoded to integer numbers since it was an ordinal variable.
* Department, EducationField, JobRole, MaritalStatus were one-hot encoded.

These preprocessing steps ensured that the data was clean, consistent, and suitable for training machine learning models.

# **4. Feature Selection**

To identify the most meaningful predictors of employee attrition, I conducted a correlation analysis and drew upon domain knowledge supported by findings from existing research on the dataset. The aim was to balance predictive power with interpretability, ensuring the final model would be both effective and accessible for HR professionals. To avoid overfitting and maintain a user-friendly feature set, I selected a total of 15 features.

These features were chosen based on their observed correlations with attrition and their relevance as highlighted in previous literature. The final list includes:

* OverTime
* MaritalStatus\_Single
* TotalWorkingYears
* JobLevel
* YearsInCurrentRole
* MonthlyIncome
* Age
* JobRole\_Sales Representative
* YearsWithCurrManager
* StockOptionLevel
* YearsAtCompany
* JobInvolvement
* BusinessTravel
* JobSatisfaction
* EnvironmentSatisfaction

# **4. Class Imbalance**

One of the key challenges in predicting employee attrition is the imbalance in the target variable. In the IBM HR Analytics dataset, only about 16% of employees are labeled as having left the company (Attrition = Yes), while the remaining 84% stayed (Attrition = No). This creates a class imbalance that can significantly affect model performance.

Fig. 2: Class Imbalance in the datasetA graph of distribution of attrition

AI-generated content may be incorrect.

Without proper handling, many machine learning models tend to favor the majority class, resulting in high accuracy but poor recall for the minority class — in this case, the employees we aim to identify.

To address this issue, I considered techniques such as:

* Stratified train-test splitting to preserve class proportions
* Evaluation metrics beyond accuracy (e.g., F1 score, precision-recall)
* Potential use of resampling techniques (SMOTE), if needed during model tuning

# **5. Model Building and Evaluation**

**Baseline Model Setting and Performance Report**

To establish a strong baseline, a Random Forest Classifier was trained using the 15 selected features. Random Forest was chosen for its robustness and its built-in feature importance capability.

The dataset was split into training and test sets using stratified sampling, with 75% of the data used for training and 25% for testing. The model was trained using default hyperparameters.

The model’s performance was evaluated using key classification metrics:

* Precision: Measures the proportion of positive identifications that were actually correct
* Recall: Measures the ability of the model to find all relevant cases (attrition)
* F1 Score: Harmonic mean of precision and recall

Below you can find the results:

|  |  |  |
| --- | --- | --- |
| **Table 2: Classification Report for Baseline Model** | | |
| **Metric** | **Class 0 (Stayed)** | **Class 1 (Left)** |
| Precision | 0.85 | 0.42 |
| Recall | 0.96 | 0.14 |
| F1 Score | 0.91 | 0.21 |
| Support | 309 | 59 |

|  |  |
| --- | --- |
| **Table 3: Overall Performance for Baseline Model** | |
| **Metric** | **Score** |
| Accuracy | 0.83 |
| F1 Score (Macro Avg) | 0.56 |

|  |  |  |
| --- | --- | --- |
| **Table 4: Confusion Matrix for the Baseline Model** | | |
|  | **Predicted: 0** | **Predicted: 1** |
| Actual: 0 | 298 | 11 |
| Actual: 1 | 51 | 8 |

|  |  |
| --- | --- |
| **Table 5: Feature Importances for the Baseline Model** | |
| **Feature** | **Importance** |
| MonthlyIncome | 0.1495 |
| Age | 0.1202 |
| TotalWorkingYears | 0.1018 |
| YearsAtCompany | 0.0903 |
| YearsWithCurrManager | 0.0759 |
| OverTime | 0.0643 |
| JobSatisfaction | 0.0623 |
| YearsInCurrentRole | 0.0595 |
| EnvironmentSatisfaction | 0.0594 |
| StockOptionLevel | 0.0590 |
| JobInvolvement | 0.0480 |
| JobLevel | 0.0401 |
| BusinessTravel | 0.0356 |
| MaritalStatus\_Single | 0.0255 |
| JobRole\_Sales Representative | 0.0086 |

The Baseline Random Forest model provided valuable insights into the relative importance of each feature in predicting employee attrition. Among the top contributors were:

* **MonthlyIncome, Age,** and **TotalWorkingYears:** These financial and experience-related features suggest that compensation and tenure play a significant role in attrition decisions. Lower income and fewer years of experience may be associated with higher turnover risk.
* **OverTime** and **JobSatisfaction:** Consistent with HR literature, employees working overtime or reporting lower satisfaction levels were more likely to leave the organization.
* **YearsAtCompany, YearsWithCurrManager, and YearsInCurrentRole:** These tenure-related features highlight the importance of internal mobility and stable management relationships in employee retention.
* **JobRole\_Sales Representative and MaritalStatus\_Single:** These demographic and role-based features, though less impactful individually, suggest that personal circumstances and job nature may influence attrition likelihood.

Overall, the model indicates that a combination of compensation, work-life balance, career development, and management support are key drivers of attrition — insights that align with existing research and provide actionable direction for HR policy.

**How other Machine Learning Algorithms perform?**

To assess the performance of different machine learning algorithms, five models were trained and evaluated using the same set of features and data splits. These included Logistic Regression, Support Vector Machine (SVM), XGBoost, a Multilayer Perceptron (MLP), and a Random Forest model trained on resampled data using SMOTE.

While detailed evaluation results were generated for each model (including classification reports and confusion matrices), this report includes only a summary of key metrics for clarity and conciseness. A more in-depth analysis is provided for the baseline Random Forest model in the previous section and for the final selected model in the last section.

**Logistic Regression**

A Logistic Regression model was trained using a pipeline with feature scaling (MinMaxScaler) and hyperparameter tuning via GridSearchCV (Best Hypeparameters: C=0.001, solver='liblinear'). To address class imbalance, the model was configured with class\_weight="balanced". The best model achieved an **F1 score of 0.70** on the test set.

While the model maintained good overall accuracy (**0.80**), its performance on the minority class (employees who left) was particularly notable:

* **Recall** for attrition cases (class 1) reached **0.69**, significantly higher than the baseline Random Forest model.
* **Precision** was lower (**0.42**), suggesting more false positives, but this is often acceptable in HR settings where missing true attrition cases is more costly than flagging a few extra.

These results highlight Logistic Regression as a strong, interpretable model that balances performance and practical application well.

**XGB Classifier**

To further improve performance and address class imbalance, an XGBoost classifier was trained using scale\_pos\_weight=9 to give more weight to the minority class (attrition). The model was tuned via GridSearchCV, optimizing hyperparameters such as the number of estimators, learning rate, maximum depth, and gamma (Best Hyperparameters: n\_estimators=200, learning\_rate=0.15, max\_depth=3, gamma=0.2).

On the test set, the best XGBoost model achieved a **macro F1 score of 0.65**. The model had recall for the attrition class to **0.56**, while precision at **0.37**. These results indicate that the model can identify employees at risk of leaving compared to the baseline random forest, but it introduces some false positives and it is outperformed by the Logistic Regression.

XGBoost also provided feature importance scores, confirming that features like MonthlyIncome and OverTime were among the most predictive, aligning with earlier findings.

**Support Vector Machine (SVM)**

A Support Vector Machine (SVM) with a linear kernel was trained using a scaled pipeline and class\_weight="balanced" to mitigate class imbalance. Hyperparameter tuning was performed on the regularization parameter C, and the best model was selected using 5-fold cross-validation.

The model achieved a **macro F1 score of 0.69**, making it one of the stronger performers in this comparison. The recall for attrition cases (class 1) reached **0.73**, the highest among all models tested. Although precision was moderate (**0.40**), the model demonstrated a favorable trade-off by capturing a higher proportion of actual leavers, a critical factor in retention planning.

Overall, the SVM model provided a strong balance between predictive power and generalization, making it a valuable candidate.

**Multilayer Perceptron (MLP)**

To explore the predictive potential of deep learning methods, a Multilayer Perceptron (MLP) was developed using the TensorFlow/Keras framework. The architecture included two hidden layers (128 and 64 units respectively) with ReLU activations and dropout regularization to mitigate overfitting. The output layer used a sigmoid activation to handle the binary classification task of predicting employee attrition.

The model was compiled with the Adam optimizer and binary cross-entropy loss function. Feature scaling was performed using MinMaxScaler, and early stopping was implemented to prevent overfitting. During training, 20% of the training set was used for validation.

On the test set, the MLP achieved a **macro F1 score of 0.69** and an overall **accuracy of 87%**. While the recall for class 1 (attrition) was **0.36**, the model maintained a relatively high **precision of 0.66** for this class, suggesting that when it predicted attrition, it did so with confidence.

However, it is important to note that **neural networks introduce multiple sources of randomness** (e.g., weight initialization, dropout, and data shuffling). Unless these are explicitly controlled with fixed seeds and deterministic execution settings, **results may vary slightly across runs**. For this project, random seeds were not fully enforced, so the results of the MLP model should be considered **non-reproducible by default**.

**Random Forest with SMOTE and Hyperparameter Tuning**

To improve class balance and enhance model performance, a Random Forest classifier was trained within a pipeline that incorporated **SMOTE oversampling** on the training set. This approach was used to address the underrepresentation of the minority class (employees who left) by synthetically generating similar examples before model training.

The model was also fine-tuned using **GridSearchCV** with 5-fold cross-validation. The hyperparameters tuned included the number of estimators, maximum depth, minimum samples split, and minimum samples leaf (Best Hyperparameters: min\_samples\_leaf=2, min\_samples\_split=5, n\_estimators=300). The best-performing model was then evaluated on the original, imbalanced test set to maintain real-world comparability.

On the test set, the model achieved an overall **accuracy of 82%** and a **macro F1 score of 0.67**. Importantly, the **recall for class 1 (attrition)** reached around **0.47**. Precision for class 1 was around (**0.44**).

The results highlight that combining **SMOTE with ensemble learning and hyperparameter tuning** did not helped a lot to address class imbalance more effectively in this problem.

**Model Comparison and Summary**

A total of six models were developed and evaluated using the same set of 15 features derived from the IBM HR Analytics dataset. These models included traditional classifiers, ensemble methods, a deep neural network, and an ensemble with SMOTE resampling. The table below summarizes the performance of each model based on key evaluation metrics:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 6: Comparison across models** | | | | |
| **Model** | **Accuracy** | **F1 (Macro)** | **Recall (Attrition)** | **Precision (Attrition)** |
| Random Forest (baseline) | 0.83 | 0.56 | 0.14 | 0.42 |
| Logistic Regression | 0.82 | 0.7 | 0.69 | 0.42 |
| XGBoost | 0.78 | 0.65 | 0.56 | 0.37 |
| SVM (Linear) | 0.78 | 0.69 | 0.73 | 0.4 |
| MLP (Keras) | 0.87 | 0.69 | 0.36 | 0.66 |
| Random Forest + SMOTE | 0.82 | 0.67 | 0.47 | 0.44 |

**Key Insights:**  
  
- Baseline Random Forest performed well in terms of overall accuracy but failed to detect attrition cases (very low recall).  
- Logistic Regression with class balancing was one of the most interpretable and balanced models, achieving high recall and macro F1.  
- SVM yielded the highest recall (0.73) for class 1, making it valuable in scenarios where identifying potential leavers is a priority.  
- MLP failed to detect leavers but reached highest accuracy.  
- XGBoost failed also to detect leavers and had moderate recall.  
- SMOTE seems to help random forest improve a bit but there is a big difference with the two linear models.

**Final Considerations:**  
  
While no model perfectly balanced all performance aspects, Logistic Regression and SVM emerged as strong, interpretable, and reliable choices, especially when recall is a business-critical priority.

**Winning Model: Logistic Regression**

**Evaluation Results**

Based on the evaluation metrics and practical considerations, Logistic Regression was selected as the final model. SVM could also be chosen, as its high attrition recall (0.73) and strong F₁ (0.69) can make it also effective in the task. Logistic Regression provided the slightly more balanced performance in terms of precision, recall and F1 score, while remaining interpretable and practical for HR use cases. The following tables summarizes the evaluation results of the final model on the test set.

|  |  |  |
| --- | --- | --- |
| **Table 7: Classification Report of Logistic Regression** | | |
| **Metric** | **Class 0 (Stayed)** | **Class 1 (Left)** |
| Precision | 0.93 | 0.42 |
| Recall | 0.82 | 0.69 |
| F1 Score | 0.87 | 0.53 |
| Support | 309 | 59 |

|  |  |  |
| --- | --- | --- |
| **Table 8: Overall Metric for the Logistic Regression** | | |
| **Metric** | **Class 0 (Stayed)** | **Class 1 (Left)** |
| Precision | 0.93 | 0.42 |
| Recall | 0.82 | 0.69 |
| F1 Score | 0.87 | 0.53 |
| Support | 309 | 59 |

|  |  |  |
| --- | --- | --- |
| **Table 9: Confusion Matrix of the Logistic Regression** | | |
|  | **Predicted: No** | **Predicted: Yes** |
| Actual: No | 254 | 55 |
| Actual: Yes | 18 | 41 |

**Feature Effects Based on Logistic Regression Coefficients**

One of the key advantages of Logistic Regression is its interpretability. Each coefficient represents the impact of a specific feature on the likelihood of employee attrition. A \*\*positive coefficient\*\* increases the probability of attrition, while a \*\*negative coefficient\*\* decreases it. The table below presents the sorted coefficients for the 15 selected features used in the model.

|  |  |
| --- | --- |
| **Table 10: Feature Influences** | |
| **Feature** | **Coefficient** |
| OverTime | 0.072007 |
| MaritalStatus\_Single | 0.058234 |
| BusinessTravel | 0.027242 |
| JobRole\_Sales Representative | 0.024665 |
| YearsAtCompany | -0.016912 |
| JobInvolvement | -0.020149 |
| TotalWorkingYears | -0.023233 |
| JobSatisfaction | -0.023724 |
| YearsInCurrentRole | -0.023829 |
| YearsWithCurrManager | -0.024384 |
| MonthlyIncome | -0.024777 |
| Age | -0.026231 |
| StockOptionLevel | -0.026756 |
| EnvironmentSatisfaction | -0.027582 |
| JobLevel | -0.028217 |

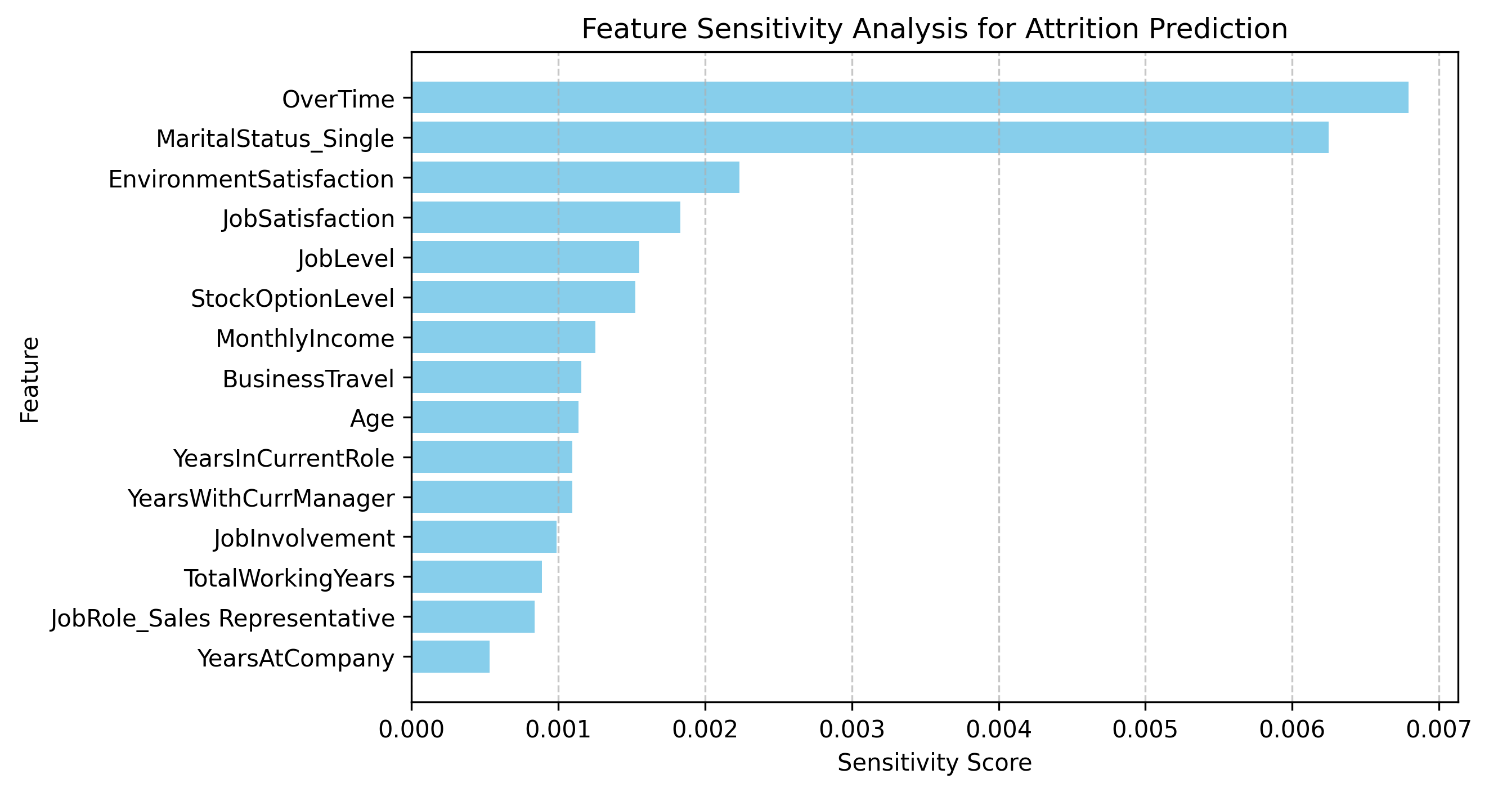
The table above highlights how each feature influences the probability of employee attrition according to the final logistic regression model. Features such as **OverTime** and **MaritalStatus\_Single** have **positive coefficients**, indicating that employees who work overtime or are single are more likely to leave the organization. Similarly, frequent **BusinessTravel** and the role of **Sales Representative** were also positively associated with higher attrition risk.

On the other hand, features like **JobLevel**, **EnvironmentSatisfaction**, **Age**, and **MonthlyIncome** have **negative coefficients**, suggesting they are associated with increased employee retention. These results align with prior research and HR intuition, reinforcing the model’s interpretability.

This interpretation can provide actionable insights for HR professionals, offering a clearer understanding of the drivers behind attrition and potential areas for targeted intervention.

**Sensitivity Analysis**

In addition to examining model coefficients, a sensitivity analysis was performed to understand how changes in individual features influence the model’s predictions. For this analysis, the value of each feature in the test set was replaced with its mean, one feature at a time, while the others remained unchanged. The model was then used to re-predict the probability of attrition for each modified dataset, and the average absolute change in predicted probabilities was computed. This method helps identify which features the model is most sensitive to. Higher sensitivity values indicate that a small change in that feature causes a greater shift in predicted attrition probabilities. Unlike coefficients, which show the direction and magnitude of influence in a linear model, sensitivity analysis measures the practical influence of features regardless of direction. It is especially helpful in understanding the robustness of predictions. The chart below summarizes the results:  
 Fig 3: Sensitivity for the best Model



The sensitivity analysis revealed that the model’s predictions were most affected by changes in the **OverTime** and **MaritalStatus\_Single** features. Even slight modifications to these inputs led to noticeable shifts in predicted attrition probabilities, indicating that the model heavily relies on them to assess risk. Other features with moderate sensitivity include **EnvironmentSatisfaction**, **JobSatisfaction**, and **JobLevel**, suggesting that employees' perceptions of their work environment and their position in the organizational hierarchy also play a key role in influencing the model's output. These insights can help HR teams prioritize areas of focus for improving retention strategies.

**Model Deployment**

To ensure that the developed machine learning solution is usable beyond a static notebook environment, the final Logistic Regression model was deployed as a RESTful API using **FastAPI**.

The deployment pipeline included the following steps:

1. **Final Model Retraining**  
   The selected Logistic Regression model (with optimal hyperparameters from GridSearchCV) was retrained on the entire dataset to maximize generalization. The model and required input features were then serialized using joblib.
2. **FastAPI Application**  
   A lightweight API was built using the FastAPI framework. It accepts employee profile data via a POST request, performs input validation (e.g., salary ranges, binary flags), and returns the predicted attrition probability and binary decision (0 = stay, 1 = likely to leave).
3. **Swagger UI for Interactive Testing**  
   FastAPI automatically generated a Swagger documentation page (/docs), allowing end users (e.g., HR professionals) to test predictions via a browser-based form without needing coding skills.
4. **Cloud Hosting with Render.com**  
   The API was deployed on **Render**, a cloud platform that provides free-tier hosting for web applications. A GitHub repository was linked to Render for version control and automatic redeployment on future updates.

The public API is accessible at:

<https://hr-attrition-api-1.onrender.com/docs>

**Limitation and Considerations**

While the results and deployment of the model are promising, several limitations should be acknowledged:

1. **Imbalanced Dataset**  
   The dataset contains a significant class imbalance, with far fewer attrition cases compared to non-attrition. Despite the use of class\_weight="balanced" and sampling techniques like SMOTE in some experiments, this imbalance still affects model sensitivity and recall for the minority class.
2. **Static Dataset and Context**  
   The IBM HR dataset is synthetic and may not fully reflect the complexity or variability of real-world HR data. Therefore, while the model performs reasonably, its generalizability to live company data may be limited unless retrained and validated on organization-specific records.
3. **Feature Scope and Depth**  
   The available features, while meaningful, may not capture deeper behavioral or qualitative factors that influence attrition (e.g., team culture, job fit, burnout). The model is constrained by the data it is trained on.
4. **Deployment Constraints**  
   The API is hosted on the Render free tier, which can lead to latency on first use (due to container "spin-up" after inactivity). This setup is suitable for demos and prototyping, but a production environment would require a more robust deployment strategy.

**Scalability in Real-World Settings**

To scale the attrition prediction model in a real-world organizational environment, the solution must move beyond a static report or one-time analysis and become part of the organization’s operational HR ecosystem. This begins with embedding the model within the company’s existing digital infrastructure — for example, by integrating the FastAPI endpoint into the organization’s HR Information System (HRIS), business intelligence tools, or employee management platforms. In this way, predictions can be generated automatically as employee data is updated, ensuring that attrition risk assessments are always current.

Scalability also requires automation. By scheduling periodic data extraction from internal systems and feeding that data directly to the API, the model can evaluate thousands of employees across departments or locations without manual intervention. Alerts can be triggered when high-risk predictions are detected, feeding directly into HR workflows for intervention. For decision-makers, these predictions can be aggregated and visualized in dynamic dashboards, providing real-time insights at both the individual and organizational level.

Finally, for true enterprise-scale deployment, the system must be governed with attention to data security, privacy, and fairness. With the right controls, model monitoring, and periodic retraining pipelines in place, the solution can grow into a robust, always-on HR decision support system. This transforms the attrition model from a project into a scalable, strategic asset that empowers proactive, personalized, and data-driven talent management.

**Conclusion**

This project developed and deployed a machine learning-based solution to predict employee attrition using structured HR data. The workflow covered the entire data science lifecycle—from exploratory data analysis and feature selection to model training, evaluation, and API deployment. Among the models tested, logistic regression was selected for its interpretability, reliability, and practical performance.

While limitations exist—such as data imbalance, synthetic and time-snapshot nature of the dataset, the project demonstrates a clear pathway toward building scalable HR analytics tools that can be embedded within modern organizations. With further data integration, feedback loops, and governance, this solution can grow into a powerful strategic asset.

**Related Projects and Case Studies**

1. [**Employee Attrition Prediction – Sanat Ladkat (GitHub)**](https://github.com/sanatladkat/Employee-Attrition-Prediction)
2. [**IBM HR Analytics – Shantanu (GitHub)**](https://github.com/shantanu1109/IBM-HR-Analytics-Employee-Attrition-and-Performance-Prediction)
3. [**Employee Attrition Prediction – Aastha985 (GitHub)**](https://github.com/aastha985/Employee_Attrition_Prediction)
4. [**Predicting Employee Attrition**](https://www.kaggle.com/datasets/pavan9065/predicting-employee-attrition)
5. [**Employee Attrition Classification Dataset**](https://www.kaggle.com/datasets/stealthtechnologies/employee-attrition-dataset)
6. [**IBM HR Analytics Employee Attrition & Performance**](https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset/code)

**Appendix**

**API Glossary:**

How to fill the fields

* **OverTime:** 0 = No, 1 = Yes
* **MaritalStatus\_Single:** 0 = Married/Divorced, 1 = Single
* **TotalWorkingYears:** any non-negative number
* **JobLevel:** integer from 1 to 5
* **YearsInCurrentRole:** any non-negative number
* **MonthlyIncome:** between 1000 and 20000
* **Age:** between 18 and 65
* **JobRole\_Sales\_Representative:** 0 = No, 1 = Yes
* **YearsWithCurrManager:** any non-negative number
* **StockOptionLevel:** integer from 0 to 3
* **YearsAtCompany:** any non-negative number
* **JobInvolvement:** integer from 1 (Low) to 4 (High)
* **BusinessTravel:** 0 = Non-Travel, 1 = Travels Often, 2 = Travels Frequently
* **JobSatisfaction:** integer from 1 (Low) to 4 (High)
* **EnvironmentSatisfaction: integer** from 1 (Low) to 4 (High)