Employee Attrition Prediction using Machine Learning Algorithms

Team Members

Sai Koushik Valluri - saikoushikvalluri@my.unt.edu ,

Venkata Durga Mani Ratna Srikar Duriseti - maniratnasrikarvdduriseti@my.unt.edu ,

Bala Narayana Subbarao Chikkala - balachikkala@my.unt.edu ,

Naga Suchandra Tirumalasetti - nagasuchandratirumalasetti@my.unt.edu

Workflow

- Data Summary.
- Data Cleansing and Pre-processing.
- Exploratory Data Analysis.
- Model building with Imbalanced data.
- Model building with Balanced data using Random Oversampling and SMOTE techniques.
- Feature Selection.
- Model Evaluation.
- UI

Abstract

Employee attrition occurs when an employee leaves the organization through any method like resignation, retirement, layoffs etc. Employee Attrition is a major challenge to the organizations. It disrupts workflow management, decreases employee morale and destroys organization reputation. This project will provide a solution to predict employee attrition using a machine learning approach. Employee attrition is defined as the process by which employees leave the organization – for example, through resignation for personal reasons or retirement – and are not immediately replaced.

Data Specification

•This is a fictitious dataset created by IBM data scientists and published in Kaggle.

•Dataset contains 35 attributes like Education, Environment Satisfaction, Job Involvement, Job Satisfaction, Performance Rating, Relationship Satisfaction, etc.

•Attrition is the target variable which is categorical (Binary Class) in nature. This is eventually a classification problem which is supervised machine learning.

•This Dataset contains 26 integer columns, 6 string columns and 3 Boolean columns.

Column	Non-Null Count	Dtype
Age	1470 non-null	int64
Attrition	1470 non-null	object
BusinessTravel	1470 non-null	object
DailyRate	1470 non-null	int64
Department	1470 non-null	object
DistanceFromHome	1470 non-null	int64
Education	1470 non-null	int64
EducationField	1470 non-null	object
EmployeeCount	1470 non-null	int64
EmployeeNumber	1470 non-null	int64
EnvironmentSatisfaction	1470 non-null	int64
Gender	1470 non-null	object
HourlyRate	1470 non-null	int64
JobInvolvement	1470 non-null	int64
JobLevel	1470 non-null	int64
JobRole	1470 non-null	object
JobSatisfaction	1470 non-null	int64
MaritalStatus	1470 non-null	object

Column	Non-Null Count	Dtype
MonthlyIncome	1470 non-null	int64
MonthlyRate	1470 non-null	int64
NumCompaniesWorked	1470 non-null	int64
Over18	1470 non-null	object
OverTime	1470 non-null	object
PercentSalaryHike	1470 non-null	int64
PerformanceRating	1470 non-null	int64
RelationshipSatisfaction	1470 non-null	int64
StandardHours	1470 non-null	int64
StockOptionLevel	1470 non-null	int64
TotalWorkingYears	1470 non-null	int64
TrainingTimesLastYear	1470 non-null	int64
WorkLifeBalance	1470 non-null	int64
YearsAtCompany	1470 non-null	int64
YearsInCurrentRole	1470 non-null	int64
YearsSinceLastPromotion	1470 non-null	int64
YearsWithCurrManager	1470 non-null	int64

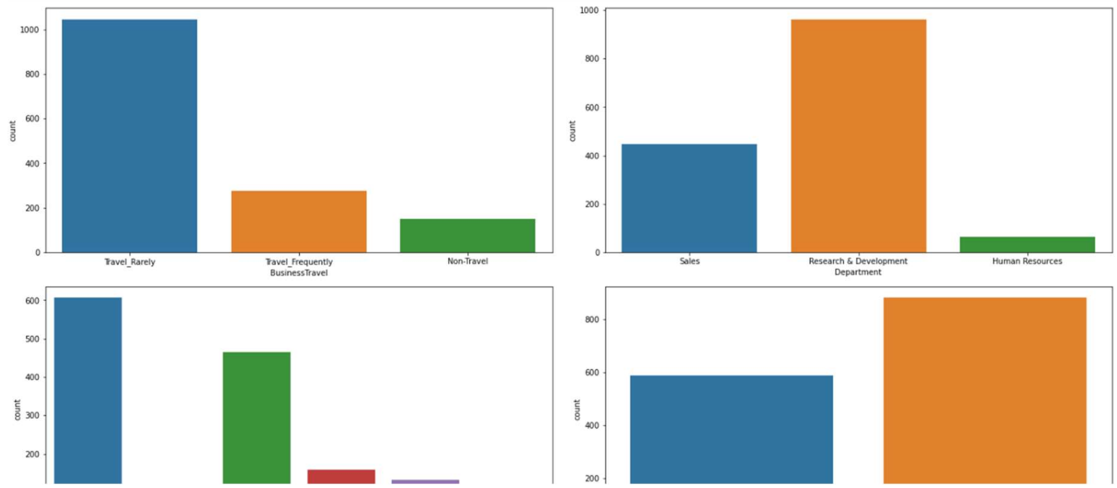
Project Design:

We used Python for building ML algorithms and used R Programming languages to build UI. Since our target variable is a binary class, we used Logistic Regression, Decision Tree Classifier and Random Forest Classifier with different balanced and imbalanced approaches. We performed EDA with less lines of code using a for loop which saved a lot of time and effort. Please find the codes in below screenshots.

```python
fig, ax = plt.subplots(4,2,figsize=(20,20))
ax = ax.flatten()

for i, col in enumerate(categorical_columns[1:]):
    sns.countplot(data[col], ax = ax[i])

plt.tight_layout()
plt.show()
```



Project Milestones:

- Data Summary.
- Data Cleansing and Pre-processing.
- Exploratory Data Analysis.
- Model building.
- Features Selection.
- Model Evaluation.
- UI development.

Project Results:

We transformed Categorical Variables from String to Numerical data type using Label Encoder since ML models do not accept string data types.

We built models using three different approaches considering all the features and the performance metrics are listed below in the table.

Balancing and Imbalanced Techniques	Classification Algorithms	Train			Test		
		Accuracy	Precision	Recall	Accuracy	Precision	Recall
Imbalanced	Logistic Regression	91%	83%	100%	93%	86%	100%
	Decision Tree	100%	100%	100%	87%	89%	86%
	Random Forest	100%	100%	100%	93%	87%	99%
Balanced using RandomOverSampling	Logistic Regression	64%	66%	59%	66%	68%	62%
	Decision Tree	100%	100%	100%	90%	99%	82%
	Random Forest	100%	100%	100%	97%	98%	96%
Balanced using SMOTE technique	Logistic Regression	67%	69%	62%	70%	70%	69%
	Decision Tree	100%	100%	100%	83%	87%	77%
	Random Forest	100%	100%	100%	92%	91%	90%

We did feature selection based on EDA and below are the results after feature selection.

Balancing and Imbalanced Techniques	Classification Algorithms	Train			Test		
		Accuracy	Precision	Recall	Accuracy	Precision	Recall
Imbalanced	Logistic Regression	84%	85%	98%	85%	87%	97%
	Decision Tree	100%	100%	100%	81%	90%	88%
	Random Forest	100%	100%	100%	86%	87%	98%
Balanced using RandomOverSampling	Logistic Regression	68%	69%	67%	64%	64%	66%
	Decision Tree	100%	100%	100%	99%	82%	91%
	Random Forest	100%	100%	100%	96%	98%	84%
Balanced using SMOTE technique	Logistic Regression	70%	69%	71%	70%	69%	73%
	Decision Tree	100%	100%	100%	80%	82%	76%
	Random Forest	100%	100%	100%	90%	89%	91%

Based on the results obtained random forest algorithm without feature selection is the best performing machine learning algorithm to predict employee attrition.

Repository / Archive:

https://github.com/SrikarDuriseti/Employee-attrition-prediction.git

References

Reference1:
https://www.researchgate.net/publication/361522993_Predicting_Employee_Attrition_Using_Machine_Learning_Approaches
Reference 2:
https://ieeexplore.ieee.org/document/9825342
Kaggle:
https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset/code