

**231-MATH-506-03**

**Fundamentals of Data Science**

**Project Proposal**

**Mamoun Alghaslan and Khaled Almutairy**

**Prepared for Dr. Jamal Alsmail**

Employee retention analysis

(<https://www.kaggle.com/datasets/tawfikelmetwally/employee-dataset>)

There are always people starting new jobs and retiring, and some move between jobs. Talent retention is imperative for organizational success, therefore understanding employee behavior is key to sustaining a good economy. Kaggle Employee dataset contains information about employees in a company, including their educational backgrounds, work history, demographics, and employment-related factors, which has been anonymized to protect privacy while still providing valuable insights into the workforce.

This dataset contains 9 features:

|  |  |
| --- | --- |
| **Feature** | **Description** |
| **Education** | Level of Education(Bachelors, Masters and PHD) |
| **JoiningYear** | The year of joining |
| **City** | To which city the employee belongs to |
| **PaymentTier** | Salary Tiers |
| **Age** | The age of the Employee |
| **Gender** | Male and Female |
| **EverBenched** | If the employee has ever benched (yes or no) |
| **ExperienceInCurrentDomain** | Employee experience in terms of years |
| **LeaveOrNot** | The target column which is a binary column (0 or 1) |

# Stage 1: Problem Understanding

In the initial phase of our proposed data science project, our aim is to gain a deep understanding of the problem with the following objectives:

1. Analyze the distribution of each feature over leave or not.
2. Identify patterns in leave-taking behaviors.
3. Build a model to predict whether an employee will leave or not.

# Stage 2: Data Preparation

In this stage, we will prepare the data to ensure it is clean, consistent, and free from errors for analysis exploring basic information about the dataset and identifying outliers if present. The dataset has 4653 records. All records have no null values or outliers. Each feature of the dataset has the following values:

|  |  |
| --- | --- |
| **Feature** | **Description** |
| **Education** | 'Bachelors' 'Masters' 'PHD' |
| **JoiningYear** | 2012 2013 2014 2015 2016 2017 2018 |
| **City** | 'Bangalore' 'Pune' 'New Delhi' |
| **PaymentTier** | 1 2 3 |
| **Age** | [22-41] |
| **Gender** | 'Male' 'Female' |
| **EverBenched** | 'No' 'Yes' |
| **ExperienceInCurrentDomain** | 0-7 |
| **LeaveOrNot** | 0 1 |

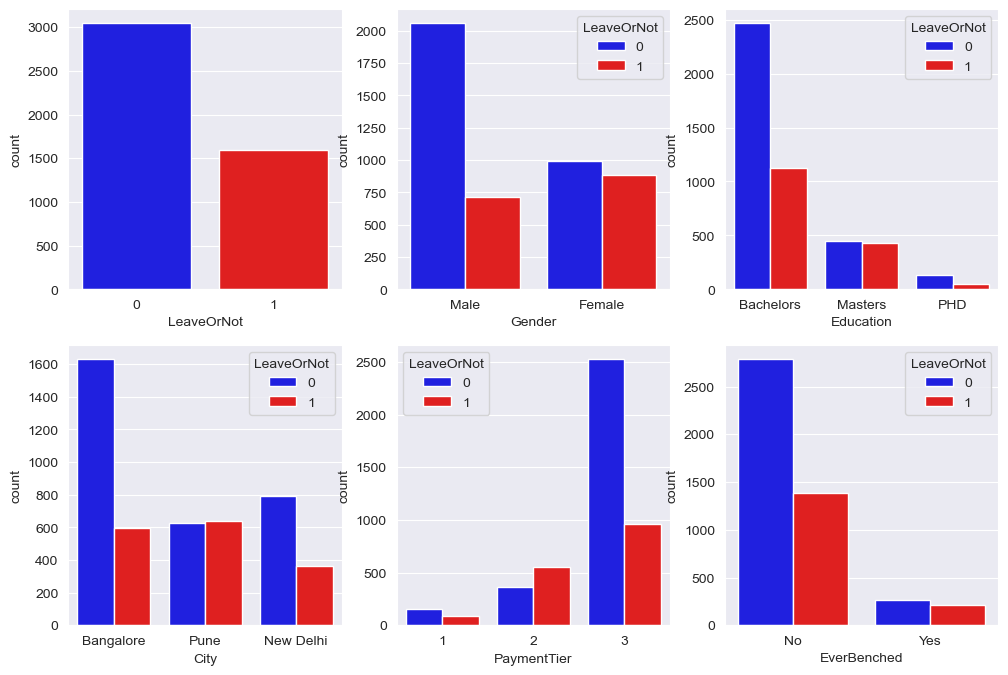
# Stage 3: Exploratory Data Analysis (EDA)

The goal in this stage is to gain preliminary insight into the data by utilizing graphical exploration. This stage is useful to uncover patterns and visualize relationships among variables. In this phase, we will:

1. **Explore Univariate Relationships** over Leave or Not: Identify relationships between predictors and the target variable “Leave or Not.”
2. **Explore Multivariate Relationships**: Identify correlations between multiple attributes.

## Univariate Relationships over the target

The data has 4653 records, of which 3053 (65%) did not leave, and 1600 (35%) left. We will explore the distribution of each column in the dataset based on the target column “Leave or Not” as a count, then as normalized.



From the above plots, we can see that:

1. Females tend to leave more frequently
2. Employees with a bachelor’s degree tend to leave more frequently
3. Employees like to stay in Bangalore
4. Employees with higher salaries tend to leave less frequently
5. Employees who have been benched tend to leave more frequently and this will become clearer with the following plots.

A screenshot of a graph

Description automatically generated

From the above plots, we can see that:

1. Females tend to leave more frequently
2. Employees with a master’s degree leave the most, then bachelor’s degree, then PHD.
3. Employees like to stay in Bangalore, then New Delhi, then Pune.
4. Employees in the middle tier tend to leave more frequently, probably looking for even better jobs or that that market is more competitive.
5. Employees who have been benched tend to leave more frequently.

A group of blue and red bars

Description automatically generated

From the above plots, we can see that:

1. There are more employees with less years of experience except for the year 2016.
2. There are more employees with less age except for the age 30.
3. Most employees in the dataset have 2 to 6 years of experience.

A screenshot of a graph

Description automatically generated

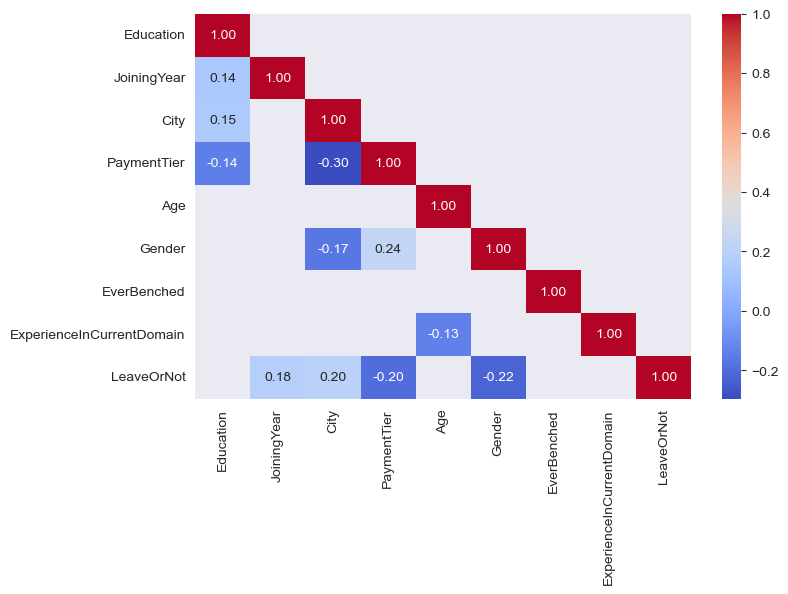
From the above plots, we can see that:

1. Almost all employees who joined in 2018 left their jobs.
2. There isn't a big variation in leave taking behavior based on age, but younger employees tend to leave slightly more frequently.
3. The same can be said to the experience in current domain, but employees with 2 or 3 years of experience tend to leave the most.

## Multivariate Relationships

We will look at the correlation between the columns to see if there are any relationships between them. The features must be converted from categorical columns to numerical columns before doing so.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| **Education** | 0 1 2 |
| **JoiningYear** | 2012 2013 2014 2015 2016 2017 2018 |
| **City** | 0 2 1 |
| **PaymentTier** | 1 2 3 |
| **Age** | [22-41] |
| **Gender** | 1 0 |
| **EverBenched** | 0 1 |
| **ExperienceInCurrentDomain** | 0-7 |
| **LeaveOrNot** | 0 1 |



From the above plot, we can see that there is no strong correlation between any two columns, even after removing any correlation less than 0.1. The remaining correlations are still weak, but most noticeable are PaymentTier and one of City or Gender, and LeaveOrNot with Gender.

# Stage 4: Setup

At this stage, we will have explored the data enough to understand the problem and the dataset, and we need to prepare it to be fed into the potential models by:

1. Separating the data into train, validation, and test sets
2. Balancing the dataset using SMOTE
3. Establishing baseline model performance

The baseline model is the simplest model that we can use to compare the performance of other models. In this case, we will use the most frequent class as the baseline model. Hence, the baseline model will predict 0 for all entries, and the accuracy will be 65.6%.

We will split the dataset into train, validation, and test sets with a ratio of 70%, 15%, and 15% respectively. Then, we will balance only the training set using SMOTE to ensure that the model is not biased towards the majority class. Finally, we will compare the performances of different models to the baseline model.

# Stage 5: Modeling

Now the dataset is set and ready, we will feed it to different algorithms/models to uncover the relationships between the columns and target. In this phase, we need to:

1. **Implement three algorithms/models**: At least three classification models.
2. **Train the models on training set and validate using validation set**: to achieve the best possible scores for the classification.
3. **Train the best model on training and validation sets combined and validate on test set**: to ensure the model is not overfitting.

The following table and charts summarize the results of the experiments:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method/Algorithm** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| Random Forest | 0.812 | 0.662 | 0.761 | 0.708 |
| Gradient Boosting | 0.821 | 0.708 | 0.755 | 0.731 |
| Voting Gradient Boosting | 0.847 | 0.733 | 0.804 | **0.767** |

A red and white squares with numbers

Description automatically generated.

1. Random Forest Confusion Matrix

A green squares with numbers and a green box

Description automatically generated

1. Gradient Boosting Confusion Matrix

A graph of a number of blue squares

Description automatically generated with medium confidence

1. Voting Gradient Boosting Confusion Matrix

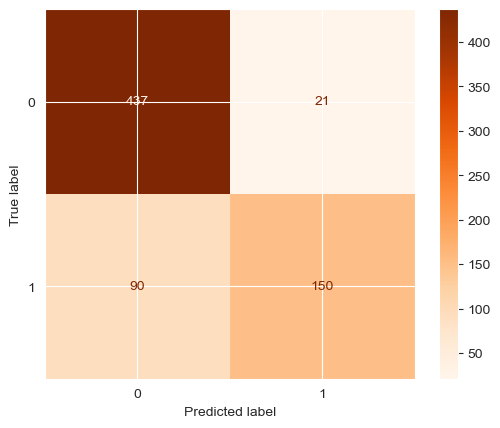
From the above results, we can see that the voting classifier is best at all measures, so we will choose it as our best model.

# Stage 6: Evaluation

In this phase, the testing portion will be trained on the training and validation sets combined, and then evaluated on the test set. The goal is to ensure that the model is not overfitting and that it can generalize well to unseen data.

The following is results of the model trained on training plus validation, and validated against the unseen test set:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method/Algorithm** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| Voting Gradient Boosting | 0.841 | 0.625 | 0.877 | 0.730 |



# Stage 7: Deployment

The model is available at the request of the audience 😊