**EMPLOYEE PERFORMANCE ANALYSIS PROJECT**

**PROJECT ANALYSIS**

**Exploratory data analysis and choices made during the project**

Exploratory data analysis of the given dataset was carried out in first part of the project to understand the current employee data provided and to gain useful insights.

* The dataset was complete without any null values.
* Information regarding department wise employee performances was obtained by calculating the average employee performance and plotting a bar chart (matplotlib) of the same.
* Feature importance were delineated by calculating the chi-squared statistic(scikit learn), which indicates the dependence of the feature to the target variable.
* To understand the root cause of the performance issue, the high and low performance employees were compared with their average values for the top 6 important features. From the insights obtained from this, appropriate recommendations were made.
* Multicollinearity was checked by plotting a heatmap(seaborn) and no strong/ very high correlations were observed.
* Imbalance in target variable values were determined using a countplot(seaborn) and an imbalance was present.

In the next part, data pre-processing was done to prepare the data for training and testing the machine learning models.

* Important data pre-processing steps like splitting predictor and target variables and encoding of categorical variables(using OrdinalEncoder from scikit learn) was already done as a part of the exploratory data analysis.
* The continuous variables were scaled using the MinMaxscaler (scikit learn) to convert the data points into values in between 0 and 1.
* Data was split in to training and testing data (scikit learn).
* Target variable imbalance in the data was addressed by using synthetic minority oversampling technique(SMOTE) (imblearn).

The next step was to choose a suitable machine learning model for the problem in hand. For this, the availabe machine learning algorithms for multi-class classification were tried out to check their accuracies in classifying the data points into 1 of the 3 performance classes.

* XGBoost algorithm proved to have the best accuracy(93.33%) and it was employed for the project.
* Hyperparameter tuning oF XGBoost algorithm was carried out using RandomizedSearchCV (scikit learn) function to obtain the model with best parameter combination.
* The best model gave a test accuracy of 100% and it was saved for deployment using the joblib package.