Salifort Employee Retention Model

2024-17-06

Introduction

The purpose of this document is to explain the steps taken to create a machine learning model for the HR department of the Salifort company by analyzing and modifying the data provided by the HR department.

Objective of the project

Create a model to determine if an employee will depart or stay in Salifort so that features which contribute to the employee retention can be found and the HR department can use those features to avoid employees exits and save money that would be used searching for replacements.

For this assignment, a project proposal was written for the senior leadership team to outline the milestones and their tasks. Then, the data of a CSV file was analyzed using the Python programming language in a jupyter notebook to create the model. The data on this file was collected from an employee survey that the HR department developed to gather information regarding the employees personal information like performance scores in evaluations, promotions over the last years, time in the company, and other information.

Finally, an executive summary was written for the leadership team to share the results of the analysis and the tests performed in the model and its limitations

Changelog

This section will provide a list of the modifications done to the data as well as documents created and tasks done to assist in the project development.

Version 1.0.0 (17-06-2024)

- Write the project proposal for the senior leadership team to explain the goal of the model and the milestones required to complete the assignment.
- Complete the Plan stage of the PACE workflow. The tasks done in this stage were:
  - Before the Jupyter notebook analysis:
    - Determine the main objective of the project and the stakeholders of it.
    - Understand what kind of information represents each data column in the CSV file.
    - Consider the ethical obligations of creating a machine learning model like who will use the model? What will it be used for? What are the consequences of the model when a false positive and a false negative are produced?
    - Research some metrics to decide which one to evaluate the model once it is built based on the goal of the project.

  - During the analysis:
    - Import the necessary python packages for the project, some of those are pandas and numpy for data manipulation, matplotlib and seaborn for data visualization and sklearn for preprocessing, model development and evaluation.
    - Load the data set into a pandas data frame called df0.
    - Compute descriptive statistics like the mean, maximum and minimum values of the numerical variables to understand their distributions and the number of different classes and frequency of the most used class of the categorical variables.
    - Modify the name of some data columns into snake case notation for further analysis.
    - Check for missing values and duplicated values in the data set.
    - Create a new data frame named df1 which has no duplicated values. This new object will be used for the following stage to make visualizations.
    - Build box plots for the numerical variables whose values are not on a scale from 0 to 1 (‘tenure_years’, ‘projects_assigned’, ‘average_montly_hours’) to check the presence of outliers in any of them.

- Begin the Analysis stage of the PACE workflow. The tasks done in this stage were:

  - Before the Jupyter notebook analysis:
    - Answer questions for the stage like:
      a. What steps need to be taken to perform Exploratory Data Analysis (EDA) in the most effective way to achieve the project goal?
      b. What initial assumptions do I have about the types of visualizations that might best be suited for the intended audience?
      c. What are some purposes of EDA before constructing a multiple linear regression model?
      d. What ethical considerations do I have in this stage?
  - During the analysis:
    - Calculate the balance proportion of the target variable by indicating the number of rows of employees who left and those that remain in the company. Then, repeat the process but express the results as a percentage of the data.
    - Compute descriptive statistics for the rows of lost and current employees.
    - Plot visuals of the target variable (employee_left) vs the predictor to understand better how each variable affects the target one. The visuals plotted are the followings:
      a. Side by Side Bar charts where the x variable is the predictor variable, the y variable is the number of occurrences for the predictor variables which are binary or categorical. The legend on the top right indicates the value of the target variable.
      b. Pandas series which show the percentage of observations of that fall on each value of the binary features below their respective bar charts.
      c. Histograms where the x variable is a numerical continuous predictor variable, the y variable is the number of occurrences for the numerical predictor variables.

Version 1.1.0 (18-06-2024)

- Complete the Analysis stage. The tasks done were:

  - During the analysis:
    - Create a function to plot and improve the current side by side bar charts. The name of this function is count_bar_chart.
    - Check the multicollinearity of the numerical predictor variables to confirm if one of the Logistic Regression Model assumptions was satisfied.

    - Write the insights found during this stage.
    - Perform feature engineering by:
      a. Capitalizing the values of the categorical variables.
      b. Performing One-Hot encoding to replace the categorical columns with binary columns. Those new columns would be saved in a new data frame called df_dummy which will be used to build the machine learning models on next stage.

- Begin the Construct stage of the PACE workflow. The tasks done in this stage were:
  - Before the Jupyter notebook analysis:
    - Answer questions for the stage like:
      a. What data visualizations, machine learning algorithms, or other data outputs will need to be built to complete the project goals?
      b. What processes need to be performed to build the necessary data visualizations?
      c. What ethical considerations do I have in this stage?
      d. Which variables are most applicable for the visualizations in this data project?
    - Research and make a list of statistical and machine learning models used for classification tasks while checking the assumptions of each model and their strengths and weaknesses.
  - During the analysis:
    - Perform hypothesis testing in the following variables:
      a. Satisfaction Level.
      b. Last Evaluation Score.
      c. Projects Assigned.
      d. Average Monthly Work Hours.
      e. Tenure Years.
    - Split the columns of the dataset in 2 groups, predictor (X) and target (y) and save those groups into new variables.
    - Split each of the 2 new objects into 3 new sets, one set for training the models (60% of the rows), one to validate them and decide which model is better at predicting the outcome (20% of the rows) and the last one to test the final model (20% of the rows). One important aspect of this step is to keep the proportion of rows that correspond to the values of the target variable to avoid inserting bias in the data sets and generate insights and conclusions which are not accurate.
    - Create a dictionary of hyper parameters for each function used to build one of the 3 models to perform Cross Validation in each model and determine the parameters which produce thr best model using the training sets.
    - Print the parameters used to build each model after the Cross Validation process as well as the best score for the metric used to select the parameters which is the f1 score.
    - Build a function to store the best metrics of the model with the parameters that produce better results. The name of this function is best_metrics_frame. The arguments are the title of the model and the model object used.
    - Use the metrics function to create a data frame with the training metrics for each model to compare their performance.
    - Use the 3 models to create 3 new sets of predictions of the target variable (one set per model) using the validation set of predictor variables.
    - Create a function to plot a confusion matrix. The arguments of this function are the set with the actual observations of the target variable, the set with the predicted observations, a title for the plot, and the model object. The name given to this function is confusion_matrix_displayer.

        - Add an explanation of what each quadrant of the matrix produced by the confusion_matrix_displayer represents.
        - Use the previous function to plot a confusion matrix for each model using the validation sets.
        - Compute the F1, recall, and precision score for each model using the validation sets.
        - Compare the matrices for each model and the metrics to decide which model would be used with the test data.

Version 1.2.0 (19-06-2024)

- Continue the Construct stage. The steps done were:
    - Repeat the Cross Validation process for the winning model, in this case the Random Forest model, using both the training and validation sets.
    - Repeat the steps done during the validation process in the previous version using the test data sets for the predictor and target variables.
    - Create a series to store the importance of each feature in the model.
    - Plot a bar chart using the series created in the previous step.
    - Answer the following questions after completing the final model creation:
        - How well does your model fit the data?
        - Can it be improved? Is there anything that could be changed about the model?
        - Is there anything odd about the model results.
    - Save the confusion matrix and feature importance bar chart to add them in the executive summary for the leadership team.

Version 1.3.0 (21-06-2024)

- Repeat the Analysis stage to gain further insights using the df1 data frame, the steps done were:
    - Create the following functions:
        - dataframe_outlier_producer which takes 3 column names of another dat frame (1 continuous numerical column and the other 2 can be discrete numerical or categorical) as arguments to produce a data frame which has the following 4 columns: the 2 discrete or categorical columns, the number of outliers and observations for each possible combination of the first 2 columns.
        - bar_boxplot_producer which takes 7 arguments (2 data frames, 3 columns names and 2 strings) to produce:
            a. A box plot using the first data frame, the first column is the y variable, the second is the x variable and the last one is used to produce side by side boxes based on the third column value. Furthermore, the first string given as argument is the title of this plot.
            b. A bar chart using the second data frame which is produced by the previous function, using the last 2 column names for the x-axis and produce side by side bars. Furthermore, the second string given as argument is the title of this plot.
    - Use the functions created in the previous step to produce frames which have number of outliers and observations to then create bar charts and box plots of the following variable combinations:
        - Satisfaction Level, Salary and Employee Lost.
        - Satisfaction Level, Projects Assigned and Employee Lost.
        - Satisfaction Level, Tenure Years and Employee Lost.
        - Satisfaction Level, Work Accident and Employee Lost.

  - Add a binary column to the df1 data frame called overworked whose value is True if the monthly work hours is higher than 176 and False otherwise.
  - Modify the overworked column in the following ways:
    - Replace the True and False values by 1 and 0 respectively.
    - Change the data type of the column from object to integer.
  - Produce 3 scatter plots (the last 2 are side by side in the same axis):
    - One for the satisfaction level vs the number of work hours per month where the points are different depending if the employees left or stayed in the company.
    - One for the satisfaction level vs the number of work hours for workers who remains in the company.
    - One for the satisfaction level vs the number of work hours for resources who left the company.

Note: each of the previous scatter plots has a red vertical line at the value of 176 work hours to separate employees who overworked.

- Produce a data frame for the values of the employee_lost feature with 2 columns: the first has the count of overworked employees and the second has the total observations.
- Produce 2 side by side scatter plots of the satisfaction level vs the average hours worked per month with a red vertical at the value of 176 hours (one plot for each value of the promotion_last_5years variable for lost employees). The values on each side of the line are from colored differently to separate overworked employees.
- Reproduce the same plots as the previous step but for employees who remain in the company.
- Plot a box plot of the satisfaction level vs the department for each value of the employee_lost feature.
- Use the count_bar_chart function to produce 3 side by side bar charts using the tenure years in the x-axis and the type of salary as the legend for the bars. The differences between the 3 plots are the following:
  - The first plot has all the observations.
  - The second plot has only the count of employees who remain in the company.
  - The third plot has only the count of employees who left in the company.
- Create the following box plots, each one with a red horizontal line was added at the value of 176 work hours and the employee_lost feature used to create side by side boxes for every value of the x variable:
  - Monthly work hours vs the tenure years.
  - Monthly work hours vs promotion over the last 5 years.
- Use the count_bar_chart function to produce:
  - A side by side bar chart with the overworked variable in the x-axis, the count of the values in the y-axis and the promotion_last_5years variable as the legend for the side bar of each x-value.
  - Two side by side bar charts using the department in the x-axis and the promotion over the last 5 years as the legend for the bars. The difference between the plots is that the first one uses all the observations of the df1 data frame and the second uses only the rows of employees who left.
- Create a heat map correlation matrix of the all numerical and binary variables of the df1 data frame except the average_monthly_hours feature.
- Recheck the no multicollinearity assumption but without taking into account the average_monthly_hours variables.

Version 1.4.0 (25-06-2024)

- Complete the second run of the Construct stage of the PACE workflow. The tasks done were:
  - Repeat the 2 to 15 steps (except the ones where functions are written) done in the first run of the Construct stage while using the same dictionaries and functions used in the first run of the stage to test how the performance of the first models changed when the satisfaction feature was no used in the model tests and the overworked variable was used instead of the average_monthly_hours. Other differences between this run and the first one are:
    - The data frame used to create the models was called df_dummy2.
    - The data frame of seventh point was modified to include the metrics of the new models and compare them with the ones of the old models.
    - The winning model for the validation test was the XGBoost model.
  - Use the plot_importance function of the xgboost package with the XGBoost model as the argument to create a plot of the features importance in the model.

Version 1.5.0 (26-06-2024)

- Complete the Execute stage of the PACE workflow. The tasks done for this were:
  - Answer the following questions before writing the model interpretation and recommendations:
    - What key insights emerged from your model(s)?
    - What are the criteria for model selection?
    - Does my model make sense? Are my final results acceptable?
    - What are my ethical obligations during this stage.
  - Answer the questions at the beginning of the Execute section in the notebook.
  - Write the conclusions, business recommendations for the executives and next steps to take after sharing this information with them.

Analysis Summary

Each stage of the PACE workflow excluding the creation of the project proposal and the executive summary were done in a Jupyter notebook using the Python programming language for the following reasons:

1. The extensive number of libraries of Python to perform tasks like data manipulation and transformations, data visualization, statistical tests and machine learning models development and testing.
2. The easy syntax of the language which can help to easily explain the codes and functions used in the analysis.

Note: for this project, the Analysis and Construct stage were repeated twice to improve the model's efficiency. The tests, visualizations and modifications done during each round of the stage will be presented in order.

Plan Stage

The analytical part of this stage involved getting familiar with the data set by performing basic EDA to study the variables present in the set and get basic information about them like the number of features, their types, the value they have and their distributions.

To achieve this, the following steps were done in a Pandas data frame which contains the loaded data set:

- The number of features, their types and number of observations were obtained using the info function of Pandas which provided the following information:
  - There are no missing values in the data set.
  - The satisfaction_level and last_evaluation variables are floats, Department and salary are stored as objects/strings, and the remaining are integers.
  - There are 14999 observations in this data set.
- Descriptive statistics were computed using the describe method of Pandas. This showed the following:
  - For the numerical variables, none of their minimum values are lower than 0 which implies there are no typos in them based on the context.
  - The maximum values for the satisfaction_level and evaluation_score variables show that the scales are correctly given for every value.
  - For the Work_accident, left, and promotion_last_5years variables, their maximum value is 1 and their minimum values and other quartiles (rows whose index is 25%, 50% and 75%) are 0 which implies that those variables are binary since their only values are 0 and 1. Also, by looking at their means, it can be seen that the value of 0 is the one that was captured mainly on those features.
  - The Department and salary features have 10 and 3 classes respectively (shown by the row whose index is unique which counts the number of classes for categorical features) with sales and low being the most repeated classes for each respective feature (shown by the row whose index is top which shows most repeated class).
  - Modify the name of some features to follow snake_case notation and display a name which indicated the information they store.
  - Check for missing values using the isna and sum methods of data frames. This helped to confirm that there are no cells in the data set with missing information.
  - Check for duplicated rows using the duplicated method. After running this method on the data frame, the presence of duplicated rows was discovered and a new data frame named df1 was created using the drop_duplicates method. The results of this was that the new data frame had 3008 rows removed from it.
  - Check for outliers in the variables which store the tenure years, projects assigned and average monthly work hours for the employees by building box plots for each feature. Those plots revealed the following insights:
  - There are no outliers in the average_monthly_work nor the project_assigned features.
  - The maximum and minimum number of projects assigned to employees are 7 and 2 respectively.
  - The maximum and minimum number of average work hours per months was slightly higher than a 300 and close to 100. The actual values which were shown in the descriptive statistics frame created before are 310 and 96.
  - The values which were higher than 6 for the tenure years are considered as outliers. After computing the number of rows where the tenure years are 6 or higher, it was discovered that this feature had 824 outliers.

Analysis Stage, first run

This stage involved continuing the EDA started in the previous stage of the workflow by creating data visualizations and doing data transformations to study the relationship of the predictor variables with the target variable (employee_left).

To achieve this, the following steps were done using the df1 data frame:

- Compute the number of observations for the 2 different values of the predictor variable and their percentage of the data using the count_values method. This revealed the following:
    - The data set is composed of 10000 employees who remain in the company (represented by the value of 0 of the feature) and 1991 employees who left after taking the HR survey (represented by the value of 1 of the feature).
    - The percentage of employees who stayed and those who left is of 83.4% and 16.6% which indicates a class imbalance in the data.
- Compute descriptive statistics of the predictor variables for each different value of the predictor variable.

Note: for all the following visualizations, the x variable is a predictor feature, the y variable is the number of occurrences with the number given at the top of the bar for side by side charts.

- Create side by side bar charts which show the number of employees who stayed and left for each group of the 2 categorical variables. Those plots revealed the following:
    - For the salary feature, employees with low or medium salaries are the ones most likely to quit their jobs while those who have a high salary prefer to remain in the company.
    - For the department feature, the sales, technical and support departments are the ones with the greatest number of employee exits. Furthermore, the randID and management departments are the ones with the lowest exit rates.
- Use the count_bar_chart function to plot visuals of the following features (the insights revealed by each plot will be written after the feature used):

Note: for the binary features which are promotion_last_5years (if the employee was promoted or not over the last 5 years) and work_accident (if the employee had or not an accident in the office), the percentage of observations for each value was given for each option of the target variable below the bar chart using pandas series.

- Work Accidents: workers who suffered accidents in the office are more likely to leave the company.
- Promotion on Last 5 Years: employees who haven't been promoted after 5 years in their positions are more likely to leave the company.
- Projects Assigned: if the employees have between 3 and 5 projects to which they contribute, there is a good chance that they will stay but if the have less or more projects assigned to them, this will increase the possibilities of them leaving the company.
- Tenure Years: workers usually leave the company after 3 years of service on it. Furthermore, there were little exits of employees who work for less than 3 years and no exits for those who have been more than 6 years.
- Create 2 histograms (one per value of the employee_left variable) for the remaining numerical features. The information displayed by them is:
    - Satisfaction Level: people with a high satisfaction level (higher than 0.5) are the ones most likely to stay in the company while most of the employees which had a satisfaction lower than 0.5 left.
    - Last Performance Evaluation Score: there is no clear relationship between the score of the employees last performance evaluation and them staying or leaving the company. This will have to be studied and analyzed further.
    - Average Monthly Work Hours: Employees who have between 150 hours and 260 hours of works are the ones more likely to stay in their positions while those who have more or less are more likely to leave.

- Confirm if the multicollinearity assumption of a Logistic Regression model (no high correlation between the predictor variables) is satisfied by computing the Variance Inflator Factor (VIF) of the predictor variables. This step revealed the following:
  - The value of the Variance Inflator Factors (VIF) are higher than 5 for each non-binary feature which implies that there is a high collinearity between the numerical features. Therefore, the multicollinearity assumption is not satisfied.
  - Two assumptions of the Logistics Regression model are not satisfied by the data set. To change this, the outliers would need to be dropped and some predictor features would need to be removed too. However, this could eliminate value information hidden in the data.
- Perform feature engineering in the categorical features of the data set by:
  - Capitalizing the classes.
  - Building a new data frame named df_dummy which has binary features for each class of the categorical features using the get_dummies function in the df1 object.

Construct Stage, first run.

This stage involved the following 3 sections in the order given:

- Model Research.
- Hypothesis Testing.
- Model Development and Selection.

Model Research

There are many classification models that could be built to achieve the goal. Some of those are:

1. Logistic regression model.
2. Bayes machine learning model.
3. Decision Tree.
4. Random Forests.
5. XGBoosting.

From the list above, the Logistic model can't be used since 2 of its assumptions (no extreme outliers in the data and no multicollinearity between the predictor variables) are violated.

The Bayes model may not be a good model to use since there are many types of Bayes models, all determined by the type of predictor variables that are present. The original data set has 2 categorical variables, 2 binary variables and 5 numerical variables. Therefore, there are not enough variables of the same type to build a Bayes prediction model.

Decision Trees, Random Forests and XGBoosting are models which don't require too much data preprocessing, can handle outliers well and any type of data (discrete, continuous and categorical). Therefore, those 3 types of model could be used to build the target model.

Hypothesis Testing.

The visuals revealed that most of the numerical discrete and continuous predictor variables have a relationship with the target variable. One of those variables didn't show a clear relationship. To fully confirm this, hypothesis tests will be conducted on the means of the variables to compare the values of employees who left and stayed in the company.

The significance level is 0.05 for each of the following tests. Furthermore, the null hypothesis will be rejected if the p value is lower than the significance level.

For each numerical variable, the null hypothesis and alternative hypothesis are:

- Null Hypothesis: the mean of variable x for employees who left the company is equal to the one of those who remained in the job.
- Alternative Hypothesis: the mean of variable x for employees who left the company is not equal to the one of those who remained in the job.

The null hypothesis was rejected for the following variables:

1. Satisfaction level with the mean of employees who stayed being higher.
2. Projects Assigned with the mean of employees who left being higher.
3. Monthly Work Hours with the mean of employees who left being higher.
4. Tenure Years with the mean of employees who left being higher.

The only variable for which the null hypothesis was not rejected is the Evaluation Score of the last performance test. However, this doesn't mean that it is concluded that the means of both scores are equal, just that there is not enough statistical significance to reject the null hypothesis.

Model Development and Selection.

In this section of the stage, the 3 models were trained using a training portion of the complete data while performing hyper parameter tuning using Grid Search Cross Validation based on the parameters that returned the higher F1 score. Then, the a validation set which had 20% of the entire data set was used to compare the models and select the one which produced less errors with the validation set. Finally, the most efficient model was retrained both with the training and validation sets and tested on the remaining 20% of the data.

After the development and selection process. The model that produced the best results was the Random Forest which produced the lower number of False Positive and False Negative errors in the validation set with the F1, recall and precision scores higher than 0.9 which indicate a very low amount of errors and high confidence in the model.

After the Random Forest was selected as the most efficient model, it was trained using both the original training set and the validation set and tested in new data, which in this case is the test set. During the testing portion, the model performed extremely well with the test data and returned little errors which indicates that this model is excellent to predict whether an employee will remain or not in the company. Furthermore, it returns lower false positives which assists another goal of the project which is to save money that would be used to research for replacements.

Finally, a bar chart graph which indicates the most important features for the model was created. This visual revealed that the most important features for the model are satisfaction level, projects assigned, tenure years, evaluation score of the last performance review, average monthly work hours and if the salary of the employee is low or not.

Analyze Stage, second run.

For this second execution of the Analyze stage, the relationships between the predictor variables and the target variables were studied further and more carefully to see if a model whose results are more efficient than the one created in the Construct stage could be produced.

This stage involved the following 2 sections in the order given:

- Factors that contribute to employee satisfaction.
- Other relationships.

Factors that contribute to employee satisfaction

Note: the legend of the visuals in this section are the values of the employee_lost feature, the y variable of the box plots and scatter plots (except the side by side scatter plots) is the satisfaction level while the one of bar charts is the number of repetitions.

The goal of this section was to further investigate how the predictor variables contribute to an employee's overall satisfaction. To achieve this, the following tasks were done:

- Use the dataframe_outlier_producer function to create new data frames which later were used to plot bar chart and box plot visuals of the following features (the insights revealed by each plot will be written after the feature used):
  - Satisfaction level and salaries given. Those plots and their data frames revealed that employees who left were more dissatisfied with their salary than the ones who stayed which is shown by the median being close to 0.4.
  - Satisfaction level and projects assigned to an employee. Those plots and their data frames revealed the following insights:
    - There are little to no outliers for the box plots of employees who stayed. Furthermore, the satisfaction of those workers is higher in average when they have less than 6 projects assigned to them.
    - There are many outliers for the box plots of employees who left. Furthermore, the satisfaction of this people is higher in average when they have less than 6 projects assigned to them.
  - Satisfaction level and tenure years of an employee. Those visuals and their data frames helped to discover the following information:
    - The box plots of employees who stayed have little to no outliers, and most of the boxes except for the group of 5 and 6 years, have both the medians (the middle line in the box) and first quartiles (bottom of the box) whose value of satisfaction is higher 0.5.
    - The box plots of employees who left have many outliers. Furthermore, the levels of satisfaction change for every year of tenure with the group of 5 and 6 being the ones which show highest medians and first quartiles.
  - Satisfaction level and whether a worker had or not an accident at the office. Those plots and their data frames revealed the following insights:
    - There are no outliers in the data frame for work accidents.
    - Having a work accidents doesn't necessarily affect the amount of satisfaction of employees
    - The satisfaction based is higher employees who stayed even if they had an accident at work.
- Plot scatter plots where the x variable is the average hours worked per month. The first plot has all the points while the second and third are shown side by side (one per value of the employee_lost feature). Furthermore, each one has a line at the value of 176 to separate people who overworked.

- Build a data frame which shows the total employees who overworked in the first column and the total amount of people in the data set in the second column for each value of the employee_lost feature. This data frame combined with the 3 scatter plots created before, revealed the following insights:
  - For the employees who left, they are grouped mainly in 3 sections (with the remaining observations being scattered): first, those who have a satisfaction between 0.35 and 0.45 worked less than 175. Then, the ones that have a satisfaction higher than 0.75 and worked more than 215 hours. Finally, those who worked more than 235 hours and have a satisfaction lower than 0.2.
  - For the employees who stayed, they are grouped mainly in 2 sections (with the remaining observations being scattered): those who have satisfaction lower than 0.3 and those who have it higher than 0.45.
  - More than 60% of the employees who stayed or left, have worked more than the expected amount of time.
- Plot 2 side by side scatter plots where the x variable is the average hours worked per month and the legend is the value of the overworked feature. The first 2 plots are for lost employees who where promoted (left plot) and not promoted (right plot), the last 2 plots are equal to the first 2 but they use the information of current employees. Furthermore, each one has a line at the value of 176 to separate people who overworked. Those visuals show that being promoted doesn't necessarily increase the satisfaction level of employees.
- Create a box plot where the x variable is the department name. This visual revealed the following insights:
  - For employees who remain in the company, their satisfaction overall is good since each box has the first quartile higher than 0.5 and the median is close to 0.7 in every department.
  - For employees who left the company, their satisfaction overall is low since each box has the first quartile between 0.15 and 0.45 and the median lower than 0.5 in every department. Furthermore, the departments with lowest satisfaction for this group are Accounting, Technical, Support, Management, RandD and IT.

Other relationships.

The goal of this section was to investigate how the predictor variables other than satisfaction level are related to each other. To accomplish this, the following tasks were done:

- Use the count_bar_chart to create 3 side by side bar charts where x variable are the tenure years and the legend is the type of salary. The first chart was built using the entire data set, the second with the data of current employees and the third one with the information of lost workers. The 3 bar charts together provide the following insights:
  - Employees who have spent less than 5 years in the company tend to have low (mainly) or medium salaries. Furthermore, there is little people on this group who have a high salary. Finally, workers with more than 6 tenure years have mainly medium salary and the rest have either low or high salary.
  - The trend is the same for employees who stay in the company. Although the difference between the number of people with medium and low salaries based on the years of tenure is lower compared to the entire data set.
  - For the employees who left the company, most of them had low or medium salary each year. Furthermore, only those which had high salary after 3 or 5 years of tenure left.
- Build the following box plots where the y variables is the average work hours per month along with a red horizontal line at the value of 176 hours and the legend is the value of the employee_lost feature:

    - A plot where the x variable is the tenure years. This helped to discover the following information:
        - Most of the workers did overwork (worked more than 176 hours per month) despite their tenure years which is shown by most of the medians being above the red line.
        - For workers who stayed, their amount of work is similar through the years while for the ones that left, many of them had more work except on their third year.
    - A plot where the x variable is the promotion variable. This revealed that being promoted doesn't necessarily increase or reduce the number of work hours per month for employees who stay.

- Use the count_bar_chart to create 3 side by side bar charts where the legend is the promotion_last_5years feature. The difference are the following:
    - The x variable of the first chart is the overworked feature. This helped to discover that overworking doesn't necessarily help people to get promoted in the company because very few employees who overworked got a promotion on the last 5 years.
    - The x variable of the 2 remaining charts is the department of the company. Also, the second chart shows the information of the entire data set and the third one only shows the data of lost workers. They revealed the following insights:
        - Almost all the employees who left hadn't been promoted in their departments over the last 5 years.
        - For each department, close to one sixth of the total workers in the data set, left their positions.

- Compute a correlation matrix of the non categorical variables (except the average_monthly_hours). This matrix provided the following insights:
    - Satisfaction level has little positive correlation with the last evaluation score, little negative correlation with the projects assigned and tenure years of employees, and little to no correlation with the remaining variables.
    - The evaluation score of the last performance review is positively correlated with the projects assigned and the overworked features and has little positive correlation with the tenure years.
    - The projects assigned to an employee have a positive correlation with the tenure years and the overworked variable.
    - The tenure years have little to no correlation with the work_accident, promotion over the last 5 years and the overworked variable.
    - Both the work accident and the promotion variables have little to no correlation with the other variables.

- Recheck the no multicollinearity assumption for the non categorical features but without the satisfaction level features and replacing the average monthly hours feature by the overworked feature. Compared to the time that the VIF values were computed in the first run of the Analysis stage, the values for the second run are lower when removing the satisfaction_level feature and using the overworked variable instead of the one which has the average monthly hours. However, the first 3 values are all still higher than 5 which implies that there is correlation between those variables and the no collinearity assumption is still violated.

The reason for the change is to check how the models are affected when satisfaction is removed. Furthermore, the overworked variable was created from the average_monthly_hours variable and won't be used to develop the new models due to their correlation.

Construct Stage, second run.

Just like in the first run of the Construct stage, the 3 models were trained using a training portion of the complete data while performing hyper parameter tuning using Grid Search Cross Validation based on the parameters that returned the highest F1 score. Then, the a validation set which had 20% of the entire data set was used to compare the models.

Unlike the first models, the new ones returned more errors (both false positives and more negatives) and lower scores on the evaluation metrics but still close to 0.9. Furthermore, after comparing the models with the validation set, the winning model was the Random Forest with a F1 score of 0.90534 and 48 errors from which 37 are false positives. The XGB had a F1 score of 0.90148 and 50 errors from which 32 are false positives. Since another the goal of this model is also save money used to search for replacement candidates, the XGB model is the one that accomplished this objective better. Therefore, it was selected as the winning model.

After the XGBoost was selected as the most efficient model, it was trained using both the original training set and the validation set and tested in new data, which in this case is the test set. During the testing portion, the model performed extremely well with the test data and returned more errors than the most efficient model created before but still performed very well since the evaluation metrics were all equal or higher than 0.89 which indicates that this model predict well whether an employee will remain or not in the company.

Finally, a bar chart graph which indicates the most important features for the model was created. This visual revealed that the most important features for the model are evaluation score of the last performance review, tenure years, projects assigned, whether the worker did overwork or not and their salary is low or not.

Another difference between this model and the previous one are the importance of the features since for the first model the most important feature was satisfaction and for this one was the score of the performance review. Furthermore, in the first model, most of the binary features have little to no importance but in the second one, they are slightly more important.

Most Relevant Conclusions

1. The random forest machine learning model which includes the satisfaction level of the employees is a highly efficient method to predict whether an employee will leave or remain in the company based on the answer he/she provides to the survey given by the HR department. While some errors are produced, the predictions are almost completely accurate.

2. In the case that satisfaction and work hours of employees can't be recorded at all for the future, use the XGBoost model instead since it give very good results when predicting if an employee will stay or leave.

3. The most important features that determine if an employee will leave or remain in the company are: satisfaction level, projects assigned, tenure years, evaluation score of the last performance review, average monthly work hours and if the salary of the employee is low or not.

4. Some factors that influence the satisfaction of people at the company are the number of projects to which they contribute, their salary and the department in which they work.

Recommendations

1. Reward employees who overwork to increase their satisfaction like promoting them or give them additional payment for the extra amount of hours they. However, give a limit to the number of hours

that can be done per day or week since many employees who left worked close to 300 hours per month which is close to 13.5 hours per business day.

2. Give promotions to employees starting after their 3 first years of work based on their overall performance.
3. Improve the satisfaction of departments with higher employee losses and with lower employee satisfaction.
4. Gather further information from the employees to improve the model's performance.
5. Deploy the model since the number of errors produced by it is small.
6. Limit the number of projects to which a worker can simultaneously contribute to avoid work overloads.
7. Hold company-wide and within-team discussions to understand and address the company work culture, across the board and in specific contexts.

Next Steps

1. Assign a supervisor to monitor the model performance after deploying it.
2. Perform research for new attributes that contribute more to employee satisfaction to increase employee retention.
3. Add the suggested features and more observations to the data set if possible to improve the model's performance.
4. Revisit the model's performance after a year to confirm if it achieve the desired result.