DOKUZ EYLUL UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING

CME 4416 INTRODUCTION TO DATA MINING
FINAL REPORT

by
2016510007 - Asude AĞAYA
2017510025 - Zekiye DOĞAN

Lecturer
Ass. Prof. Feristah DALKILIÇ

January, 2021
İZMİR

CONTENT

1. INTRODUCTION

The aim of this project is to show the factors that cause the employees to wear out and leave the job by performing operations on the selected dataset (from Watson Analytics data). "What is the effect of the job role on attrition?", "What is the breakdown of the employee's overtime on attrition?" To enable the discovery of answers to important questions such as.

First, the dataset was examined in detail. The distribution of the features, the relations between each other and the target were analyzed. The presence of missing value, outlier and duplicate data in the dataset was checked and its effects on it were investigated. In the continuation of these processes, it was tried to obtain features that could be more effective on the model from the existing features. Classification models were used to classify the data. And since the target value is included in the train dataset, it was supervised. Target takes values 0 and 1.

2. DESCRIPTION OF DATASET

The dataset used is the IBM in Employee dataset. The dataset we are working on is a small dataset. However, considering the problem, this dataset is at a normal level. In total, there are 35 features and 1470 instances with target. Some example features: Education, JobLevel, PerformanceRating, MonthlyIncome, PercentSalaryHike.

Then, a detailed analysis of the data was started. The review was initially started manually. In this analysis, it was examined which data types the data had (categoric or numeric) and it was observed in which range the values were. The meanings of these values were examined. For example:

Education

1. 'Below College'

2. 'College'

3. 'Bachelor'

4. 'Master'

5. 'Doctor'

As a result of manual trials, it was tried to predict which features would be more effective. During manual testing, two instances were randomly selected and a comparison was made between these features. The features that were thought to be effective as a result of the comparison were EnvironmentSatisfaction, MonthlyIncome, TotalWorkingYears.

When viewed from a statistical view, EmployeeCount, Over18, and StandardHours and EmployeeNumber values were fixed, so they were deleted from the dataset as they would not have any effect on future predictions.

The minimum, maximum, mean and standard deviation values of each feature were examined by making a data quality report. One of the reasons for doing this is to determine the features that may contain outliers by looking at which values the data are in, mean and standard deviation.

Then, the analysis report was obtained by using the sweetviz library through the Spyder tool. In this analysis report, the features of each feature were examined in detail and the relationship of these features with other features was observed statistically. The most important features when examining are the correlation relationship of the features with the target (Attrition), their distribution and how effective they are on the dataset. Thus, an idea was gained on both obtaining information about the feature and manipulating that feature.

To give an example of analysis,

In terms of numerical association, Age has the highest correlation with TotalWorkingYears, while in categorical association it has a 52% relationship with JobLevel and 16% with Attrition.

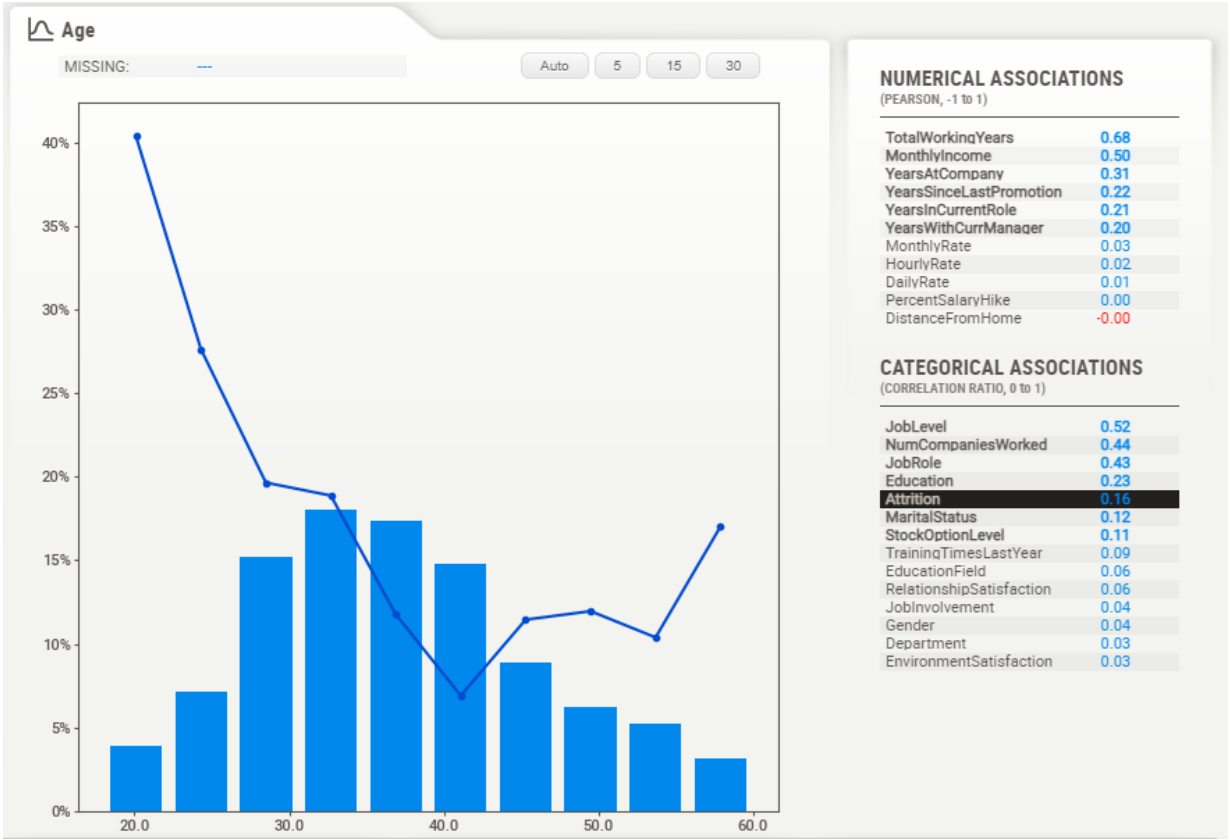

Figure 1: Analysis for Age Feature

It has been observed that the salaries of the employees vary according to the JobLevel and the departments they are in. It is seen that if the JobLevels of the employees in the same department are different, if the salary of a person with a higher level is less, the tendency to leave the job increases.

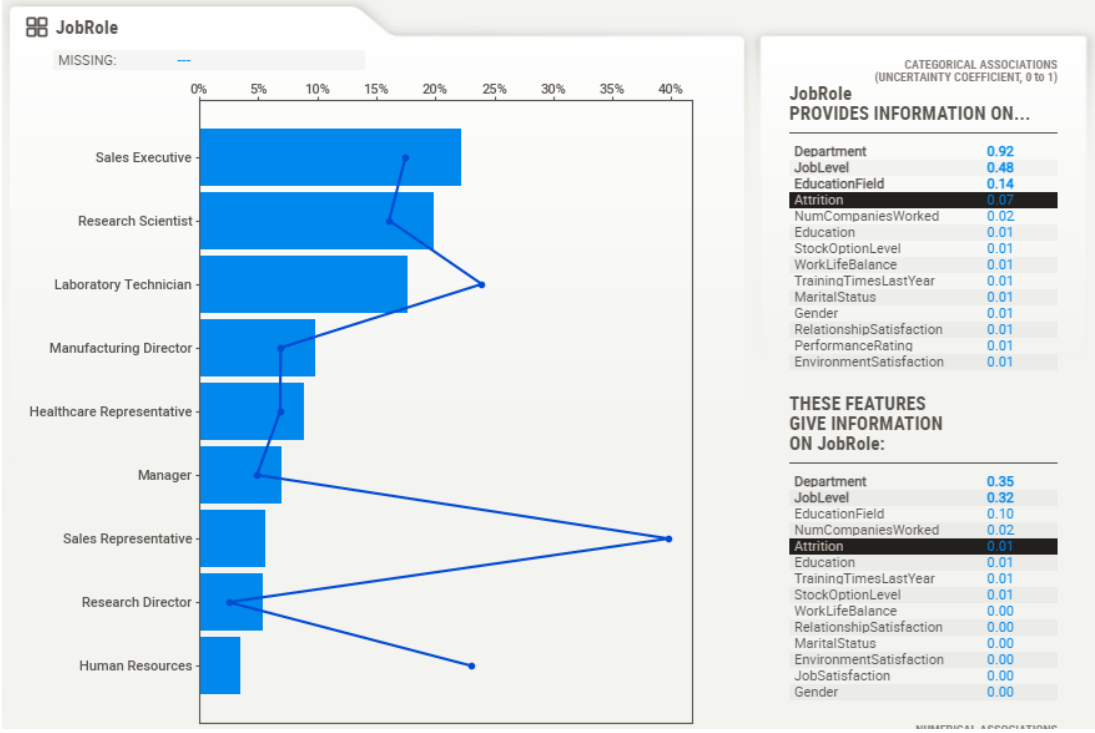

Figure 2: Analysis for JobRole Feature

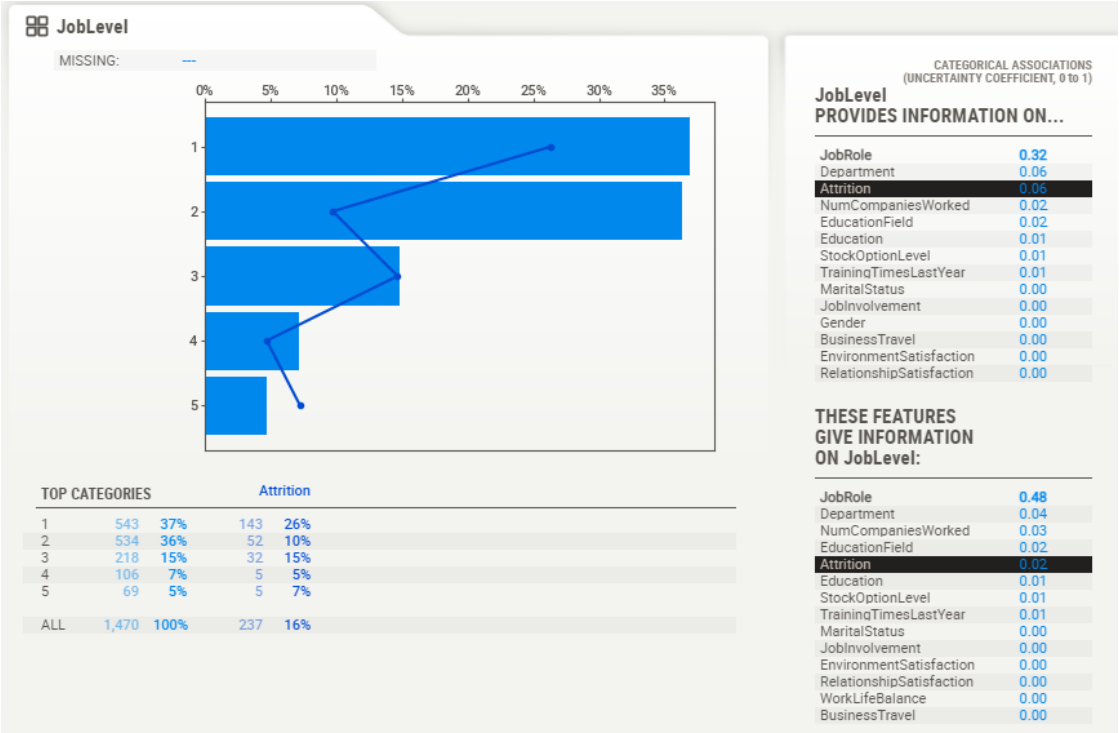

Figure 3: Analysis for JobLevel Feature

When TotalWorkingYears is examined, it is seen that it is one of the values with the highest relationship with Atrrition. The high relationship with JobLevel indicates that the job level also rises depending on the time spent at the workplace.

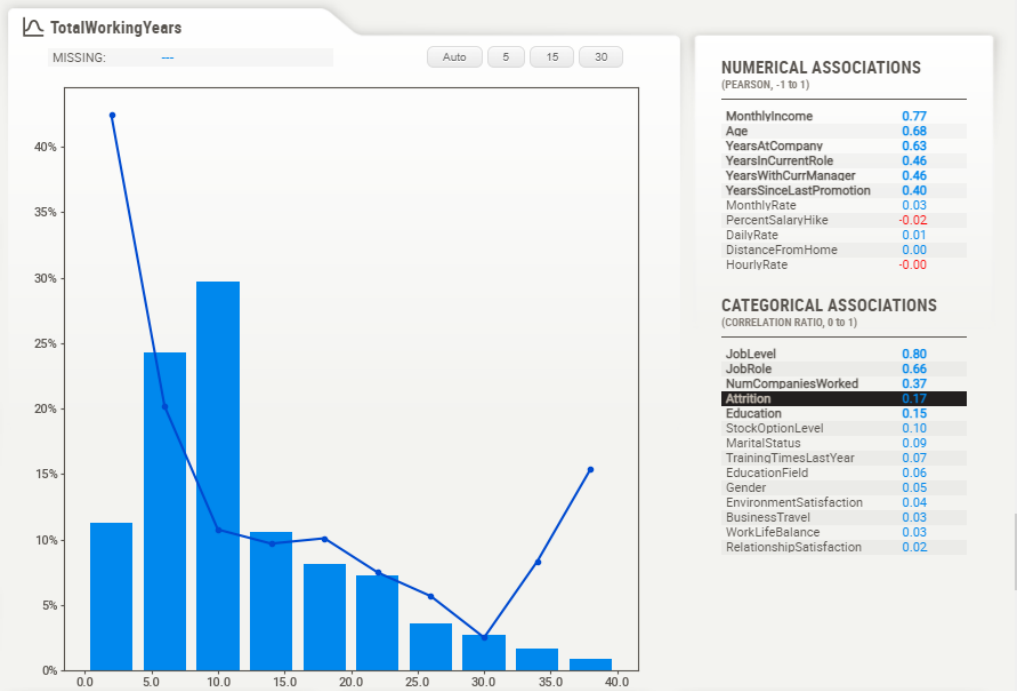

Figure 4: Analysis for TotalWorkingYears Feature

It has been observed that DailyRate, HourlyRate and MonthlyRate have a relatively small effect on Attrition, while they have a negative relationship with YearsWithCurrManager.

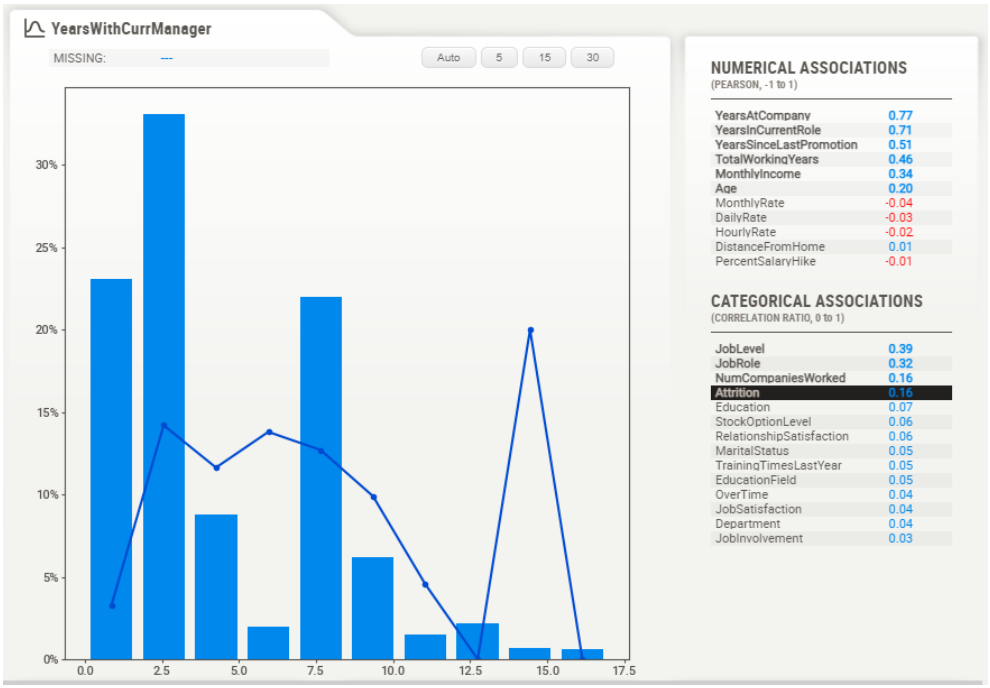

Figure 5: Analysis for YearsWithCurrManager Feature

Then the correlation matrix of the dataset was obtained. This matrix contains both categorical and numerical features. The information became more understandable thanks to the graphic that summarizes the relationships between them and helps visually. Looking at this, it was seen that the features that affected Attrition the most were Age, BussinesTravel, JobInvolvement, JobLevel, JobRole, MaritalStatus, MonthlyIncome, OverTime, StockOptionLevel, TotalWorkingYears, YearsAtCompany, YearsInCurrentRole and YearsWithCurrManager.

In the Correlation table, it is understood that the DailyRate, DistanceFromHome, Gender, HourlyRate, MonthlyRate, PercentSalaryHike, PerformanceRating, RelationShipSatisfaction, TrainingTimeLastYear, WorkLifeBalance features have a very weak relationship with other features and Attrition. Therefore, it is not planned to be included on the model when estimating. While the PercentSalaryHike and PerformanceRating features have interactions with each other, they have little or no relationship with other features.

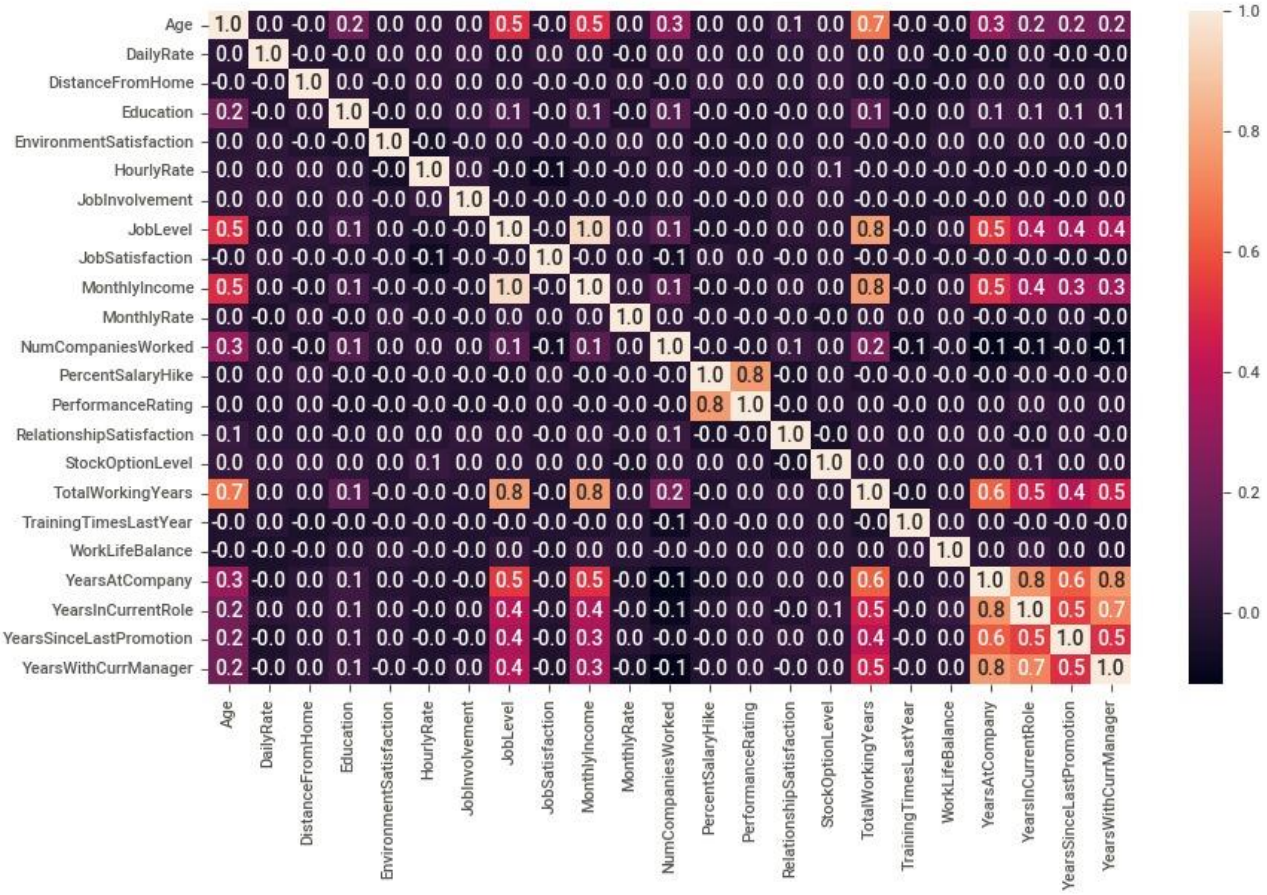

Figure 6: Correlation Table

After the correlation analysis, it was checked whether there was a missing value in the dataset. No missing value was found in any instance. Then, it was checked whether there was a duplicate on the data, but it was seen that there was no duplicate data.

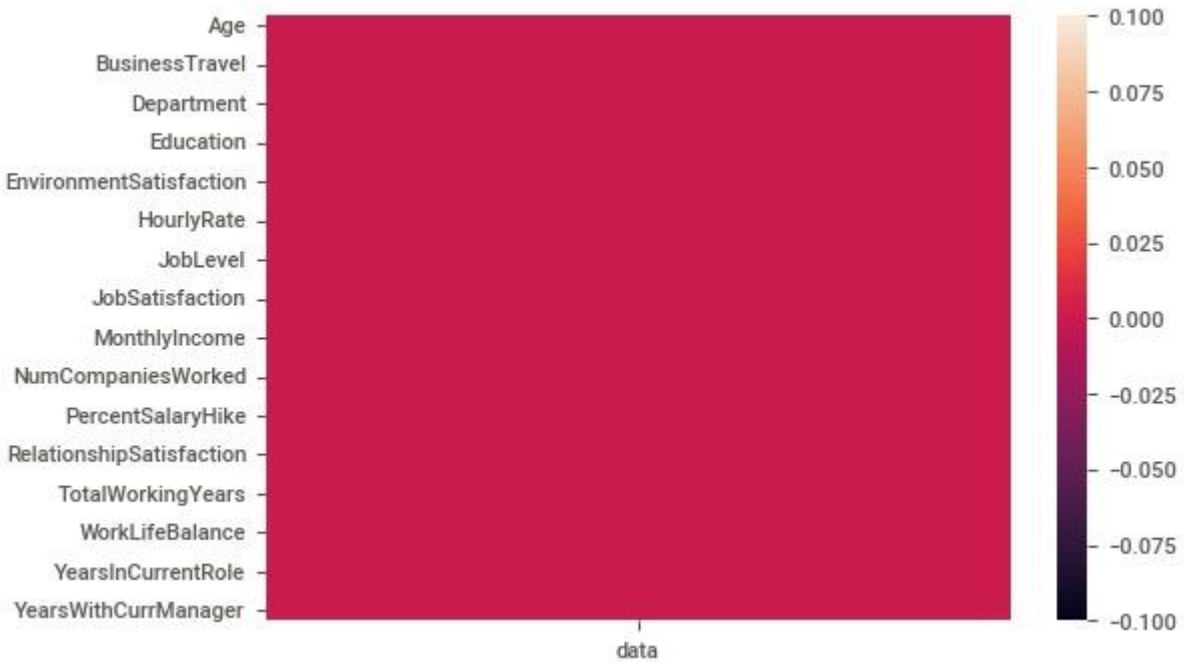

Figure 7: No Missing Value

3. DATA PROCESSING

First of all, some pre-processes should be applied to the dataset in order to be able to operate on the dataset. It was checked whether there was a missing value before, and therefore no action was taken for them. Previously, it was understood that there was no missing value in the data. For this reason, no action was taken against them. Then, boxplot was used to check whether there was an outlier in the dataset. Outlier detected in some features.

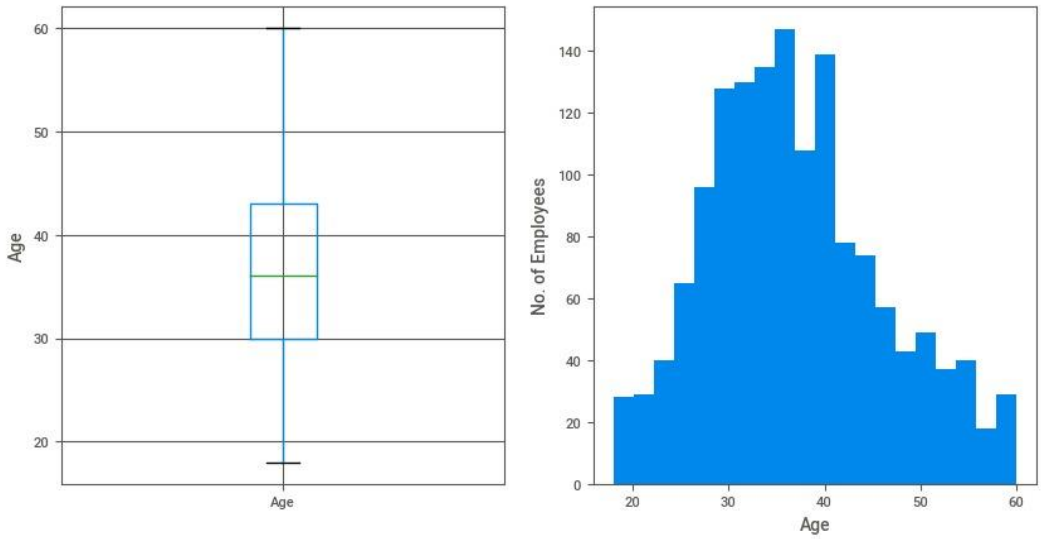

Figure 8: BoxPlot of Age Before Outlier Handling

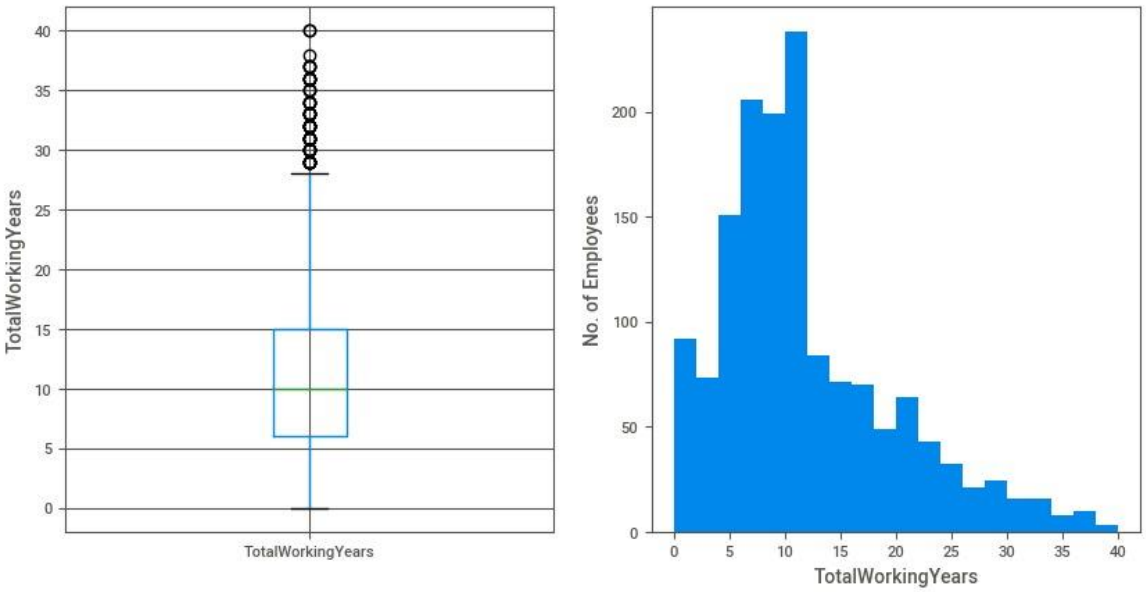

Figure 9: BoxPlot of TotalWorkingYears Before Outlier Handling

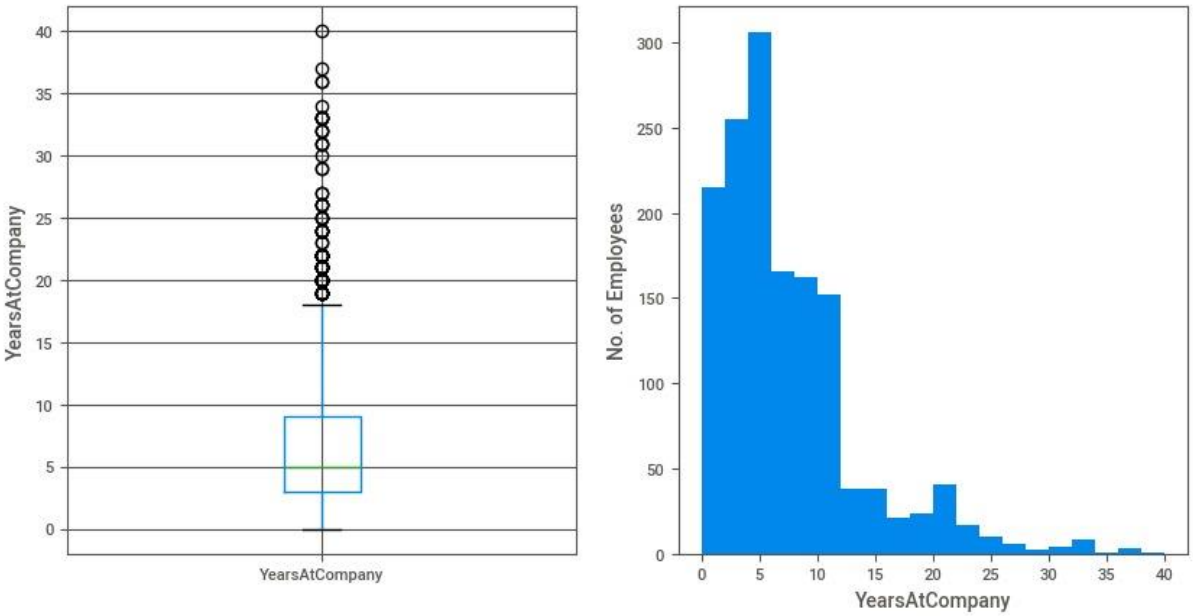

Figure 10: BoxPlot of YearsAtCompany Before Outlier Handling

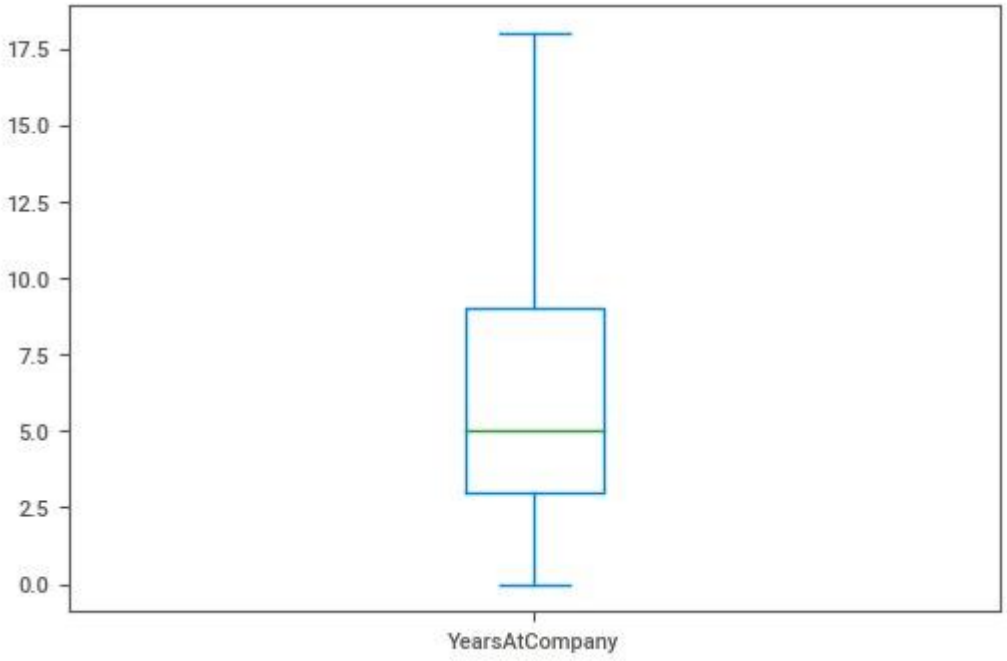

Figure 11: BoxPlot of YearsAtCompany After Outlier Handling

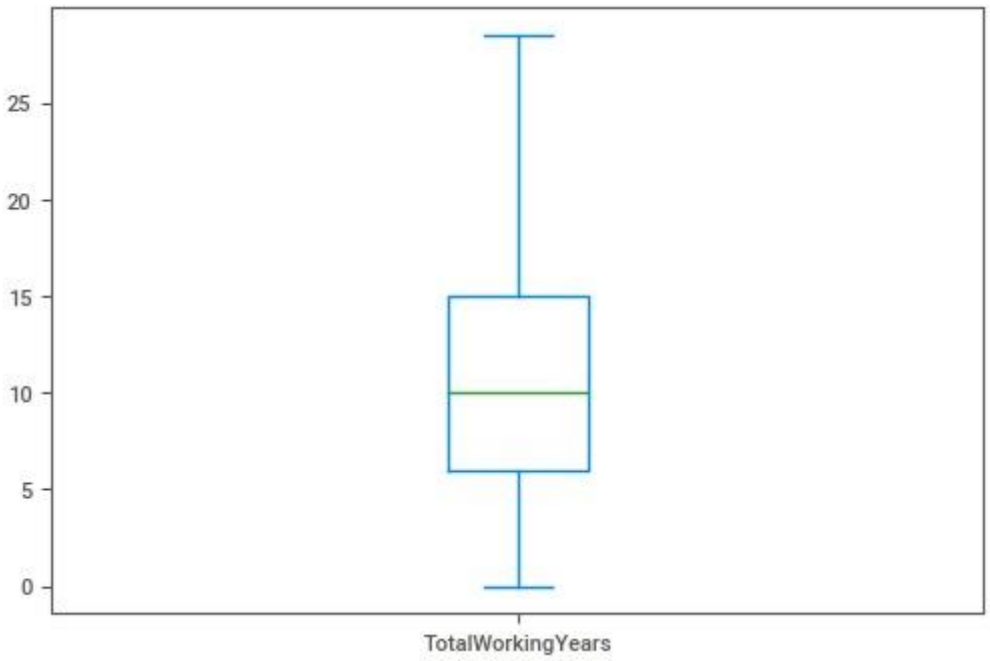

Figure 12: BoxPlot of TotalWorkingYears After Outlier Handling

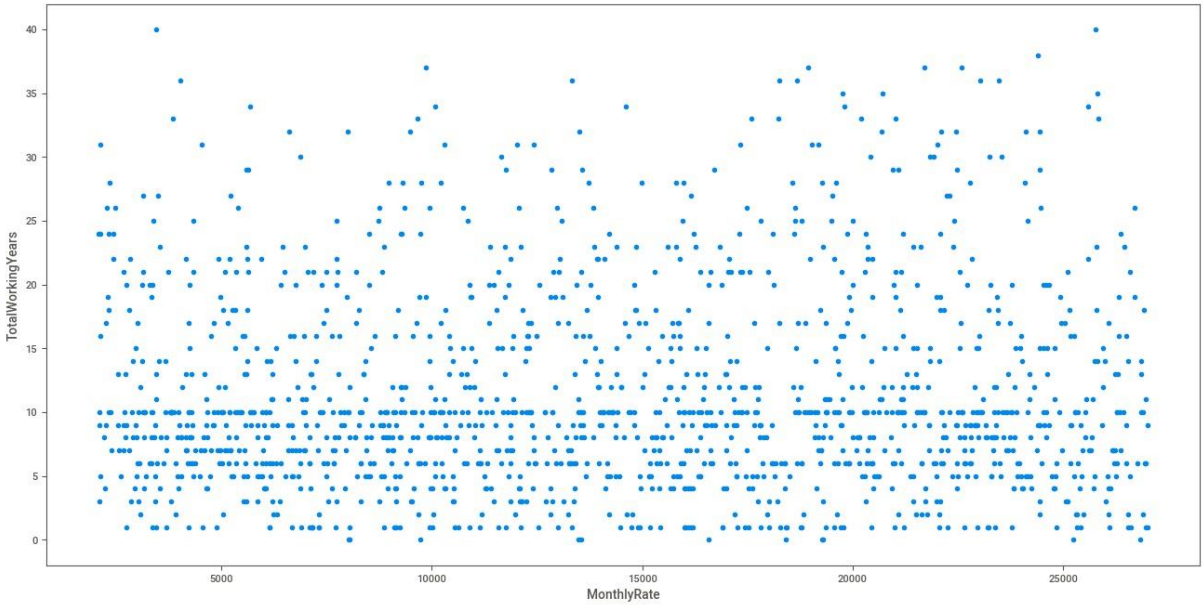

Figure 13: Scater of TotalWorkingYears and MonthlyRate

For discrete features, outliers were determined with the IQR (Inter Quartile Range) method and outlier values that were less than the minimum thresold value were set to the minimum value, and those that were greater than the maximum were set to the maximum value.

A new feature has been produced considering the relationship between MonthlyIncome and JobLevels. In this feature, MonthlyIncome values of instances with the same JobLevel were averaged. If MonthlyIncome is lower than this value, the new feature is given a Low value, and a High value if it is higher.

Range normalization applied on MonthlyIncome. The reason for this is that MonthlyIncome has higher values than other features. Because it can distort the accuracy of the result to be taken from the model by putting pressure on other data.

4. SELECTED ALGORITHMS

The dataset is suitable for supervised classification. Among the appropriate algorithms, Naive Bayes, RandomForest and Logistic Regression algorithms were chosen to use.

Why Naive Bayes?

Although the size of the dataset is sufficient for the purpose, it is a relatively small dataset. Algorithms have been selected considering this situation.

Naive Bayes is a classification model based on Bayes' theorem. It is a probabilistic approach to the pattern recognition problem that can be used with a premise that seems quite restrictive at first glance. This proposition is that each descriptive attribute or parameter to be used in pattern recognition should be statistically independent. The way the algorithm works is to calculate the probability of each situation for an element and classify it according to the highest probability value. It can produce very successful works with a little training data.

Naive Bayes was chosen because the Naive Bayes algorithm requires a small amount of training data to estimate the necessary parameters, as well as they are easy and quick way to predict the class of the dataset.

Why RandomForest?

Random Forest is a supervised learning algorithm. One of the important advantages of the algorithm is that it can be used in both classification and regression problems. It can be used to identify the most important feature among the features available in the training dataset. It creates random forests and combines them for a more stable and accurate prediction. When splitting a node, instead of looking for the most important feature, it searches for the best feature among a random subset of features. This results in a wide variety, which often results in a better model.

Since the dataset is small, it may be possible for the models to be developed to memorize the train and to have over-fitting. As a solution to this situation, it was decided to use the RandomForest algorithm. Random Forest Classifiers facilitate the reduction in the over-fitting of the model and these classifiers are more accurate than the decision trees in several cases.

Why Logistic Regression?

The target value of the dataset takes two different values, yes and no. With this in mind, we chose to use Logistic Regression. Logistic Regression is useful for understanding the effect of independent variables on a single outcome variable.

Logistic Regression is a regression method for classification. It is used to classify categorical or numerical data. It works if the dependent variable, namely the result, can only take 2 different values. It thinks that predictors are independent and there is no missing data. It is useful for understanding the effect of independent variables on a single outcome variable.

5. TOOLS AND ALGORITHMS USED

During the development of the project, the Spyder open source software in Anaconda was used and the Python programming language was used. The reason why Python was chosen is that it makes very complex operations easier to do with the libraries it provides.

Evolution metrics were made using 3 different algorithms in total within the IBM dataset used. These are Random Forrest Classifier, Logistic Regression, Naive Bayes. If the algorithms are to be defined,

Random Forrest Classifier is an algorithm that aims to increase the classification value by generating more than one decision tree during the classification process, which is an ensemble learning method. Individually created decision trees come together to form a decision forest. The decision trees here are randomly selected subsets from the data set to which they are connected. It consists of a combination of the Bagging method and the Random Subspace methods. It gives results in a short time. It removes noisy data. It produces results using not very small datasets and variables with a large number of class labels. The low correlation among the trees it creates makes the predictions more precise. Trees are constructed with selected bootstrap samples and m randomly selected estimators at each node separation. Care is taken to ensure that m (number of variables used in each node) is considerably smaller than the total number of estimators. Each decision tree created is left in its widest form and is not pruned. At each node, m variables are randomly selected among all variables and the best branch is determined among these variables. The number of m variables taken equal to the square root of the number of m variables generally gives the closest result to the optimum result. Gini index, lowest output row values, is the current index value for that row. The comparison between the

trees created is performed by looking at this index. If the test dataset values and the indices of the trainer dataset are the same, it falls into that class. In the development for this project, the *sklearn.ensemble* library was used to use Random Forrest. After the preprocessing operations on the used dataset, the implementation of the model was carried out. In this implementation, evaluation metrics are calculated after the model is fitted.

Logistic regression is a statistical method used to analyze a dataset with one or more independent variables that determine an outcome. The result is measured with a binary variable. Like true or false, yes or no, male or female. The purpose of logistic regression is to find the most appropriate model to describe the relationship between a two-way characteristic and a set of related independent variables. Events are independent. The homogeneity of variance need not be satisfied. Errors need to be independent, but not normally distributed. It uses maximum probability estimation (MLE) instead of ordinary least squares (OLS) to estimate parameters and therefore relies on large sample approaches. It does not assume a linear relationship between the dependent variable and the independent variables, but assumes a linear relationship between the logits of the explanatory variables and the response. There should be no outliers in the data that could be evaluated by converting the continuous predictors to standard or z-scores and removing values below -3.29 or above 3.29. In logistic regression, the probability of realization of one of the values that the dependent variable can take is estimated and no preconditions are required for its implementation. The *sklearn.linear_model* was used to use logistic regression in the project. linear_model is a class of the sklearn module if contain different functions for performing machine learning with linear models. The term linear model implies that the model is specified as a linear combination of features.

Bayes' theorem shows the relationship between conditional probabilities and marginal probabilities within the probability distribution for a random variable.

P (A | B) = probability of event A occurring when event B occurs

P (A) = probability of event A occurring

P (B | A) = probability of event B occurring when event A occurs

P (B) = probability of event B occurring

Naive Bayes theorem is also based on Bayes theorem. It can work on unbalanced datasets. The way the algorithm works is it calculates the probability of each state for an element and classifies it according to the highest probability value. It can produce very successful works with a little training data. Like all other Bayesian classifiers, Naive Bayes assumes that each feature is not related to any other feature and works without any Bayesian model. In addition, since it is a typical method using a large number of samples, discretization method is used instead of probability estimation distribution. If a certain class or feature did not appear together in the training set, the probability estimate will be below 0. To avoid this, it is often necessary to change the estimate of the small sample. In the Naive Bayes classifier, according to the maximum

posterior probability (MAP) decision rule, correct classification can be obtained as long as the posterior probability of the correct class is higher than the other classes. Thus, it does not matter if the probability estimate is slightly or even severely uncertain, it will not affect the correct classification result. In this project, *sklearn.naive_bayes* is used to use Naive Bayes.

6. TECHNIQUES USED

Two techniques were used within the scope of the project. One of them is to separate the data set we have with *90% train and 10% test data*. In order to do this, train_test_split function is applied to the data set that has been processed before (mapping, drop, normalization, find outlier etc.). *train_test_split*(dataset_final, target, test_size=0.1, random_state=1) This function is pulled from the sklearn.model_selection library. The reason why the scikit-learn package, which is mostly used in development, is used so much is that it contains most of the basic methods we need, especially in subjects such as machine learning. Another reason is that there is no need for another package in data analytics applications, thanks to its separate modules such as filling in missing values in the data, cross-validation, and evaluation of results.

Another technique applied is *n-fold cross validation*. Cross-validation is a statistical resampling method used to evaluate the performance of the machine learning model on data it does not see, as objectively and accurately as possible. The reason for choosing cross validation is to optimize the developed model and increase the performance of the model. If we are to explain the N-fold cross validaiton over the data set used. The value of n is set to 10. While dividing this data, some deviations may occur according to the distribution of the data. To avoid this, n-fold cross validation divides it into equal parts in accordance with a determined n number and minimizes deviations and errors. Accuracy for each separated part is calculated one by one, then all the results found are summed and averaged, and the value we get gives the performance of our model.

7. TEST AND RESULTS

First of all, the libraries that will be used during development were imported. Then the csv file is read. There were some features that were previously reviewed and deleted for preliminary evaluation. In addition, the values that had little effect on the Attrition value but caused a decrease in the accuracy value in the performance measurements performed on our model were deleted. In order to calculate on the model, the Attrition attribute in the dataset was removed and Attrition was determined as the target attribute.

A report was created that enables analysis on the dataset. Used *sweetviz.analyze* to do this. Thanks to the graphics and values provided here, the distribution of categorical and numeric values has been seen. In this report, while some features are taken as categorical, it is seen that they are numeric in the dataset. That's why it was encoded. But a problem arose. In terms of Department, when we give the value of 1 to the Sales department and the value of 2 to the Research & Development department, the following situation arises; Research&Development>Sales. But this is not the right approach. For this reason, values that increase between them were encoded. ie BusinessTravel. After the encoding of this feature, no change was observed when the correlation was checked. The reason why Attrition was encoded is that when we processed our dataset with models, the evaluation metrics could not be measured. Because this feature has string values. After this procedure, it was observed that some values in the correlation table were negatively affected. However, in the subsequent examination, it was seen that there should be a negative relationship between these values and Attrition (like Age). Thus, the values were in a more accurate orientation.

Then, the missing value and duplicate values in the data set were examined with the *isnull(), nunique() and heatmap()* functions. Continuous and discrete values in categorical and numeric values were separated. Then, outliers in the dataset were found and it was tested whether better results are obtained when processing is performed for them, that is, when they are replaced with borders, or better results are obtained when no operation is performed on the outlier. It was observed that there was no big difference between them. For this reason, transactions were carried out on the outliers.

The *get_dummies()* function is used to get columns to represent our categorical data. Then, the categoric and numeric values are combined (with *train_test_split()*) to prepare the final dataset. Here, the size can also be determined by giving a parameter for the test data. After setting Taget to be 0 and 1, they are divided into x_test, y_test, x_train, y_train in the model. There is no Attrition attribute in X_train. Y_test and y_train contain the Attrition to be trained and tested.

With *SMOTE(),* it provides optimization on classification problems and unbalanced datasets. It provides this by increasing the yes value in Attrition in our dataset. This process is called oversampling. Accurancy value was giving lower results than expected because the number of 0 values was more than 1 values. Also, the metric values were 1.0 or 0.0 (recall, precision). *model_implement()* also provides the fit of the smoothed train and test data. Thus, the training of the model is ensured. Since the prediction of what will be tested with *model_predictions()* is requested, x_test i is sent. Since the Attrition values we want to predict from now on are y_test, y_test and model_predictions are used for the next evalution metrics. After the models were created, they were put into the testing phase for improvement studies. First of all, manual changes were made on the parameter values used. It has been observed that

the random_state value is important here. By making changes on this parameter, the change in the accuracy value was observed and continued with the value of 4.

For the n-fold cross validation, the n value was determined as 10 and the dataset was divided into 10 parts and started to be tested using the models. Changes have been made to the solver, penalty and max_iteration values for logistic regression. It was observed that the liblinear value for the Solver parameter gave better results. However, it was seen that the max_iteration value did not contribute.

Hyperparameter tuning was used for the random forest, and experiments were carried out by processing the model with all the values it can take for the random forest classifier. A total of 300 iteraions have been completed. As a result, the most appropriate values and the most appropriate results that can be obtained are returned. It has been observed that naive bayes achieves lower values than the other two algorithms. For this reason, it has been seen that the most suitable algorithms among the selected algorithms are Random Forest and Logistic Regression.

```
Accuracy:  0.8367346938775511
              precision    recall  f1-score   support

           0       0.85      0.97      0.91       120
           1       0.64      0.26      0.37        27

    accuracy                           0.84       147
   macro avg       0.74      0.61      0.64       147
weighted avg       0.81      0.84      0.81       147

Results of sklearn.metrics:
MAE: 0.16326530612244897
MSE: 0.16326530612244897
RMSE: 0.40406101782088427
R-Squared: -0.08888888888888902
```

Figure 14: Random Forest Classifier with %90 train, %10 test

```
dict_keys(['fit_time', 'score_time', 'test_acc', 'test_prec_macro', 'test_rec_micro',
'test_roc_auc', 'test_f1_macro', 'test_mae', 'test_mse', 'test_r2'])
[0.93273543 0.94618834 0.94170404 0.92825112 0.93273543 0.93721973
 0.92792793 0.92792793 0.93243243 0.94144144]
```

Figure 15: Random Forest Classifier with 10-Fold Cross Validation

```
Accuracy:  0.8095238095238095
              precision    recall  f1-score   support

           0       0.88      0.88      0.88       120
           1       0.48      0.48      0.48        27

    accuracy                           0.81       147
   macro avg       0.68      0.68      0.68       147
weighted avg       0.81      0.81      0.81       147

Results of sklearn.metrics:
MAE: 0.19047619047619047
MSE: 0.19047619047619047
RMSE: 0.4364357804719847
R-Squared: -0.2703703703703706
```

Figure 16: Logistic Regression with %90 train, %10 test

```
dict_keys(['fit_time', 'score_time', 'test_acc', 'test_prec_macro', 'test_rec_micro',
'test_roc_auc', 'test_f1_macro', 'test_mae', 'test_mse', 'test_r2'])
[0.93273543 0.94618834 0.92376682 0.9058296  0.93273543 0.93273543
 0.90990991 0.92342342 0.93693694 0.90540541]
```

Figure 17: Logistic Regression with 10-Fold Cross Validation

```
Accuracy:  0.6870748299319728
              precision    recall  f1-score   support

           0       0.90      0.69      0.78       120
           1       0.33      0.67      0.44        27

    accuracy                           0.69       147
   macro avg       0.61      0.68      0.61       147
weighted avg       0.80      0.69      0.72       147

Results of sklearn.metrics:
MAE: 0.3129251700680272
MSE: 0.3129251700680272
RMSE: 0.5593971487843206
R-Squared: -1.087037037037037
```

Figure 18: Naive Bayes with %90 train, %10 test

```
dict_keys(['fit_time', 'score_time', 'test_acc', 'test_prec_macro', 'test_rec_micro',
'test_roc_auc', 'test_f1_macro', 'test_mae', 'test_mse', 'test_r2'])
[0.79372197 0.73991031 0.76681614 0.75784753 0.79820628 0.82511211
 0.80630631 0.72972973 0.81081081 0.8018018 ]
```

Figure 19: Naive Bayes with 10-Fold Cross Validation

8. REFERENCES

https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-logistic-regression/

https://www.sciencedirect.com/topics/computer-science/logistic-regression

https://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/

https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

https://www.datasciencearth.com/algorithm-naive-bayes-classifier/

https://stackabuse.com/the-naive-bayes-algorithm-in-python-with-scikit-learn/

https://devhunteryz.wordpress.com/2018/09/20/rastgele-ormanrandom-forest-algoritmasi/

http://inseaddataanalytics.github.io/INSEADAnalytics/groupprojects/January2018FBL/IBM_Attrition_VSS.html#business_problem

https://www.knime.com/blog/predicting-employee-attrition-with-machine-learning

https://www.datacamp.com/community/tutorials/categorical-data

https://www.analyticsvidhya.com/blog/2021/03/zooming-out-a-look-at-outlier-and-how-to-deal-with-them-indata-science/

https://machinelearningmastery.com/machine-learning-in-python-step-by-step/

https://machinelearningmastery.com/data-preparation-techniques-for-machine-learning/

https://towardsdatascience.com/6-ways-to-improve-your-ml-model-accuracy-ec5c9599c436

https://erdincuzun.com/makine_ogrenmesi/naive-bayes-classifier/

https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

https://scikit-learn.org/stable/modules/cross_validation.html

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html?highlight=logistic%20regression#sklearn.linear_model.LogisticRegression

https://stackabuse.com/cross-validation-and-grid-search-for-model-selection-in-python/

https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/

https://medium.com/@ekrem.hatipoglu/machine-learning-classification-logistic-regression-part-8-b77d2a61aae1

https://medium.com/@afozbek_/sklearn-kütüphanesi-kullanarak-linear-regression-modeli-nasıl-geliştirilir-692a0bf13998

https://scikit-learn.org/stable/modules/model_evaluation.html

https://newbedev.com/evaluate-multiple-scores-on-sklearn-cross-val-score

https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://docs.w3cub.com/scikit_learn/modules/generated/sklearn.model_selection.cross_validate

https://towardsdatascience.com/6-ways-to-improve-your-ml-model-accuracy-ec5c9599c436

https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/

https://towardsdatascience.com/ways-to-evaluate-regression-models-77a3ff45ba70

https://www.datatechnotes.com/2019/10/accuracy-check-in-python-mae-mse-rmse-r.html

http://inseaddataanalytics.github.io/INSEADAnalytics/groupprojects/January2018FBL/IBM_Attrition_VSS.html