Artificial Intelligence

Section (2)

Assignment Title:
Technical Report.

Assignment Term:
Final Term.

Submitted By:
Osama Zamari.

Spring 2023

Table Of Content

Introduction:

The project's task is to solve the problem of employees' future by trying to predict whether the employee will leave or stay in the company sooner. After solving the problem, we will have many advantages and a clear plan for the future that will help the organization to determine who to keep and who to sack.

Impact On Organizations:

Feedback: When we know the employees whether they are leaving or not, we can understand why all that happens by finding correlations that help us to determine the problem and understand what bothers employees, in this case, we will know how to handle the employees. In addition, it will increase job satisfaction which leads to a greater reputation for the organization.

Better Management: by knowing the employees status (leaving or staying), we can set projects and hire specific employees for them. for example, if we want a specific employee to stay, we might give him an exciting project that might give the organization more time to look for alternative solutions to make him stay.

Wrong Decisions: We might have false predictions, then plan for a totally wrong plan according to our predictions. For example, if we determine who are the employees that will be sacked from the organization due to our predictions, we might take steps that affect them and the reputation of the organization.

Less Cost: instead of paying much money for someone that is more likely to leave the organization, it would be better to minimize the amount of money and hire someone else on a project that is likely to contribute on the project as same as the first employee.

Impact On Users:

Anger Of Being Controlled By Technology: sometimes it might lead to aggression according to the idea of being predicted by a technology machine that determines your future in the organization. That might lead to dissatisfaction in the organization. In addition, it might be the emotions or personal issues for the employee that might lead to a false prediction.

Future Awareness: By knowing their future whether they are staying in the organization or leaving, they can take it as a challenge to improve themselves. Furthermore, if they took the opportunity to stay in the organization, they will focus more on the knowledge to earn it.

It would be so much greater to know the future of employees by taking into consideration the emotions not only the data that would determine his/her future. The project aims to optimize the plans and strategies that an organization is making, by giving them accurate predictions that help them to know what they have to do with these employees and how they are going to keep them and what are the necessary changes that they need to do about them.

Materials:

https://www.kaggle.com/datasets/tejashvi14/employee-future-prediction

The source, the collection methodology and how many projects the dataset used in are private and not provided in Kaggle.

To solve the problem, we have the dataset that shows the education level of the employee, his/her joining year in the company, city where the office is located, the payment tier for the employee, his/her age, the gender of the employee, ever benched which means whether an employee kept out of projects for 1 month or more, years of experience in current field and the target which shows the employee left or stayed in the company. With this data, we will use machine learning to predict whether an employee will leave the company for the next two years approximately or not.

For this dataset, the data is dummy, which means it is a generated copy of the original dataset or it might be generated manually and kept private.

Attributes In Details:

Education Level: consists of three levels; Bachelors, Masters and PHD, which show the education that an employee has.

JoiningYear: a range between 2012 to 2018 shows what year an employee has joined the company.

City: consists of three cities; Bangalore, Pune, and New Delhi, which show where the post office is located.

PaymentTier: consists of three tiers, tier one is the highest pay, tier two is the medium pay, tier three is the lowest pay.

Age: A range between 22 to 41 which shows the age of the employee.

Gender: Shows the employee gender whether he is a male or she is a female.

EverBenched: A true or false that shows whether an employee has kept out from projects for one month or more.

ExperienceInCurrent: A range between 0 to 7 which shows how many years the employee has experience in a field.

LeaveOrNot: A true of false which shows whether the employee left the company (True) or stayed in the company (False).

Dataset Size:

The dataset contains 8 columns and the 9th is the target column which is LeaveOrNot. And contains 4653 rows. Further information, it contains 100 null values distributed between the 8 columns without the target. In addition, the size of the target value was imbalanced, Staying employees 3053, and leaving employees 1600.

Projects Used The Dataset:

The dataset has been posted in public so data scientist could try this dataset to analyze it and make predictions, one of the projects was great and done by Sonali Singh that used decision tree classifier and got an accuracy score equals to 77%, then she used Ada Booster to improve weak learners such as the decision tree that she used to improve the accuracy to 78%. Another project was sone by Manthan Shettigar, he used multiple models but the best one he got was decision tree with an accuracy equals to 83.5% and a precision equals to 93%.

In general, the dataset is used for personal training, there were 52 codes that have been shared to public and used this for their own benefit, and the dataset got almost 7755 downloads for the dataset.

Pre-Processing Used:

Preprocessing the data is obligatory to start implementing machine learning, which includes eliminating null values and changing the categorical to numerical values so the machine can understand because it cannot understand letters. In addition to methodologies to prepare the data for example normalization and balancing the target data.

I pre proceeded the data to start fitting it in the models, I did some techniques to prepare the data to be trained and predict:

Education: I filled the missing values which counted (13) with unknown because I did not want to change the whole education for a record, so the best way is to fill it with unknown value. Then. Then, I used label encoder to turn them to numerical values.

Joining Year: I filled the missing values which counted (7) with 2017, because the data shows that a big percentage of employees joined the company in 2017. Then, I changed the whole column from float type to integer type because we don't have month and day, we only have the year which could be represented as integer number.

City: I filled the missing values which counted (22) with conditions, first, I filled some of the null values with Bangalore if the educational level of that row was Bachelor because from analyzing, I saw that the Bangalore office is the biggest office and has the most employees and most of them are bachelors. Then, I filled some of null values with New Delhi if the educational level was Masters or PHD, because from analyzing, it shows that New Delhi takes a lot of high degrees employees and especially a lot of them are high educational level in New Delhi, then, I filled the rest null values with Unknown because it is not known what they would be. Then I used label encoder to change the categorical to numerical.

Payment Tier: I filled the missing values which counted (12) with conditions, first I filled the missing values with tier 1 if the city was Bangalore because the city has the most tier 1 payment, and number 2 if the city was Pune because it has more the most tier 2 payment, and number 3 if the city was New Delhi because it has better percentage between the number of employees and the payment tier which shows that the best fit is tier 3 in New Delhi

Age: I filled the missing values which counted (9) with the mean of the Age column, then I changed the data type from float to integers because the mean of the column's ages would give a float value. Then I did data discretization by putting range from 22 to 28 called 'Young', 29 to 35 called Mature and 36 and above called Old, then, I used label encoder to turn the words into numbers.

Gender: I filled the missing values which counted (13) with unknown because after analyzing, I saw that the percentage between men and women are nearly equal, men were 1117, women were 978. Then I used label encoder.

Ever Benched: I filled the missing values which counted (9) with conditions, if the educational level was bachelor, fill the row with the value yes, and if it was masters or PHD then fill it with the value No, because after analyzing, it shows that there are very few numbers of masters and PHDs employees that has been benched, and for the bachelor, there was some of benched employees.

Experience In Current: I filled the missing values which counted (15) with the median which gave the value (3), and then, I changed the float type to integer to remove the float writing type and be more specific.

Leave Or Not: There were no missing values in the column. However, the 0 was 1249, and the 1 was 859, it can be considered as balanced data, but I used oversampling to balance the target values by increasing the minority, and modeled both, with over sampling and without.

Balancing technique, I used over sampling to balance the target data to be approximately balanced.

Normalization technique, I used standard scaler to put the data in the same scale.

I defined all features except Leave Or Not (because it is the target) as x (Victor Features), and I defined Leave Or Not (The target) as y (Target Feature).

Since the data was classified such as education (it contains only 3 values) I used label encoder to change the categorical values into numerical values.

Exploration Used:

To explore the dataset, I used some built in functions that python provides:

Df.head(): I used this function to see the first 5 rows from the dataset to have a look on it.

Df.columns: I used this function to see what the columns are.

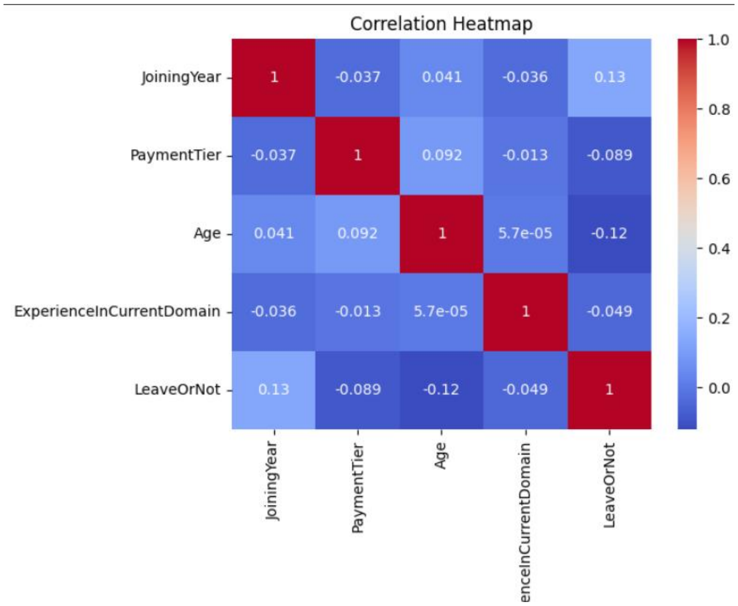
Df.describe(): I used this function to see some statistics of the data such as the count, min, and max.

Df.info(): I used this function to see the data type of the columns and the number of rows and columns.

Df.isnull.sum(): I used this function to see the total null values in each column.

Df.duplicated().sum(): I used this function to check whether there is duplicated data or not, and it shows that it contains 1851, so, I deleted it which lead the dataset size to 2108 rows.

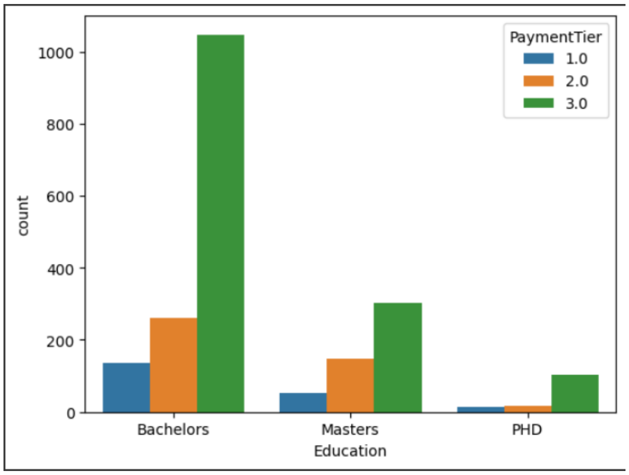
Visualizing:

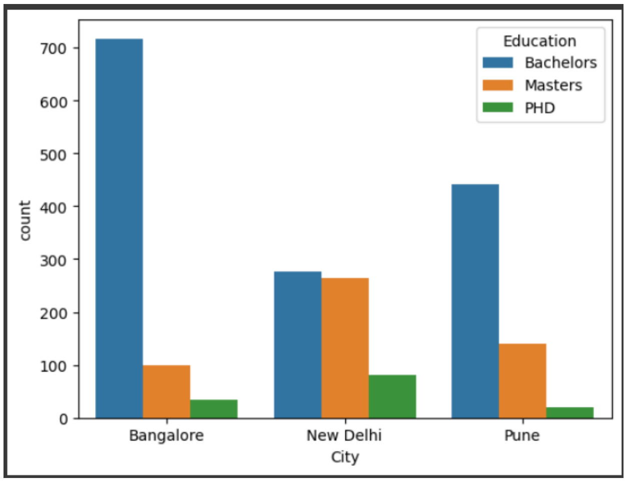


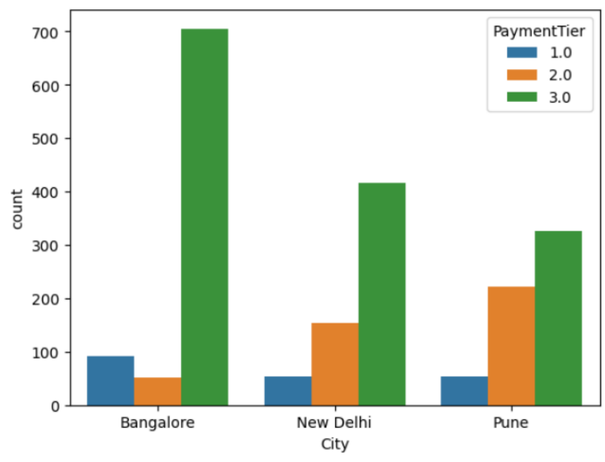
Heatmap: it is a graph that shows the relation between each column and how they are related to each other by coloring, the deeper the color is the more relationship between each other they have. It shows that there are no big relationships between the features.

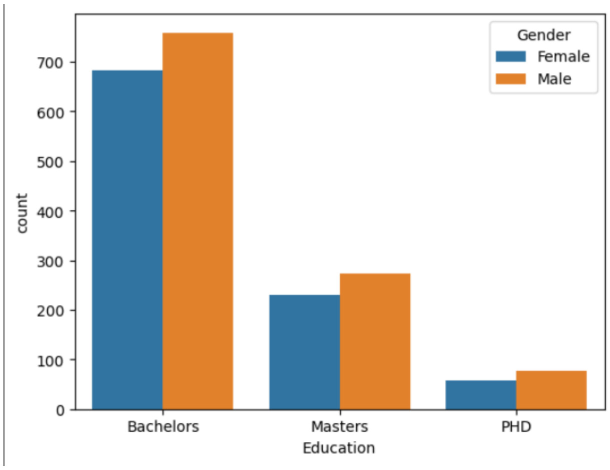
Pie plot: it is a circle graph that looks like a pie which is used for showing how much of each value.

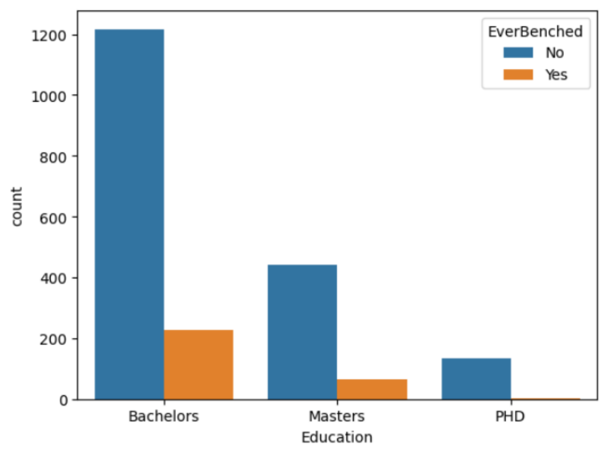
Bar plot: a rectangular bar that expresses the value of specific object which is good for comparing and can help people to understand how much the difference between these values is.

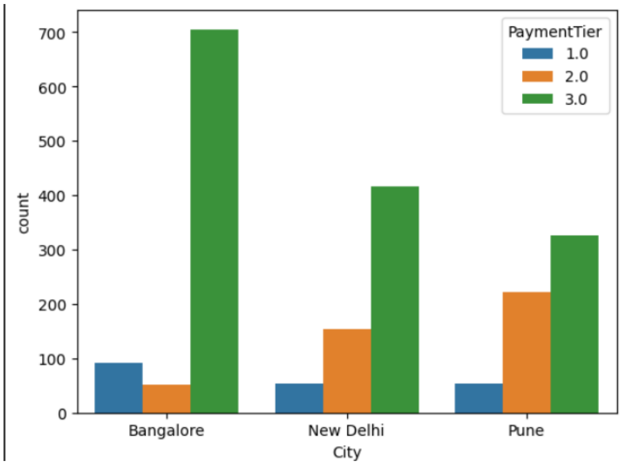


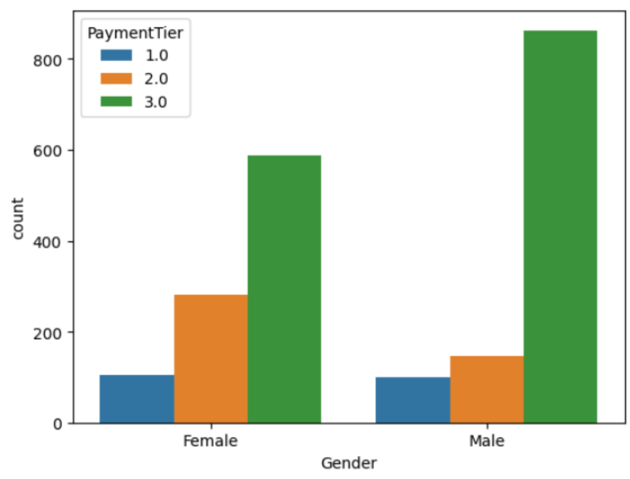


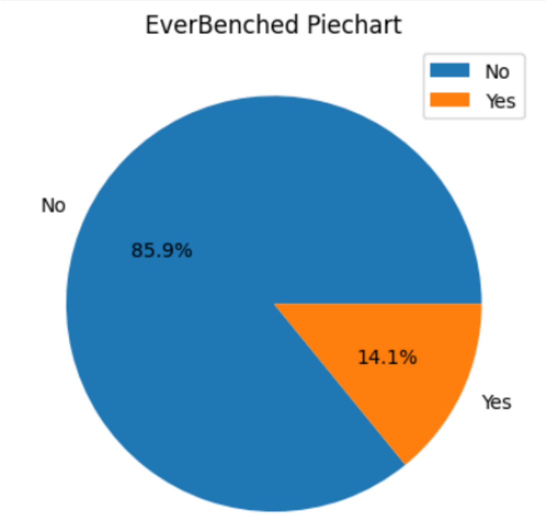

PaymentTier
1.0
2.0
3.0
count
0
100
200
300
400
500
600
700
Bangalore
New Delhi
Pune
City

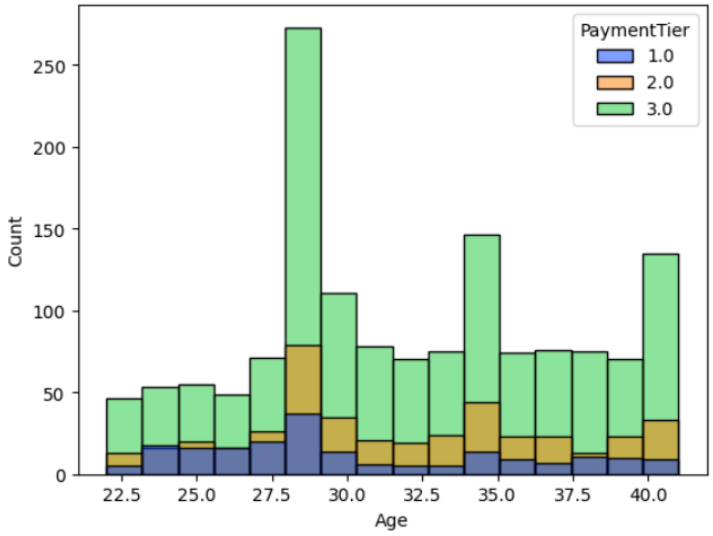

Gender
Female
Male
count
0
100
200
300
400
500
600
700
Bachelors
Masters
PHD
Education

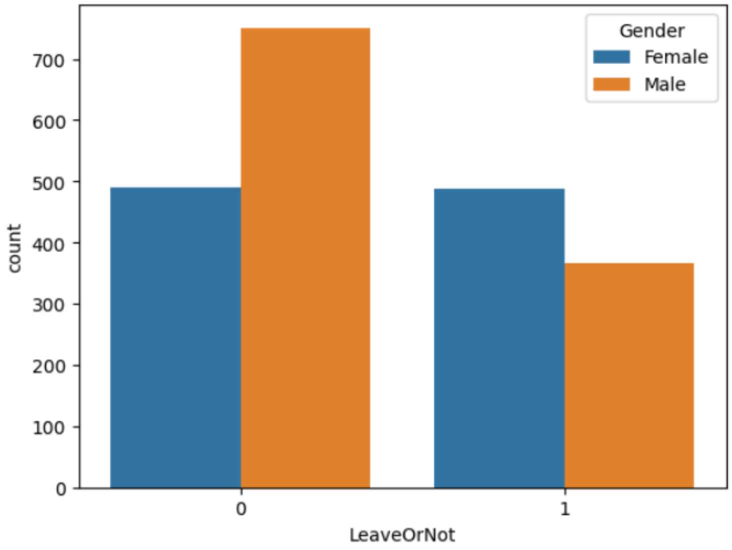

EverBenched
No
Yes
count
0
200
400
600
800
1000
1200
Bachelors
Masters
PHD
Education


PaymentTier
1.0
2.0
3.0
count
0
100
200
300
400
500
600
700
Bangalore
New Delhi
Pune
City


PaymentTier
1.0
2.0
3.0
count
0
200
400
600
800
Female
Male
Gender


EverBenched Piechart
No
Yes
No
85.9%
14.1%
Yes





Some of the information that I got due to the visualization and analyzing is:

Bangalore is the biggest office, then Pune, then New Delhi.

New Delhi has the highest number of Masters and PHDs employees.

New Delhi loves to take high educational employees.

Educational level is not a determiner for the payment tier.

In all offices, the correlation between the educational level and the payment tier is not huge.

The best place that gives more salary is Pune.

Most of masters and PhDs employees have not benched.

85.9% of employees were not benched from projects.

Bangalore is the biggest office and most of them from tier 3 payment.

Nearly in all educational levels, we have equal gender percentage.

Women percentage are more paid than men.

Women percentage are more likely to leave the company.

Models Used:

In the project, I have used two models which are:

Random Forest Classifier:

A machine learning model that is made of multiple decision trees that leads to a final result based on questions that have been answered previously. It's work by making multiple decision trees which they make prediction on a subset of a feature, then, the random forest algorithm predicts something according to what the other trees predicting, in another meaningful way, the random forest algorithm will choose what the majority has predicted.

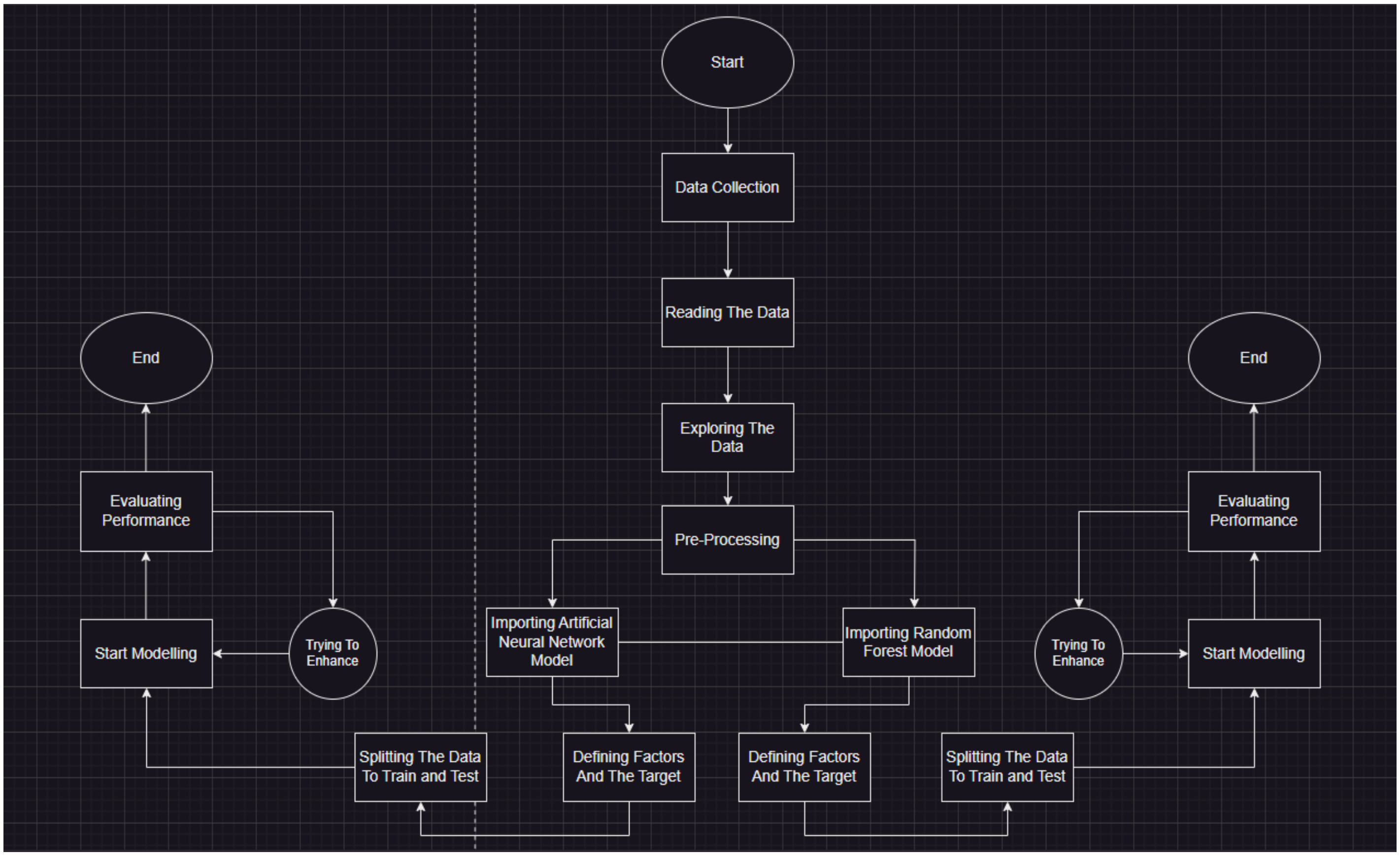
I used random forest model because Its advantage is correcting overfitting decision trees in training stage. In addition, instead of implementing decision tree model, I used the enhanced version of it, which is random forest. Furthermore, I used decision tree model, but the performance was poor compared to random forest.

Artificial Neural Network:

A deep learning model is made based on human's brain functions to solve complex problems. It is created by neurons that can be considered as a block of information to perform computations and produces the output and 3 core layers which called input layer where the data is fed inside the network, hidden layers which can be more than one which help the network to learn more complex relationships in the data and the output layer which provide the last result. In addition, Weights and activation functions which allows the network to learn some patterns in the data. Then, the ANN model train to give the prediction, and every time it adjusts the weights based on the error to increase the accuracy.

I used ANN because it handles the nonlinear relations between features properly. In addition to the model describe itself that ANN is a deep learning models to solve complex problems and give good accuracy.

Pipeline For Both Models:


Start
Data Collection
Reading The Data
Exploring The Data
Pre-Processing
Importing Artificial Neural Network Model
Importing Random Forest Model
Defining Factors And The Target
Defining Factors And The Target
Splitting The Data To Train and Test
Splitting The Data To Train and Test
Start Modelling
Start Modelling
Trying To Enhance
Trying To Enhance
Evaluating Performance
Evaluating Performance
End
End

Technical Implementation:

For Both Models:

First, I uploaded looked up for the dataset in Kaggle which provides a lot of datasets to use on, after finding the dataset which is predicting the future of the employees whether they are staying or leaving, I opened google colab and imported the dataset locally so I can use it, then, I read the data using pandas library, after that, I explored the dataset with (info, describe, duplicated) functions, info function is to see the data type of each column and the number of rows and how many values each column has, describe function to see some statistics measures for each numerical column and duplicated function to check the duplicated rows, after I saw that there are duplicated rows, I used drop_duplicates(keep = False) to delete them.

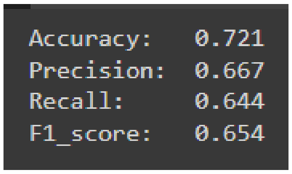
Then, I analyzed the features by using matplotlib and seaborn libraries to help me visualize it to make it easier to understand it.

Secondly, I preprocessed the columns to prepare it for the model, starting with 'Education' column, where I filled the null values with 'Unknown' value so it does not affect the pattern, then I used label encoder to change the words to numbers. Then, for the Joining Year, I filled the null values with 2017 because it is the most joining year for the company, then changed the datatype of the column to integer just for more organizing and logical data. After that, for City column, I filled the null values with the conditions as mentioned in the preprocessing section. Then, for the payment tier column, I filled the null values with conditions as mentioned in the preprocessing section, then changed the data type to integer just for more organizing and logical data. Then for the Age column, I filled the null values with the mean instead of a specific age, then I switched the column data type to integer to remove the float values that caused from filling null values with the mean, and for organizing and logical data. In addition, I ranged the values from 22 to 28 shall be called Young, 29 to 35 shall be called Mature, above, and equal to 36 shall be called Old, then used label encoding. Then for the Gender column, I filled the null values with Unknown then I used label encoding. For Ever Benched, I filled the null values with conditions as mentioned in the preprocessing section. Then for experience in current domain column, I filled the null values with the median, then changed the data type of the column to integer just for organizing and logical data.

Lastly, I did imbalance technique to make the target data more balanced, so , I installed imbalance learn to implement it in python google colab.

Random Forest:

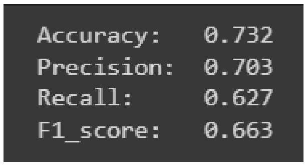
First, I defined what x is which is every column expect the target "LeaveOrNot", and defined y which is LeaveOrNot column only. Then, I imported the train test split library to split the data into training set and testing set and put the size of the testing set to 20 percent of the data. Then, I imported random forest model and defined an object of the model with 300 trees. After that, I imported the random over sampler from imblearn library to balance the data, I used over sampling technique with minority strategy, then defined new x and y train called x over and y over and equalize it with oversample fitting with x train and y train. Then I imported the standard scaler method which is a normalizing technique to put the data into a same scale. Then I imported the accuracy, recall, precision and f1 score functions from metrics library, then I defined an empty list for each metrics score. Then starting a for loop (size 50), then fitted the x over and y over to the object of random forest model, then defined (PredictU) to predict the x test, then, I defined a variable called Acc and equalize it with the accuracy score function of y test and what our model predicted (PredictU), then appended the result to the accuracy list, same for recall and precision and f1 score. After that, I defined new variables for each metrices score and did an operation which takes the sum of the metrices score and divided by its number, for example, take the sum of the accuracy list and divided by the number of elements it contains. Then print out the four metrices score with round function equals to 3 to print only 3 numbers after the dot (printing the variable that holds the sum of the list divided by its number. The measures were:



```
Accuracy:   0.721
Precision:  0.667
Recall:     0.644
F1_score:   0.654
```

Artificial Neural Network:

First, I defined what x is which is every column expect the target "LeaveOrNot", and defined y which is LeaveOrNot column only. Then, I imported the train test split library to split the data into training set and testing set and put the size of the testing set to 20 percent of the data. Then, I used under sampling technique with majority strategy, then defined new x and y train called x over and y over and equalize it with under sample fitting with x train and y train. Then I imported the standard scaler method which is a normalizing technique to put the data into a same scale. Then, I imported Sequential function from keras.model library, then I defined an object that equalizes with Sequential function. Then, imported Dense from keras.layers library which allows me to add the required layers, first I added the input layer with 50 units and relu activation function, then two hidden layers, the first has 70 units and the second has 85 units, and the output layer has 1 unit and Sigmoid activation function. After that, I used function compile to add an optimizer ‘adam’, and an evaluation for the loss ‘binary_crossentropy’ , and the accuracy metrics. Finally, I initialized y_pred to take the prediction of x test, then put a condition for it to be more than 0.5 consider it as true and less than 0.5 will be considered false, then printing the measures score of it.



```
Accuracy:   0.732
Precision:  0.703
Recall:     0.627
F1_score:   0.663
```

Project’s Benefits For Organization:

As mentioned before, the project aims to predict whether the employee is going to leave the organization, the project can assist so many sides of the organization, by exploring the reasons and try to analyze it, which let us know what the causes are to let an employee leaves or stays. In management section as I imagine, the project would let us know how to handle the predicted leavers of employees by using their services in projects that their experiences are a must to implement the project, or how make meetings with them to understand and predict why would they leave the organization and whether they have intentions to leave or not to understand what they would need to keep them. In addition, in financial section, we would make raises on the most loyal to the organization and handle how the money should be spent on employees. In addition, the project can be integrated into another model in the organization, the other model should tell the organization whether they should give the employees raises and promotions since they are staying to extend their time in the organization and increase the loyalty for it. and not giving raises and promotions for employees that are more likely to leave the organization sooner.

Evaluation Measures:

For both models, I used accuracy score to get the percentage of the correct predictions compared to the total number of results, and it is used for classification learning. I used precision score to get the percentage of the correct positive predictions compared to all positive results, which shows us how the model’s performance avoids false positives. I used recall score to get the percentage of the correct predicted positive compared to all actual positive results, which shows us how the model’s performance can get all the positive cases. I used f1 score to measure how the model performs by combining precision results and recall results to a single score which shows the positive results and avoid false positive results and false negative. In addition to all of this, I used round function with the parameter of 3 to get only 3 elements to get a simpler number.

For Artificial Neural Network only, I used binary crossentropy to check how much the model make losses of the data, by giving the dissimilarity between the predicted results and the true binary. I used the accuracy of the training of the model which shows how accurate the model on training with the data.

Model Performance Enhancement:

For random forest, I tried to remove some features from x to see whether there is a feature that affects the performance of the model, but the best was to keep all features. Then , I tried to change the number of trees, I tried from 100 to 500, and it shows that the best score is when I set the number of trees to 300. After that I tried to remove the standard scaler, but it was not ideal due to no big change in the performance but it is better to put the data into the same scaler. Then I tried to change the testing size to .33 and to .42, but it shows that the best is to set the data testing size to 20 percent only of the data. After that I tried to change the balancing technique to under sampling, but it affected the performance of the model so I kept it on the over sampling technique. However, I tried to train the data without using sampling technique, it is nearly the same percentage between both with sampling technique and without it. Lastly, I did 50 iterations to train the data on different splitting.

For Artificial Neural Network, I tried to remove features of x to check whether it does affect the performance of the model, but it showed that when you take all the features the performance is better. Then I tried to change the size of the test but it showed that 20 percent is the best. Then I tried to change the sampling technique to under sampling, but it showed that when I use over sampling it is better for the performance of the model. After that, I set the input layer units to 50, in the first, I put the input units to 8, but the performance was not that good, then changed it to 12 and it was getting better slightly, after multiple experiments I found that 25 is a good number but it loses about 28 percent of the data, then, I set the number of units to 50, and it was the best choice with almost 26 percent of data loss, and an accuracy of training to 86 percent. I added more hidden layers with different units, and it showed that the first hidden layer set to 70 is the best and the second hidden layer set to 85 units. After that I changed the number of sampling technique, I used the under sampling technique and I saw that the performance of the model has been increased, so I changed from over sampling to under sampling, over sampling gave f1 score equals to 64, but in under sampling it gave f1 score equals to 66. Then I tried ANN without balancing technique, and it shows that the performance was nearly the same.

Analysis Of The Results:

Models With Sampling Technique:

As I mentioned before, I used Random Forest and ANN, first, let’s analyze the random forest first:

With Using Sampling Techniques:

Measure	Random Forest	Artificial Neural Network
Accuracy	0.718	0.725
Precision	0.662	0.685
Recall	0.641	0.631

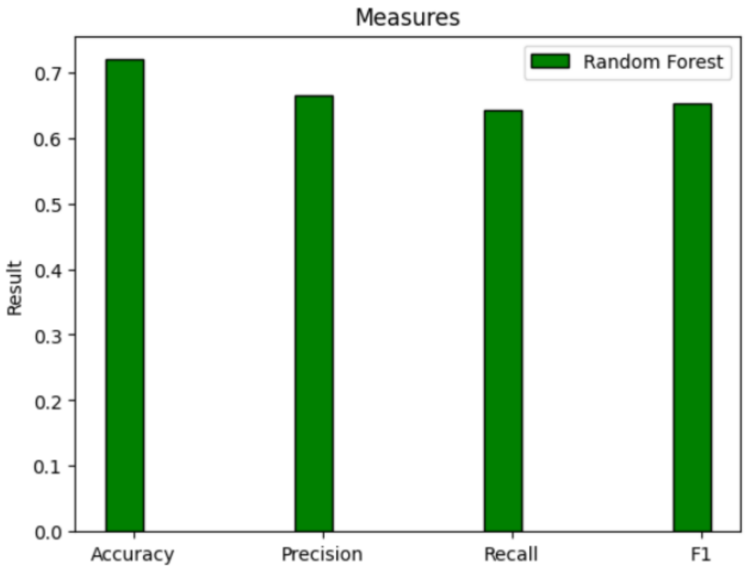
F-1	0.651	0.657

We can see that with using sampling techniques the accuracy of ANN is slightly higher than random forest at predicting correctly . And for precision, ANN is higher than random forest at predicting the employees who left. And for recall, Random forest is higher than ANN at identifying the employees who actually left the company. And for f1 score, ANN is nearly equal with random forest at predicting positive and negative cases.

Without Using Sampling Techniques:

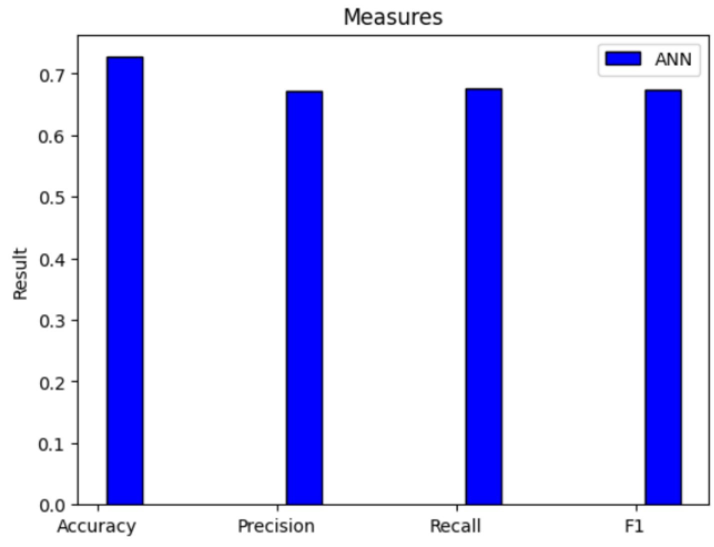
Measure	Random Forest	Artificial Neural Network
Accuracy	0.732	0.739
Precision	0.7	0.723
Recall	0.612	0.608
F-1	0.652	0.66

We can see that without using sampling techniques the accuracy of ANN is slightly higher than random forest at predicting correctly . And for precision, ANN is higher than random forest at predicting the employees who left. And for recall, Random forest is higher than ANN at identifying the employees who actually left the company. And for f1 score, ANN is higher than random forest at predicting positive and negative cases.

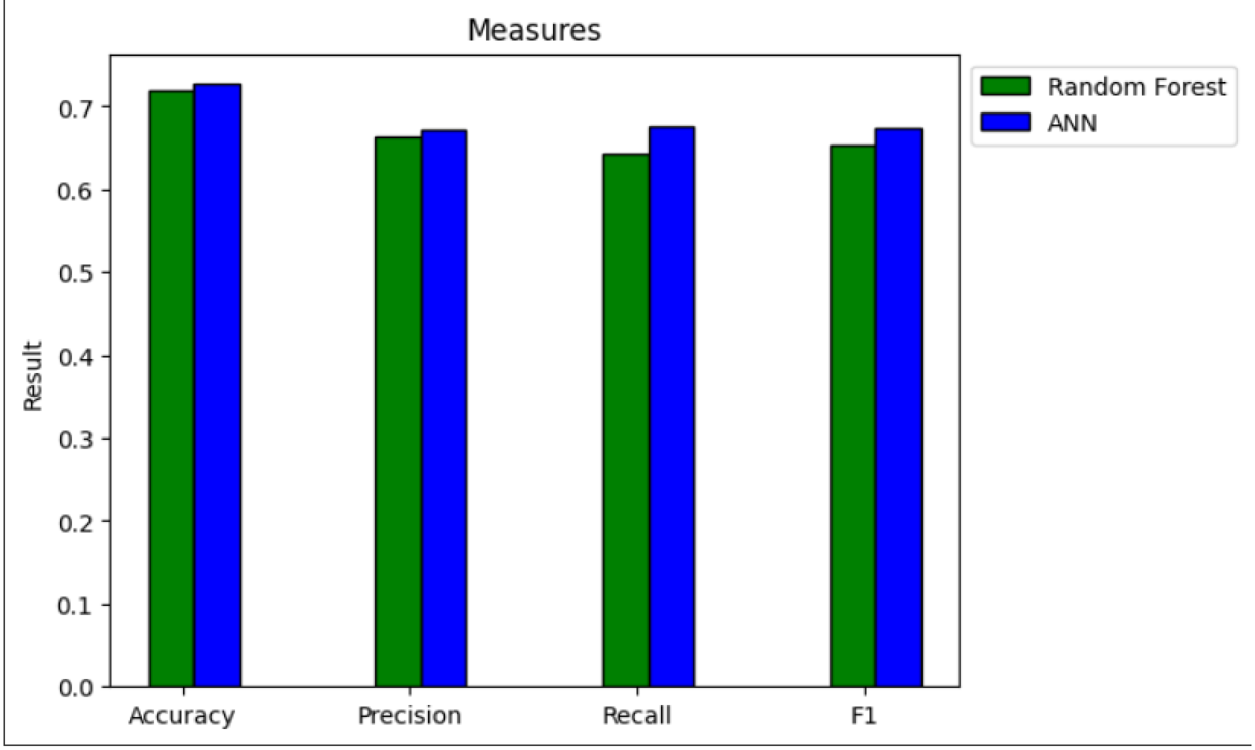


As we can see, the measures are higher than 60% and the maximum is the accuracy which is nearly equals to 72%, which shows that the model is not that very accurate due to multiple reasons such as wrong preprocessing, even though what I have preprocessed is close to what the data is, but nothing is guaranteed. Lastly, the number of data is not enough to train on, the rows were less than 2500 rows which did not give the chance for the model to train on multiple and complex patterns.

Now for the ANN:



As it shown in the figure, it is nearly the same as random forest but with slightly better numbers, but the difference is that ANN model requires huge amount of data to practice on because it is a model of deep learning which gives more accuracy and better performance exchange of the huge amount of data that are clear to practice on it.



This is the difference between both models in one chart, it proves that ANN better performance, even though the number of rows were few comparing of what an ANN model requires to.

Keep in mind that all the experiments I did was not as good as the last one which I'm talking about, because I tried everything to change the weakness of the two models such as changing the sampling technique, vectors, changing preprocessing methods, increasing, or decreasing the number of trees in random forest and iterations, increasing, or decreasing the number of epochs, batch size, number of layers and units, and the final result is as it shown in the figure.

Implement The Project In Jordan:

Based on the results of the two models, I can say that for this time, the project is not capable to do good predictions due to lack of the data, but since the models are doing a good job comparing to the amount of data, I can say that project will be implemented in Jordan after couple of years, but it requires to collect more data and as much as of new patterns so it gives the accurate decisions. The benefits of this project in Jordan are not different from what I have mentioned before, we will be able to give decision about management of the employees and financial department. In addition, we will be able to unleash the potential of employees when they are in the right position and the right projects that they would like to be in. For example, if we knew that an employee would leave and the organization does not intend to keep him, the organization can give him a good project that he would love to work hard on which increases the reputation of the organization that they care and want the best for their employees.

Further Enhancement:

For the future, even when the models are more trained and more accurate, there are many unpredicted features that are very strong to completely change the status from leaving or staying to staying or leaving such as the emotions, many employees are very satisfied in the job but the model might give that he is leaving the organization, because the features are about the general information about the employee, but not very deep into the satisfaction of the employee, that's why we need to take into consideration other probabilities in each time we want to make a decision, we can improve the project by adding more information such as job satisfaction which gives us a more obvious about the employees and strengthen the accuracy of the model to understand what our employees feel. In addition to all of that, we can analyze the results and try to fix it much easier by making more features such as job satisfaction such as payment satisfaction or projects satisfaction if they enjoyed the projects or not.

My Role:

As a data scientist that intends to improve the project to beneficial the organization, I would like to keep working on the models and receive more data and adding new features to make our project more reliable so we can trust the machines to work on the hidden patterns and predict the impossible to make up with the technology and reach to the modern world. In addition to increase our organization profits.

References

Analytics Vidhya. (2021). *Introduction to Artificial Neural Networks*.

IBM (2023). *What is Random Forest? | IBM*. [online] www.ibm.com.