Machine Learning (CS3102) Project – Multi Step Regression-Classification-Attrition

The abstract of this project :-

Most HR analytics projects usually focus only on predicting whether an employee will leave the company or not, and not on how it can impact the company in terms of **profit and losses**.
In this project, we solve the issues faced by HR analytics regarding the future salary of employees likely to stay and the potential losses faced by the company when certain employees decide to leave, which is a real-world scenario that HR departments around the world strive to quantify.

The methods used to solve this problem are as follows:

1. For predicting whether an employee will leave or not, we use **Logistic Regression**, **Decision Tree** and **SVM** to find the best fitting model in order to lower the chances of overfitting / underfitting.
To check which model works the best, we make use of metrics like *F1-score* and *AUC-ROC*.

2. We then simulate future salaries for each employee using a **performance-based growth system**. This is because it rewards certain employees for working with more efficiency instead of incrementing everyone's salary with one metric despite the hard work some employees put in compared to the others.

3. Predict future salaries but this time for employees likely to say. For this we use regression models like **Random Forest Regressor**, **Ridge, Lasso** and **Support Vector Regression** (SVR).

4. We then identify employees that are likely to stay by defining a certain threshold and simulate their future salaries.

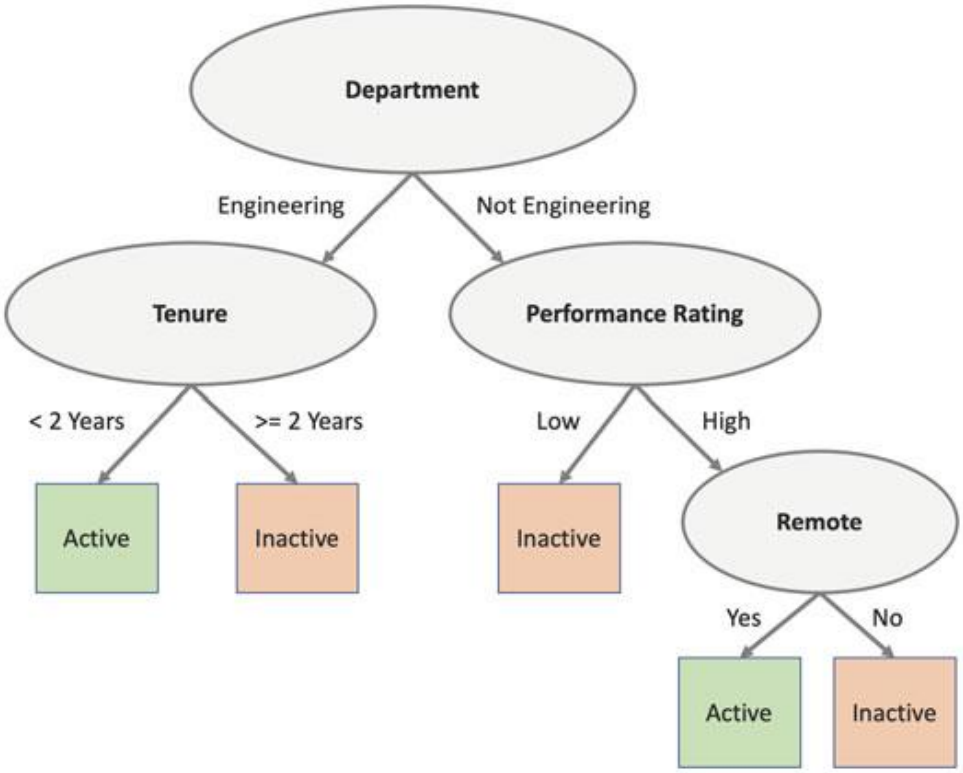
5. For the employees unlikely to stay, we simply estimate the expected salary loss for each employee and take the aggregate across all employees. This helps a company answer how much value they would be losing if certain employees decided to leave.

Introduction :-

In most companies, the HR analytics department has a fair share of problems to solve regarding the future of the company. One of them is to estimate / predict how much losses they would incur if some employees decided to leave the company on their own accord.

Usually, these companies focus only on predicting whether employees are likely to leave the company or not, and don't take into considering the **potential financial losses** they can face.

This project aims to solve that problem by taking attrition, classification and regression, all three at once to model not only who might leave, but also the **financial impact** of their departure.

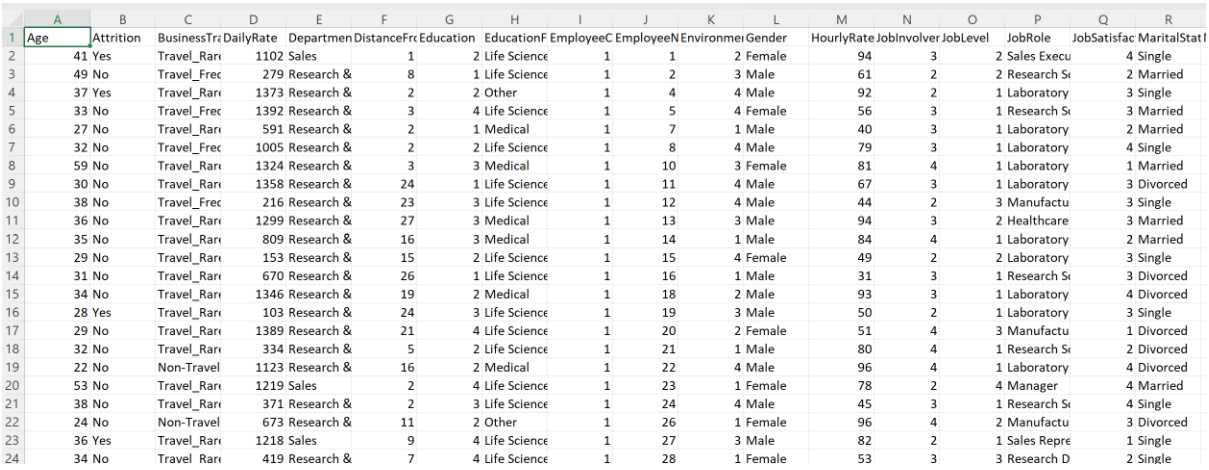


The methodology behind this project :-

To start off, we made use of three models – Logistic Regression, Decision Tree and SVM, to predict whether an employee would leave or not. In the predictions, a '1' meant that the employee would leave, and a '0' meant otherwise.

We did this using the sci-kit learn library and more importantly the pandas library to load our IBM HR Analytics Employee Attrition dataset from Kaggle, in the form of a '.csv' file.

Brief look at the dataset we'll be working with



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Age	Attrition	BusinessTra	DailyRate	Departmen	DistanceFr	Education	EducationF	EmployeeC	EmployeeN	Environmer	Gender	HourlyRate	JobInvolver	JobLevel	JobRole	JobSatisfac	MaritalStat
2	41	Yes	Travel_Rare	1102	Sales	1	2	Life Science	1	1	2	Female	94	3	2	Sales Execu	4	Single
3	49	No	Travel_Freq	279	Research &	8	1	Life Science	1	2	3	Male	61	2	2	Research Sc	2	Married
4	37	Yes	Travel_Rare	1373	Research &	2	2	Other	1	4	4	Male	92	2	1	Laboratory	3	Single
5	33	No	Travel_Freq	1392	Research &	3	4	Life Science	1	5	4	Female	56	3	1	Research Sc	3	Married
6	27	No	Travel_Rare	591	Research &	2	1	Medical	1	7	1	Male	40	3	1	Laboratory	2	Married
7	32	No	Travel_Freq	1005	Research &	2	2	Life Science	1	8	4	Male	79	3	1	Laboratory	4	Single
8	59	No	Travel_Rare	1324	Research &	3	3	Medical	1	10	3	Female	81	4	1	Laboratory	1	Married
9	30	No	Travel_Rare	1358	Research &	24	1	Life Science	1	11	4	Male	67	3	1	Laboratory	3	Divorced
10	38	No	Travel_Freq	216	Research &	23	3	Life Science	1	12	4	Male	44	2	3	Manufactu	3	Single
11	36	No	Travel_Rare	1299	Research &	27	3	Medical	1	13	3	Male	94	3	2	Healthcare	3	Married
12	35	No	Travel_Rare	809	Research &	16	3	Medical	1	14	1	Male	84	4	1	Laboratory	2	Married
13	29	No	Travel_Rare	153	Research &	15	2	Life Science	1	15	4	Female	49	2	2	Laboratory	3	Single
14	31	No	Travel_Rare	670	Research &	26	1	Life Science	1	16	1	Male	31	3	1	Research Sc	3	Divorced
15	34	No	Travel_Rare	1346	Research &	19	2	Medical	1	18	2	Male	93	3	1	Laboratory	4	Divorced
16	28	Yes	Travel_Rare	103	Research &	24	3	Life Science	1	19	3	Male	50	2	1	Laboratory	3	Single
17	29	No	Travel_Rare	1389	Research &	21	4	Life Science	1	20	2	Female	51	4	3	Manufactu	1	Divorced
18	32	No	Travel_Rare	334	Research &	5	2	Life Science	1	21	1	Male	80	4	1	Research Sc	2	Divorced
19	22	No	Non-Travel	1123	Research &	16	2	Medical	1	22	4	Male	96	4	1	Laboratory	4	Divorced
20	53	No	Travel_Rare	1219	Sales	2	4	Life Science	1	23	1	Female	78	2	4	Manager	4	Married
21	38	No	Travel_Rare	371	Research &	2	3	Life Science	1	24	4	Male	45	3	1	Research Sc	4	Single
22	24	No	Non-Travel	673	Research &	11	2	Other	1	26	1	Female	96	4	2	Manufactu	3	Divorced
23	36	Yes	Travel_Rare	1218	Sales	9	4	Life Science	1	27	3	Male	82	2	1	Sales Repre	1	Single
24	34	No	Travel_Rare	419	Research &	7	4	Life Science	1	28	1	Female	53	3	3	Research D	2	Single

From the sci-kit learn library we imported train_test_split for model selection, LabelEncoder and StandardScaler for pre-processing of the data, our model (Logistic Regression / Decision Tree / SVM), and metrics like f1_score, roc_curve and auc. To add on, we make use of the matplotlib.pyplot library to draw the AUC-ROC curve for our better understanding.

Pre-processing is a crucial step for this project as real-world data is

usually not clean, consistent, or immediately usable by models. In this case we encode the variable “Attrition” (Yes/No) as binary (1/0), drop irrelevant columns like *EmployeeNumber, EmployeeCount, Over18* and *StandardHours*, and finish the pre-processing by encoding all the remaining categorical columns numerically using the **LabelEncoder** function.

Since we are also predicting the future salary of the employees, we add an increment column based on **performance**. 10% raise for rating 4 and 5% otherwise. We then compute the future salary by applying the raise to the **MonthlyIncome** column in our dataset. This simulates the income of each employee after they get a raise.

Further, we predict simulated future salary for employees likely to stay by using four models to ensure the best one. The models we use in this part are Random Forest Regressor, Ridge and Lasso regression, and SVR (Support Vector Regression). You might ask why we’re doing this when we already have the model to predict future salary. The reason is, in the previous part we used a fixed formula to generate a **ground truth**. In this part we train a regression model to learn patterns from those simulated values. This allows us to:

- Avoid using a hardcoded rule for salary prediction;
- Predict future salary flexibly based on features like performance, tenure, job level;
- Generalize to new employees without needing to reapply the simulation formula manually.

This enables downstream steps like estimating potential financial loss when an employee is at risk of leaving.

In the next step, we identify employees that are likely to stay in the

company by using the attrition model's predicted probability :

- P_leave = model.predict_proba(X)[i][1]

- P_stay = 1 – P_leave

We define a threshold, like in this case if P_stay is greater than 0.6, then we say that the employee is likely to stay. We then only predict future salary for these employees in order to understand how valuable they can be to the company.

Finally, we combine both classification and regression to answer the question : "If someone leaves, how much value (in ₹ or $) are we **losing**?". We make use of the following formulas :

For each employee: Expected loss(i) = P(attrition(i)) * FutureSalary(i)

Aggregate across all employees: Total expected loss = sum of expected loss for each employee

This tells us how valuable each employee that is likely to leave would be to the company and the total expected loss incurred. We finally have our answer.

Results generated :-

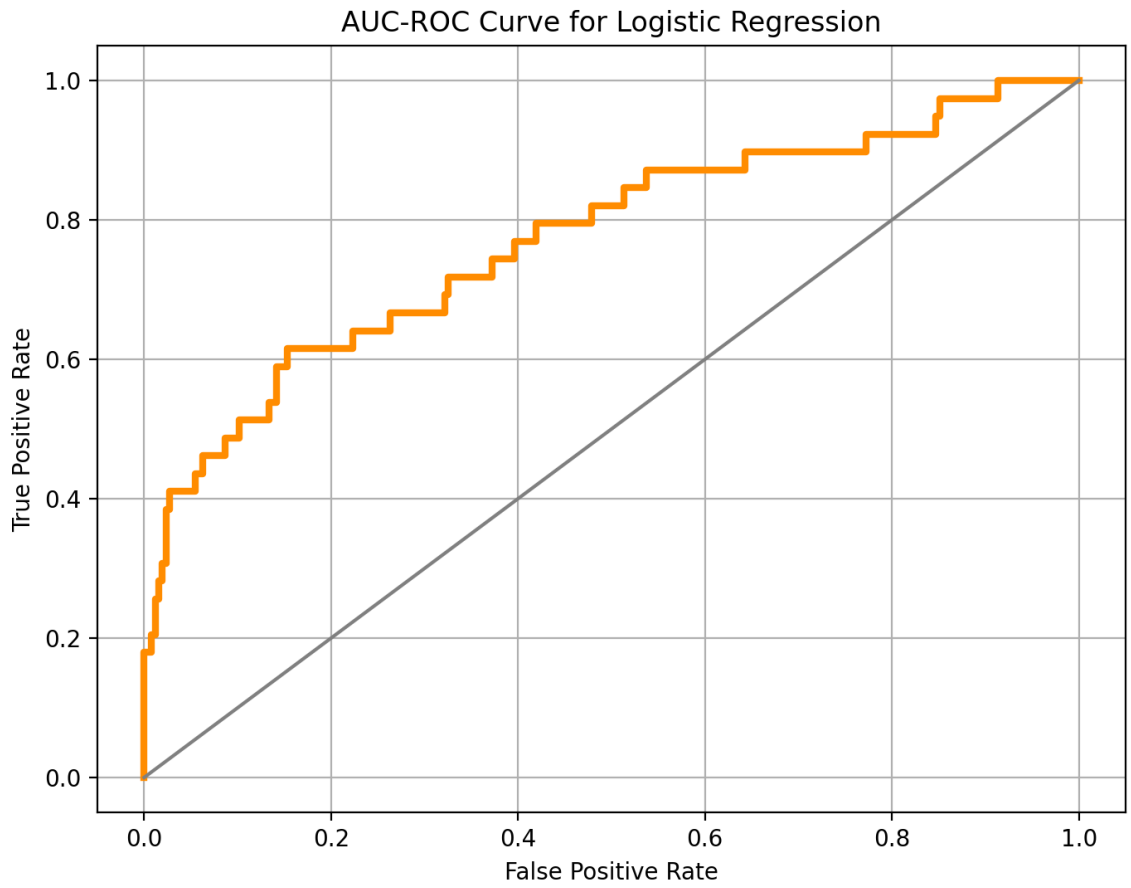
Part 1 (Attrition Prediction) :

After running the three models mentioned above, these were the results that showed up…

Logistic Regression :

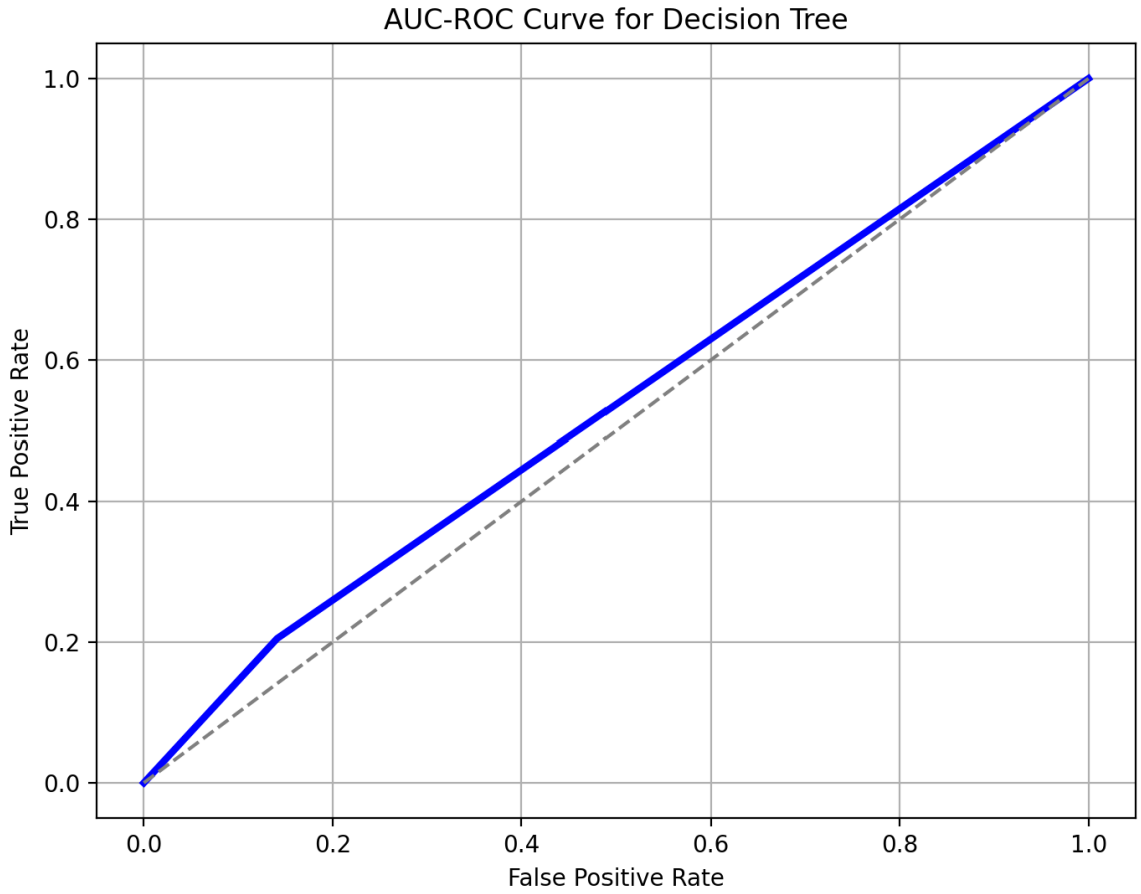
AUC : 0.7713
F1 Score : 0.4746

AUC-ROC Curve



Decision Tree :

AUC : 0.532
F1 score : 0.1928

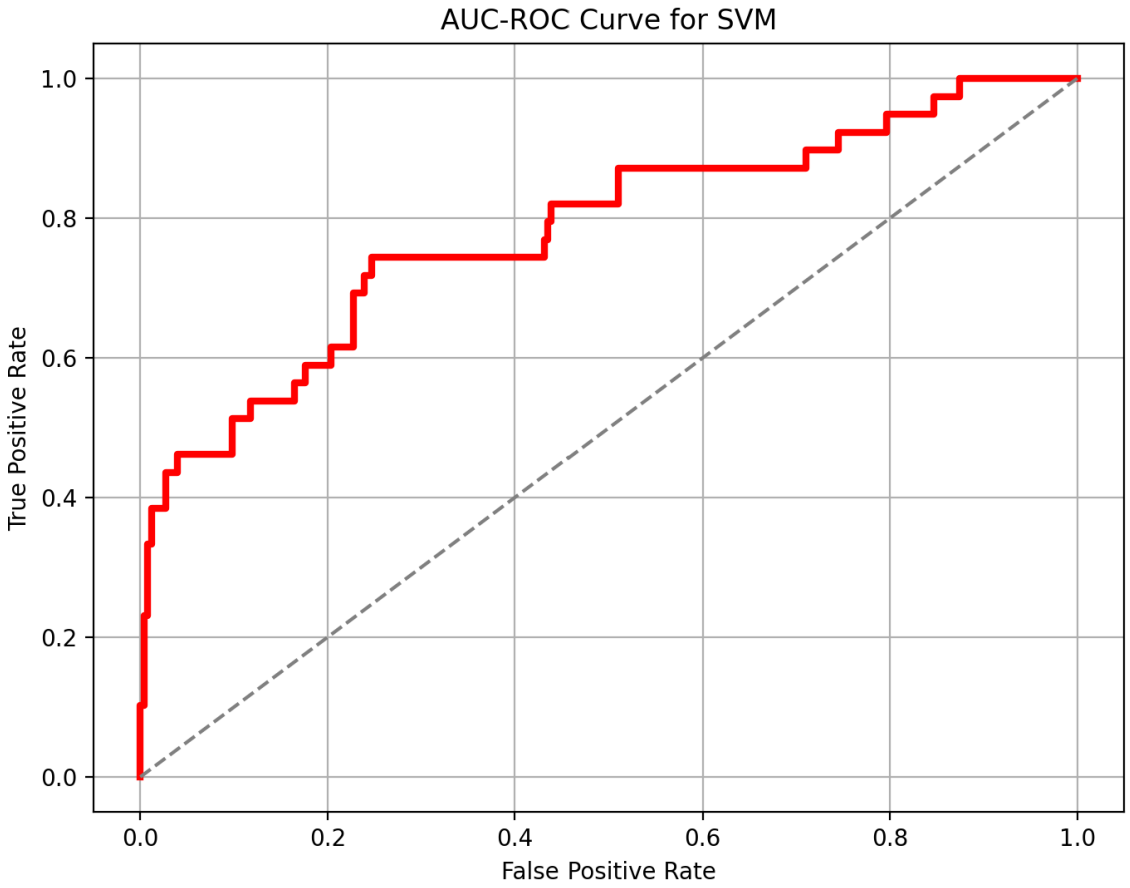


SVM :

AUC : 0.7822

F1 score : 0.4727

Accuracy : 0.9013



Part 2 :

Future salaries for the employees were generated :-

0	6292.65
1	5643.00
2	2194.50
3	3054.45
4	3641.40
	...

1465 2699.55

1466 10490.55

1467 6756.20

1468 5659.50

1469 4624.20

Note : Index 0 is the first person, and there are 1470 employees in total.

Part 3 :

After running the four models mentioned above, these were the results…

Random Forest Regressor :-

R2 score : 0.9399

RMSE : 1239.18396

Lasso Regressor :-

R2 score : 0.9035

RMSE : 1570.3977

Ridge Regressor :-

R2 score : 0.90348

RMSE : 1570.5501

SVR :-

R2 score : -0.10879

RMSE : 5323.24

From Part 3, we concluded that the **Random Forest Regressor** model worked the best as it had the highest R2 score as well as the lowest RMSE, and so we used it for Part 4 and Part 5 as well, just to achieve better predictions and results.

Part 4 :

In this part, we found out the number of employees likely to stay and predicted their salaries.

There were **1233** employees that were likely to stay.

Part 5 :

The total expected loss, after adding up the expected loss for each employee that is unlikely to stay, was $1378389.14129.

Conclusion :-

As we worked on this project, we found some interesting points regarding it :

1. The SVM model worked the best for classification purposes with regards to accuracy and AUC parameters, however it had a **pretty low F1-score** which could indicate that many of the positive predictions were actually incorrect, or that the model missed many actual positives. These problems may arise due to factors like imbalanced datasets, model bias, over fitting and under fitting.

2. In terms of regression, the **Random Forest Regressor** performed the best and hence is a more reliable model than the others for performing regression on a dataset. This is far opposed to the results the Decision Tree model showed us, and it's ironic how a Random Forest is *multiple decision trees at once*!

3. Certain limitations exist with these models as evident by the poor F1 score of the classification models despite having a decent AUC and accuracy. These could be fixed by implementing better models like XGBoost or LightGBM, or by tuning the thresholds.

This model was trained on historical data from a single department, which may limit generalizability across the company. Regardless of all the problems faced by the model currently, it remains a crucial step in highlighting the importance of integrating predictive models into HR planning to reduce potential attrition costs and improve workforce stability.