



11/30/2021

Classification Analysis – Employee attrition

Olivier FONTAINE

TABLE OF CONTENTS

Analysis objective.....	2
Data set description	2
Data exploration, data cleaning & feature engineering.....	5
Missing values.....	5
Unnecessary features	6
Outliers	8
Feature engineering.....	10
Scaling	12
Encoding	13
Findings from exploratory analysis.....	14
Hypothesis testing.....	17
Training Classifier models	18
Model recommendation.....	21
Key Findings and Insights	22
Suggestions for next steps in analyzing this data	22
Appendix: code.....	23

ANALYSIS OBJECTIVE

- ➔ Main objective of the analysis that specifies whether your model will be focused on prediction or interpretation and the benefits that your analysis provides to the business or stakeholders of this data.

The objective of this analysis is to find the features that are the most important to explain employee attrition in a company.

So, here we are more focusing on interpretation than prediction.

For the HR department and senior management of the company, the benefits will be the potential elaboration of retention strategies based on the key findings of this analysis.

DATA SET DESCRIPTION

- ➔ Brief description of the data set you chose, a summary of its attributes, and an outline of what you are trying to accomplish with this analysis.

The data set name is *IBM HR Analytics Employee Attrition and Performance*. This a fictional data set created by IBM data scientists. The data set allow uncovering the factors that lead to employee attrition. The data set is available on KAGGLE: <https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset>

The data set has **1470 rows** and **35 columns**.

The column **Attrition** is the target variable.

Here is the list of columns:

- Certain **numerical** columns contain **ordinal** data and the meaning of the numerical value is described on the KAGGLE link. I copied the values in the tab below.
- For certain **numerical** data, the unit is unknown and for others like levels, the meaning is unknown as well.
- For **categorical** data, I added, in the tab below, the different values for each categorical features.

Column	Features	Data type	Data type	Values
0	Age	int64	Numerical	In years
1	Attrition	object	Categorical	'Yes' 'No'
2	BusinessTravel	object	Categorical	'Travel_Rarely' 'Travel_Frequently' 'Non-Travel'
3	DailyRate	int64	Numerical	In a currency not known
4	Department	object	Categorical	'Sales' 'Research & Development' 'Human Resources'
5	DistanceFromHome	int64	Numerical	In a distance unit not known

Column	Features	Data type	Data type	Values
6	Education	int64	Ordinal	1 'Below College' 2 'College' 3 'Bachelor' 4 'Master' 5 'Doctor'
7	EducationField	object	Categorical	'Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree' 'Human Resources'
8	EmployeeCount	int64	Numerical	Counter: 1 for each row
9	EmployeeNumber	int64	Numerical	Probably Employee ID
10	EnvironmentSatisfaction	int64	Ordinal	1 'Low' 2 'Medium' 3 'High' 4 'Very High'
11	Gender	object	Categorical	'Female' 'Male'
12	HourlyRate	int64	Numerical	In a currency not known
13	JobInvolvement	int64	Ordinal	1 'Low' 2 'Medium' 3 'High' 4 'Very High'
14	JobLevel	int64	Ordinal	From 1 to 5 but the meaning of the level is not known.
15	JobRole	object	Categorical	'Sales Executive' 'Research Scientist' 'Laboratory Technician' 'Manufacturing Director' 'Healthcare Representative' 'Manager' 'Sales Representative' 'Research Director' 'Human Resources'
16	JobSatisfaction	int64	Ordinal	1 'Low' 2 'Medium' 3 'High' 4 'Very High'
17	MaritalStatus	object	Categorical	'Single' 'Married'

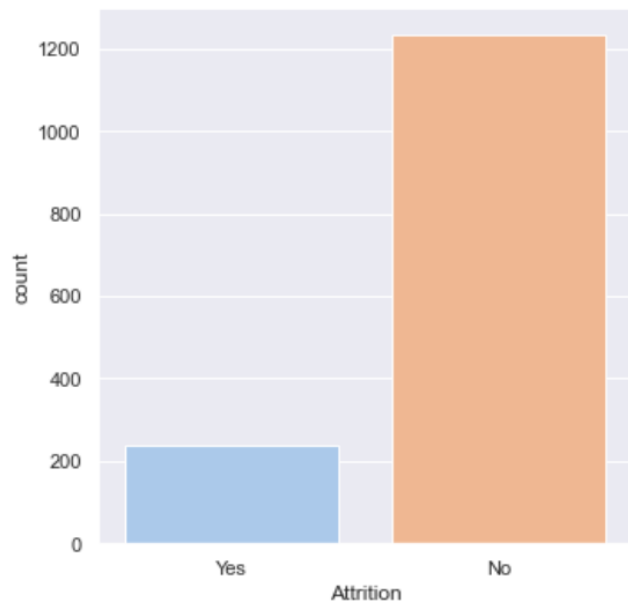
Column	Features	Data type	Data type	Values
				'Divorced'
18	MonthlyIncome	int64	Numerical	In a currency not known
19	MonthlyRate	int64	Numerical	In a currency not known
20	NumCompaniesWorked	int64	Numerical	Integer
21	Over18	object	Categorical	Y
22	OverTime	object	Categorical	'Yes' 'No'
23	PercentSalaryHike	int64	Numerical	%
24	PerformanceRating	int64	Ordinal	1 'Low' 2 'Good' 3 'Excellent' 4 'Outstanding'
25	RelationshipSatisfaction	int64	Ordinal	1 'Low' 2 'Medium' 3 'High' 4 'Very High'
26	StandardHours	int64	Numerical	In hours
27	StockOptionLevel	int64	Ordinal	From 0 to 3 but the meaning of the level is not known.
28	TotalWorkingYears	int64	Numerical	In years
29	TrainingTimesLastYear	int64	Numerical	In times
30	WorkLifeBalance	int64	Ordinal	1 'Bad' 2 'Good' 3 'Better' 4 'Best'
31	YearsAtCompany	int64	Numerical	In years
32	YearsInCurrentRole	int64	Numerical	In years
33	YearsSinceLastPromotion	int64	Numerical	In years
34	YearsWithCurrManager	int64	Numerical	In years

Attrition is the target variable and has 2 outcomes:

- **Yes**, the employee leaves the company
- **No**, the employee stays in the company

Thus, the problem is a **binary classification** problem.

Let's see now the distribution between employees who leaves and those who stays in the company:



This distribution is **imbalanced**:

- 237 employee leaves the company (16%)
- 1233 employees stays in the company (84%)

So, in this project we will look for the following metric Recall as accuracy is not a good metric for imbalanced classification.

After data exploration, data processing, dataset splitting, we will test several models, perform hyperparameters tuning and evaluate them.

DATA EXPLORATION, DATA CLEANING & FEATURE ENGINEERING

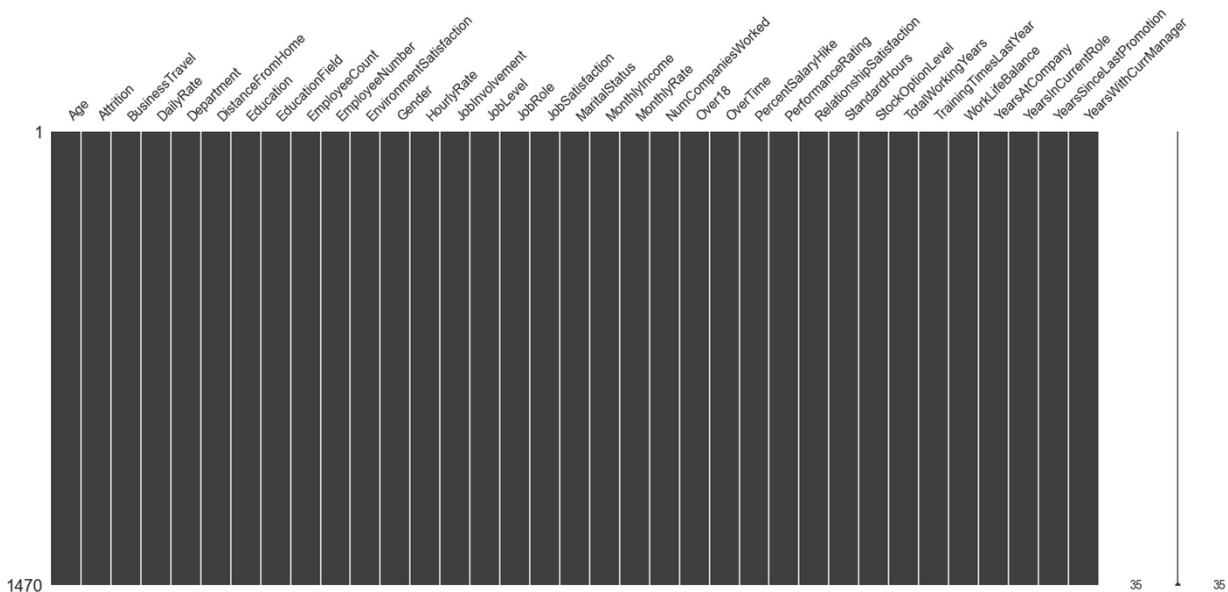
➔ Brief summary of data exploration and actions taken for data cleaning and feature engineering.

MISSING VALUES

In this dataset, there are no missing values.

Indeed, there are no blank in the following missing data chart:

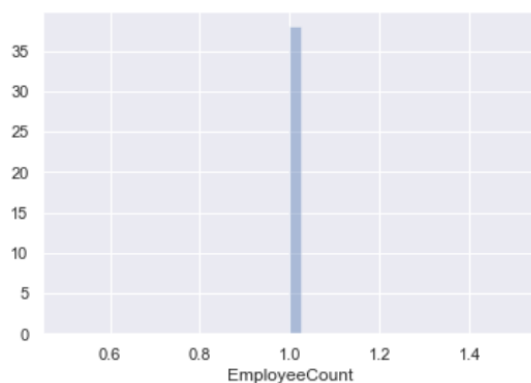
Classification Analysis — Employee attrition



UNNECESSARY FEATURES

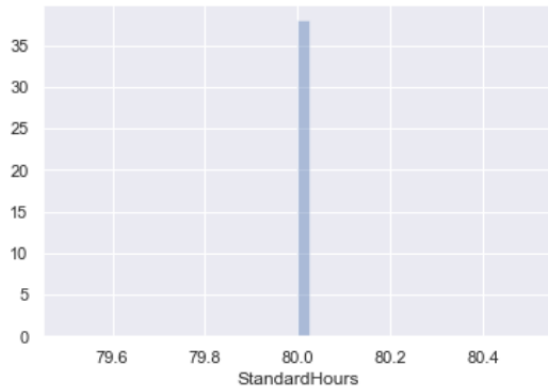
Let's see if the dataset contains unnecessary features.

Here is the distribution of feature **EmployeeCount**:



There is a unique value for all samples (probably an employee counter), so this feature can be removed as it does not bring any valuable information.

It is exactly the same with feature **StandardHours**:



All employees have 80 standard hours, so this feature can be removed as it does not bring any valuable information.

The categorical feature **Over18** has only 1 value: Y for Yes

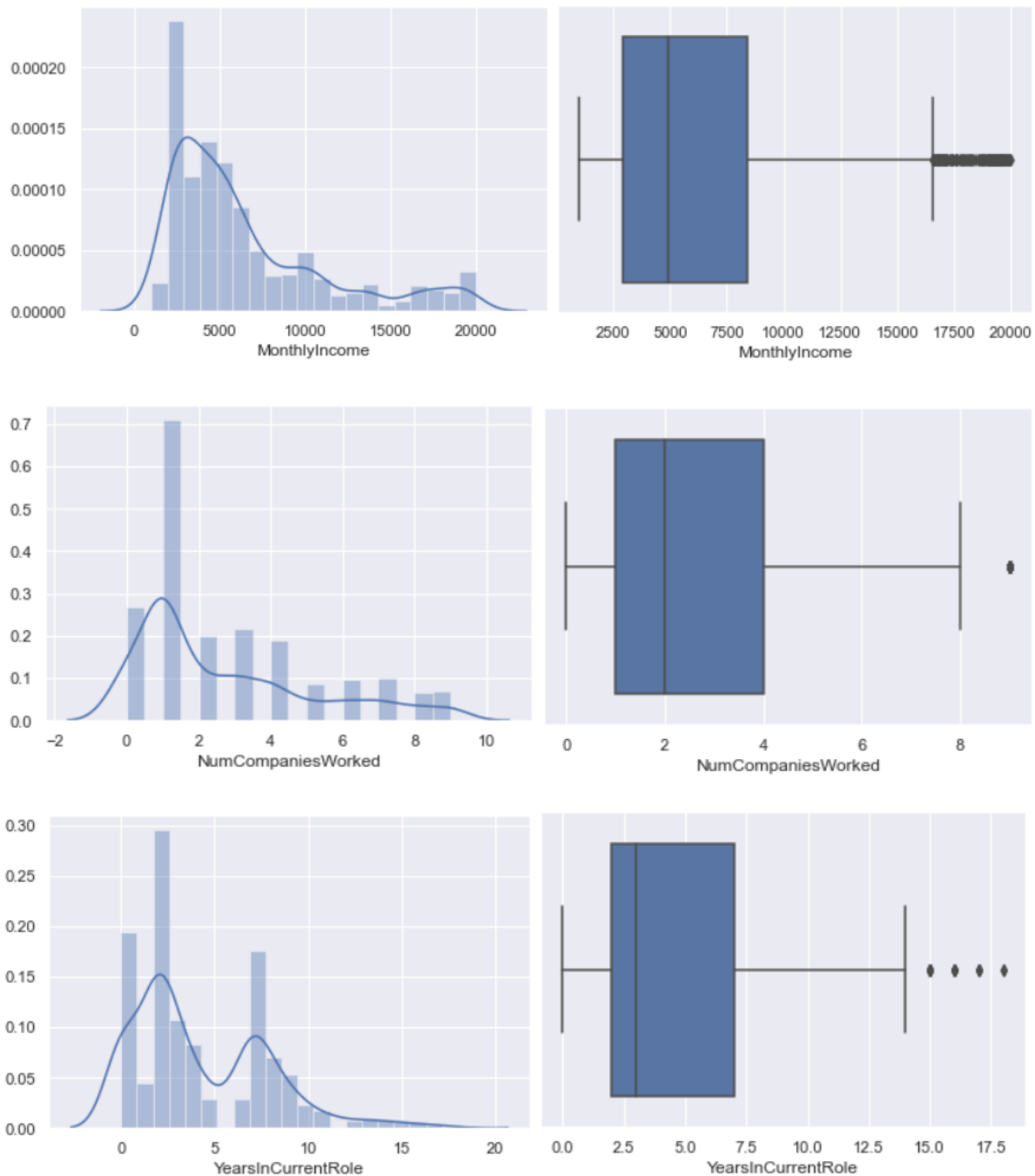
As all employees are over 18 years, this feature can be removed as it does not bring any valuable information.

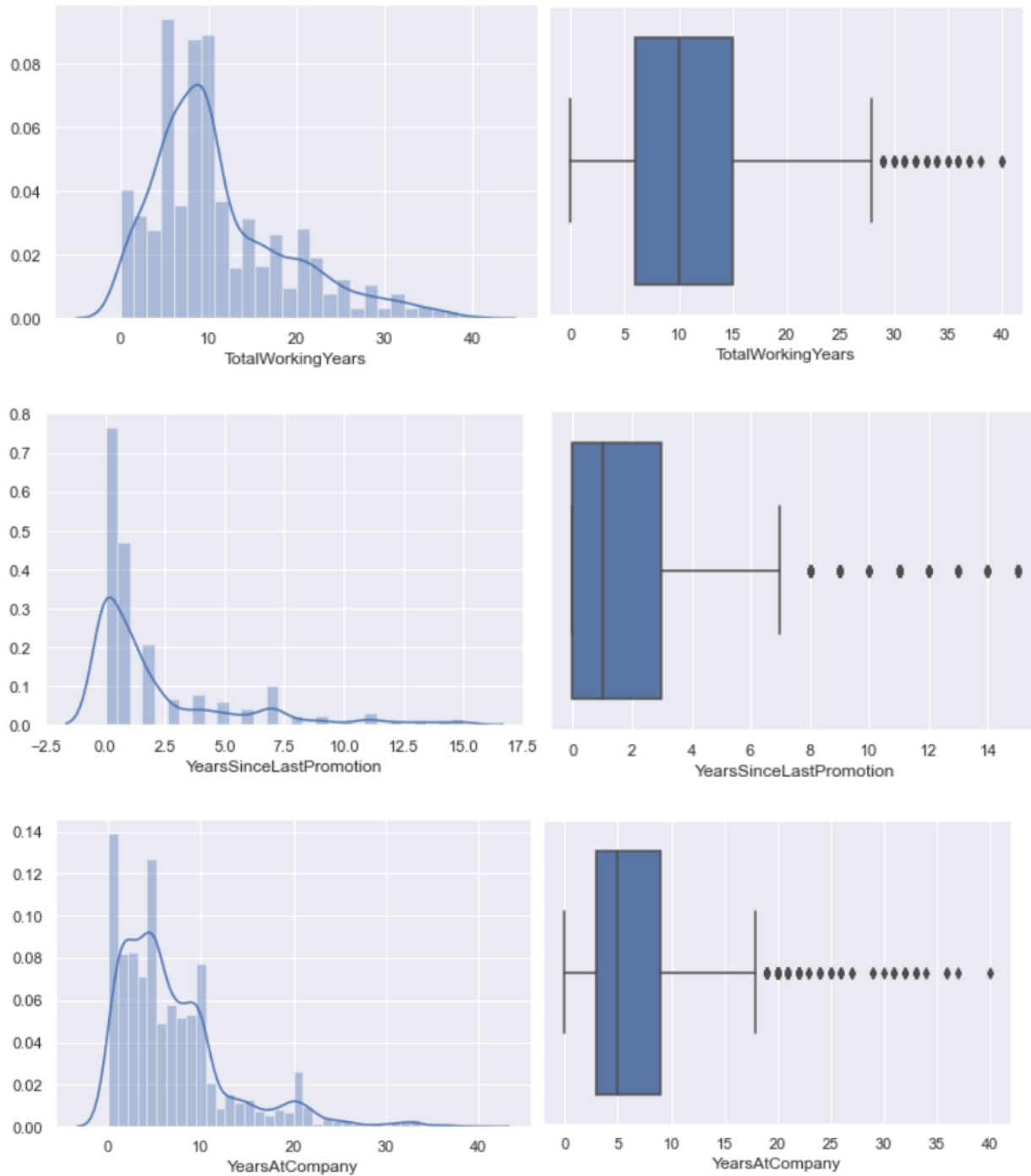
The feature **EmployeeNumber** contains the Employee ID and thus does not bring any valuable information. But, before removing it, it can be interesting to check for duplicates.

After verification, there are no duplicates on the feature **EmployeeNumber**, so this feature can be removed.

OUTLIERS

Now, regarding outliers, we have the following features with outliers:





MonthlyIncome, TotalWorkingYears, YearsSinceLastPromotion, YearsAtCompany have a lot of outliers.

At this stage, I keep the outliers with the assumption that I will focus later on models that are resistant to outliers.

FEATURE ENGINEERING

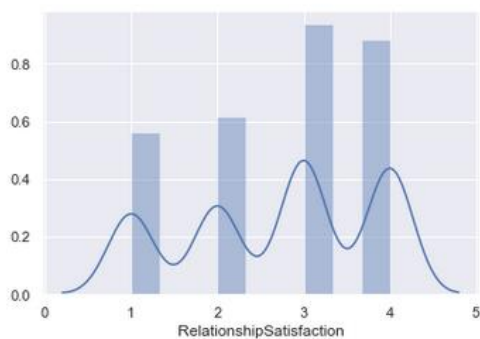
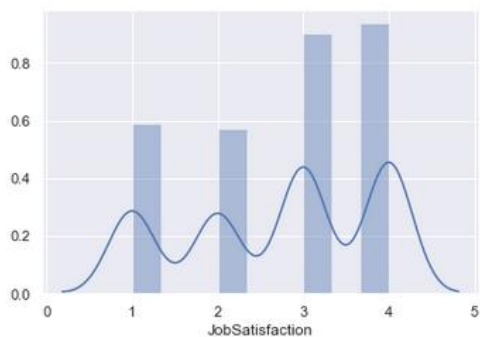
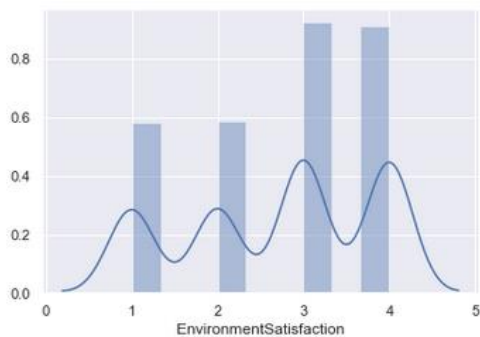
I notice that there are 3 features related to satisfaction:

- EnvironmentSatisfaction
- JobSatisfaction
- RelationshipSatisfaction

They are all ordinal data with the same possible values:

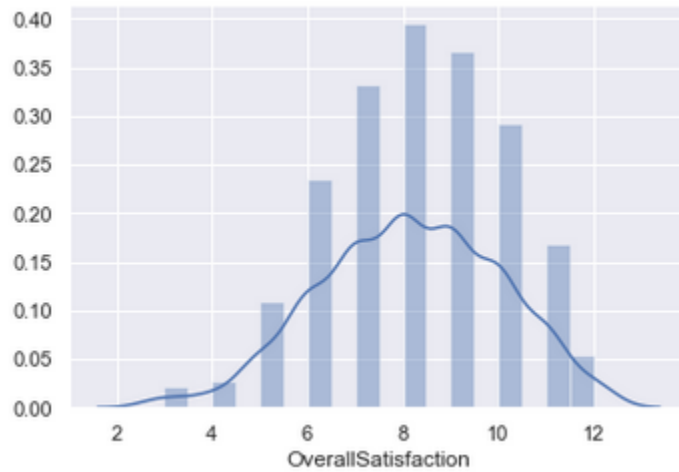
- 1 - 'Low'
- 2 - 'Medium'
- 3 - 'High'
- 4 - 'Very High'

Here are their distribution:



I can group them in a single one, we can name OverallSatisfaction. This new feature will contain the sum of the 3 types of satisfaction.

Here is the distribution of the new feature:



This looks like a nice normal distribution!

I could go deeper in feature engineering by regrouping values with few samples in categorical features.

SCALING

Usually, numerical data of a dataset have different scales.

This is the case here where features **MonthlyIncome** and **MonthlyRate** have a bigger scales than other features:

	mean	std	min	25%	50%	75%	max
Age	36.923810	9.135373	18.000000	30.000000	36.000000	43.000000	60.000000
DailyRate	802.485714	403.509100	102.000000	465.000000	802.000000	1157.000000	1499.000000
DistanceFromHome	9.192517	8.106864	1.000000	2.000000	7.000000	14.000000	29.000000
HourlyRate	65.891156	20.329428	30.000000	48.000000	66.000000	83.750000	100.000000
MonthlyIncome	6502.931293	4707.956783	1009.000000	2911.000000	4919.000000	8379.000000	19999.000000
MonthlyRate	14313.103401	7117.786044	2094.000000	8047.000000	14235.500000	20461.500000	26999.000000
NumCompaniesWorked	2.693197	2.498009	0.000000	1.000000	2.000000	4.000000	9.000000
PercentSalaryHike	15.209524	3.659938	11.000000	12.000000	14.000000	18.000000	25.000000
TotalWorkingYears	11.279592	7.780782	0.000000	6.000000	10.000000	15.000000	40.000000
TrainingTimesLastYear	2.799320	1.289271	0.000000	2.000000	3.000000	3.000000	6.000000
YearsAtCompany	7.008163	6.126525	0.000000	3.000000	5.000000	9.000000	40.000000
YearsInCurrentRole	4.229252	3.623137	0.000000	2.000000	3.000000	7.000000	18.000000
YearsSinceLastPromotion	2.187755	3.222430	0.000000	0.000000	1.000000	3.000000	15.000000
YearsWithCurrManager	4.123129	3.568136	0.000000	2.000000	3.000000	7.000000	17.000000

Later, I will probably evaluate models that require scaled features. So, I will need to proceed to numerical data scaling.

ENCODING

In the dataset, there are categorical data and ordinal data. They need to be encoded before they can be included in certain models.

Categorical data overview before encoding:

	count	unique	top	freq
Attrition	1470	2	No	1233
BusinessTravel	1470	3	Travel_Rarely	1043
Department	1470	3	Research & Development	961
EducationField	1470	6	Life Sciences	606
Gender	1470	2	Male	882
JobRole	1470	9	Sales Executive	326
MaritalStatus	1470	3	Married	673
OverTime	1470	2	No	1054

Ex with the first 5 lines of the dataset:

	Attrition	BusinessTravel	Department	EducationField	Gender	JobRole	MaritalStatus	OverTime
0	Yes	Travel_Rarely	Sales	Life Sciences	Female	Sales Executive	Single	Yes
1	No	Travel_Frequently	Research & Development	Life Sciences	Male	Research Scientist	Married	No
2	Yes	Travel_Rarely	Research & Development	Other	Male	Laboratory Technician	Single	Yes
3	No	Travel_Frequently	Research & Development	Life Sciences	Female	Research Scientist	Married	Yes
4	No	Travel_Rarely	Research & Development	Medical	Male	Laboratory Technician	Married	No

➔ I will one hot encode those categorical features

Ordinal data overview before encoding:

	count	mean	std	min	25%	50%	75%	max
Education	1470.0	2.912925	1.024165	1.0	2.0	3.0	4.0	5.0
EnvironmentSatisfaction	1470.0	2.721769	1.093082	1.0	2.0	3.0	4.0	4.0
JobInvolvement	1470.0	2.729932	0.711561	1.0	2.0	3.0	3.0	4.0
JobLevel	1470.0	2.063946	1.106940	1.0	1.0	2.0	3.0	5.0
JobSatisfaction	1470.0	2.728571	1.102846	1.0	2.0	3.0	4.0	4.0
PerformanceRating	1470.0	3.153741	0.360824	3.0	3.0	3.0	3.0	4.0
RelationshipSatisfaction	1470.0	2.712245	1.081209	1.0	2.0	3.0	4.0	4.0
StockOptionLevel	1470.0	0.793878	0.852077	0.0	0.0	1.0	1.0	3.0
WorkLifeBalance	1470.0	2.761224	0.706476	1.0	2.0	3.0	3.0	4.0

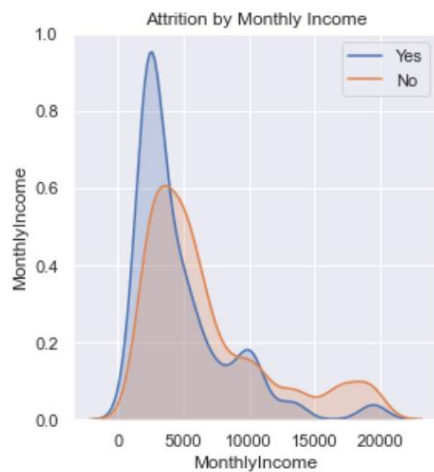
➔ I will encode these ordinal data with ordinal transformer

FINDINGS FROM EXPLORATORY ANALYSIS

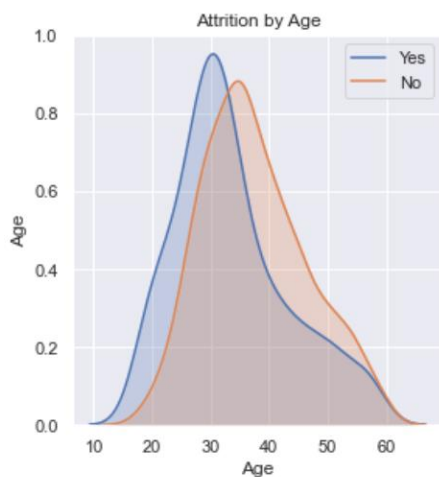
Employees with lower monthly income are more likely to leave the company.

The market is probably more attractive for those employees.

Increasing lowest salaries could reduce attrition.

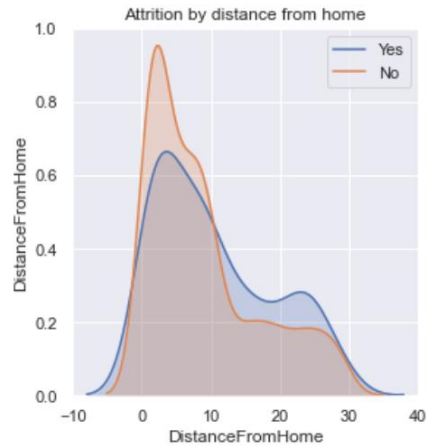


Young employees are more likely to leave the company. There might be a correlation also with monthly income, to be investigated:



Employees who live next to the company are less likely to leave the company:

Classification Analysis — Employee attrition



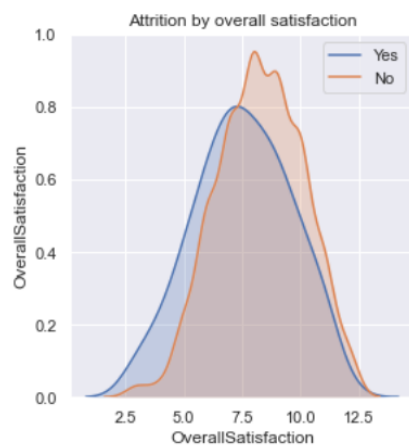
Majority of employees work in the Research & Development Department.

This department has the highest numbers of leavers, probably because it has the higher number of employees.

Actually, employees in Sales and HR are more likely to quit:



Employees with overall satisfaction (new created feature) are less likely to quit:

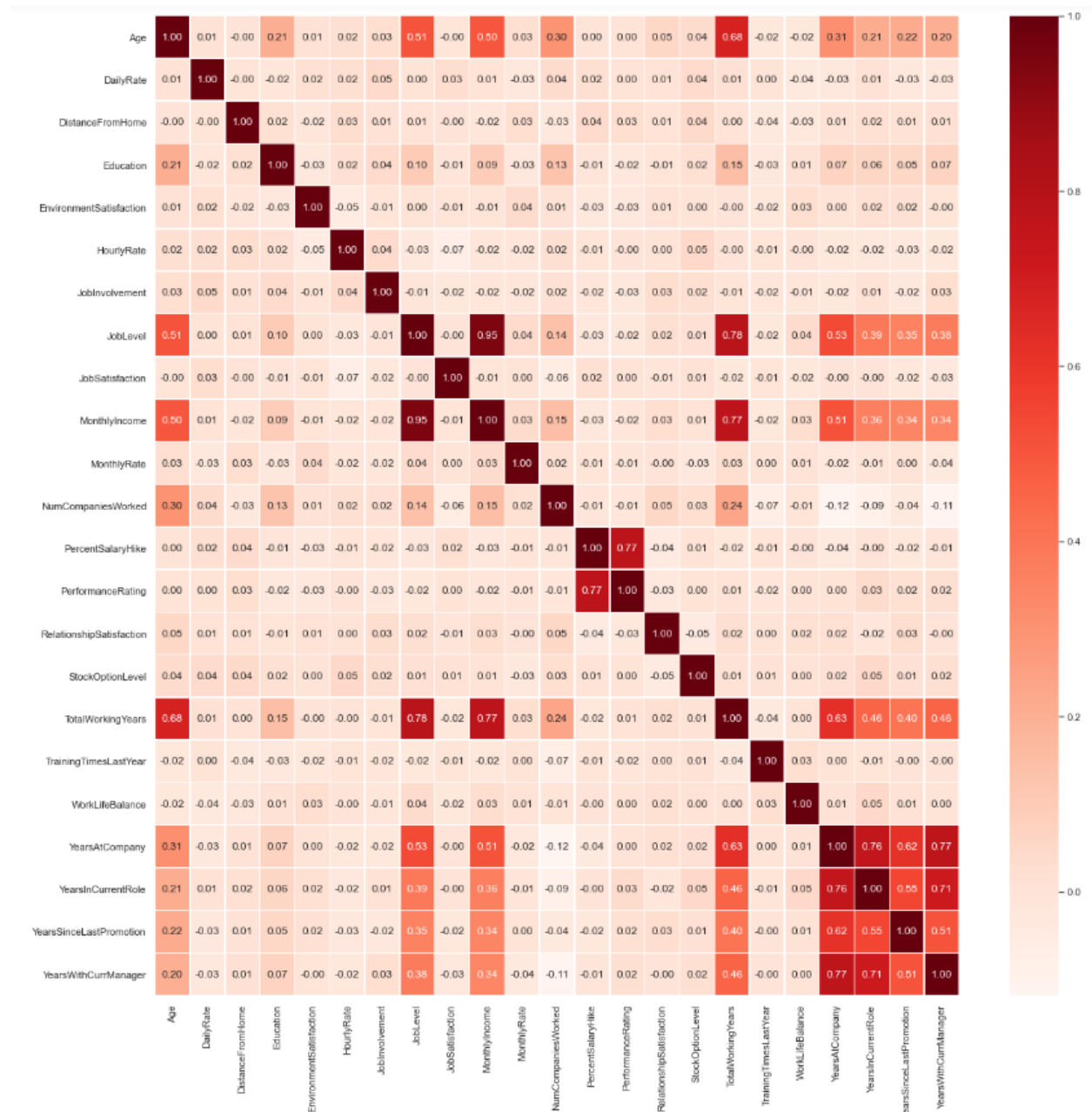


Classification Analysis — Employee attrition

From the correlation analysis, we can see that:

- Monthly Income and Job Level are strongly correlated (0,95)
- Total Working Years are significantly correlated with
 - o Job Level (0,78)
 - o Monthly Income (0,77)
 - o Age (0,68)
- Performance Rating and Percentage Salary Hike are significantly correlated (0,77)

There are also a lot of correlation with features linked to years of service as a whole.



- ➔ As we have correlated data, we will probably need later to select the features we keep for models evaluation.

HYPOTHESIS TESTING

I formulate the following hypothesis:

Null hypothesis: Group of employees who stays and leaves the company have the same **overall satisfaction** mean.

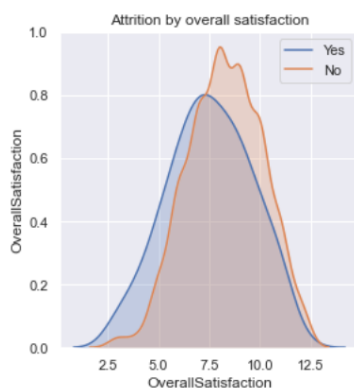
Alternative hypothesis: Group of employees who stays and leaves the company do not have the same **overall satisfaction** mean.

I will proceed to a **two sample T-test**, also known as the independent samples T-test.

This type of statistical test compares two averages (means) and will give us information if these two means are statistically different from each other. The t-test also tells you whether the differences are statistically significant. In other words it lets you know if those differences could have happened by chance.

I will have a look now on the assumptions of this parametric test:

- Assumption 1: Are the two samples independent? Yes, employees who stays and those who leaves are different.
- Assumption 2: Are the data from each of the 2 groups following a normal distribution? Yes



- Assumption 3: Do the two samples have the same variances (Homogeneity of Variance)? Yes

OverallSatisfaction								
	count	mean	std	min	25%	50%	75%	max
Attrition								
No	1233.0	8.283861	1.824849	3.0	7.0	8.0	10.0	12.0
Yes	237.0	7.531646	2.061566	3.0	6.0	8.0	9.0	12.0

I decide to fix the significance level $\alpha = 5\%$.

The two sample T-test gives a P value = P-value = 1.56e-08

The P-value of the test is less than the significance level alpha (e.g., 0.05). I reject the null hypothesis.

- ➔ This means that I can conclude that average overall satisfaction of leavers is statistically different from the average overall satisfaction of employees who stays in the company.

TRAINING CLASSIFIER MODELS

- ➔ Summary of training at least three different classifier models, preferably of different nature in explainability and predictability. For example, you can start with a simple logistic regression as a baseline, adding other models or ensemble models. Preferably, all your models use the same training and test splits, or the same cross-validation method.

5 classification models are evaluated:

- Logistic Regression
- KNN
- Linear SVM
- Decision Tree
- Random Forest

For imbalanced class, it is best practice to do a stratified test-train split, to ensure that we keep that balance for both our train and test set, in regards to those unbalanced classes.

Then, I could have introduced data sampling techniques to improve the performance of algorithms (not done in this project).

Then, I performed hyperparameters tuning on each of the selected algorithms. We use GridSearchCV with StratifiedKfold in order to preserve the imbalanced class distribution in each fold.

For each algorithm, I tried:

- A null '0' version with no parameters
- A version with hyperparameters

Here are the set of parameters tested for selected algorithms:

```
# Hyperparameters for each model
lr0_params = {}
knn0_params = {}
knn_params = {'KNN__n_neighbors': list(range(1,10))}
svc0_params = {}
svc_params = {'SVC__C': [0.1,1, 10, 100], 'SVC__gamma': [1,0.1,0.01,0.001], 'SVC__kernel': ['rbf', 'poly', 'sigmoid']}
cart0_params = {}
cart_params = {'CART__max_depth':list(range(1,7)),
               'CART__max_features': list(range(1,10))}
rf0_params = {}
rf_params = {'RF__n_estimators': [5, 10, 50, 100],
             'RF__max_depth':list(range(1,15)),
             'RF__max_features': list(range(1,10))}
```

Here are the results with the best estimators and best parameters:

```
Best parameters Logistic Regression :
{}
Best parameters K-nearest neighbors 0 :
{}
Best parameters K-nearest neighbors best estimator :
{'KNN__n_neighbors': 5}
Best parameters SVC 0 :
{}
Best parameters SVC best estimator :
{'SVC__C': 10, 'SVC__gamma': 0.01, 'SVC__kernel': 'rbf'}
Best parameters Decision Tree Classifier 0 :
{}
Best parameters Decision Tree Classifier best estimator :
{'CART__max_depth': 3, 'CART__max_features': 8}
Best parameters Random Forest Classifier 0 :
{}
Best parameters Random Forest Classifier best estimator :
{'RF__max_depth': 8, 'RF__max_features': 9, 'RF__n_estimators': 10}
```

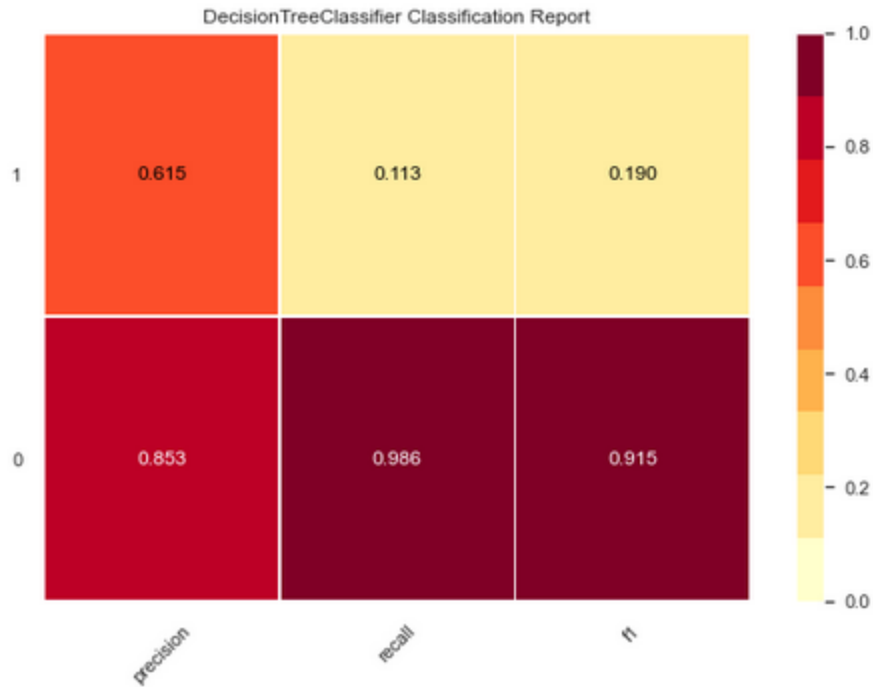
For each algorithm, following information are calculated:

- Train accuracy
- Test accuracy
- Precision
- Recall
- AUC (Area Under the Curve)

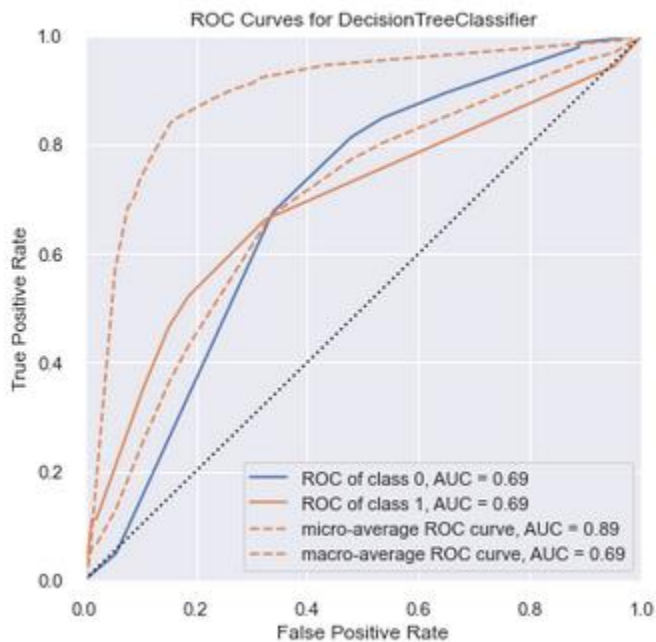
	Model name	Train Accuracy	Test Accuracy	Precision	Recall	AUC
0	Logistic Regression	0.8960	0.8821	0.720930	0.436620	0.702094
4	SVC best estimator	0.9349	0.8776	0.707317	0.408451	0.688009
7	Random Forest Classifier 0	1.0000	0.8526	0.666667	0.169014	0.576399
3	SVC 0	0.9018	0.8503	0.692308	0.126761	0.557975
6	Decision Tree Classifier best estimator	0.8581	0.8458	0.615385	0.112676	0.549581
8	Random Forest Classifier best estimator	0.9456	0.8435	0.550000	0.154930	0.565303
1	K-nearest neighbors 0	0.8678	0.8413	0.529412	0.126761	0.552569
2	K-nearest neighbors best estimator	0.8678	0.8413	0.529412	0.126761	0.552569
5	Decision Tree Classifier 0	1.0000	0.7732	0.347368	0.464789	0.648611

For imbalanced class, the Precision Recall curve makes more sense than the ROC curve.

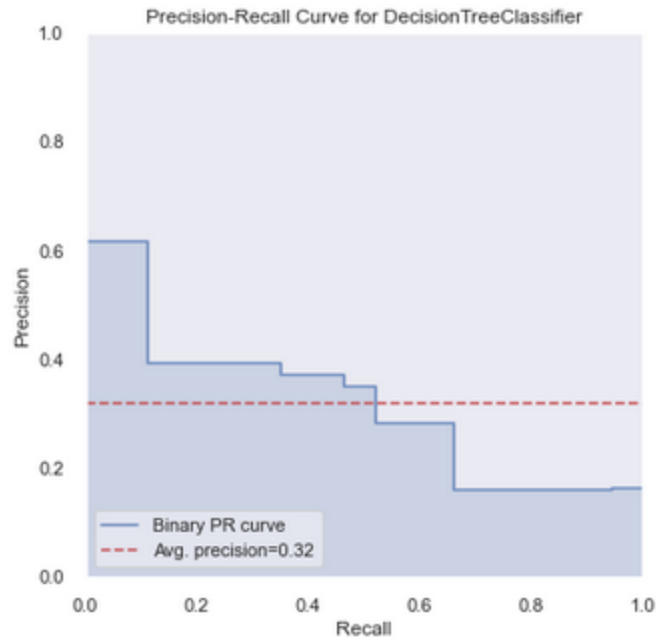
Here is the confusion matrix for the Decision Tree classifier



Example of ROC curve for the Decision Tree classifier:



And the Precision Recall curve for the same model:



MODEL RECOMMENDATION

- ➔ A paragraph explaining which of your classifier models you recommend as a final model that best fits your needs in terms of accuracy and explainability.

I recommend the Logistic Regression model as it is the one with the best recall and highest precision with good performance on train and test set.

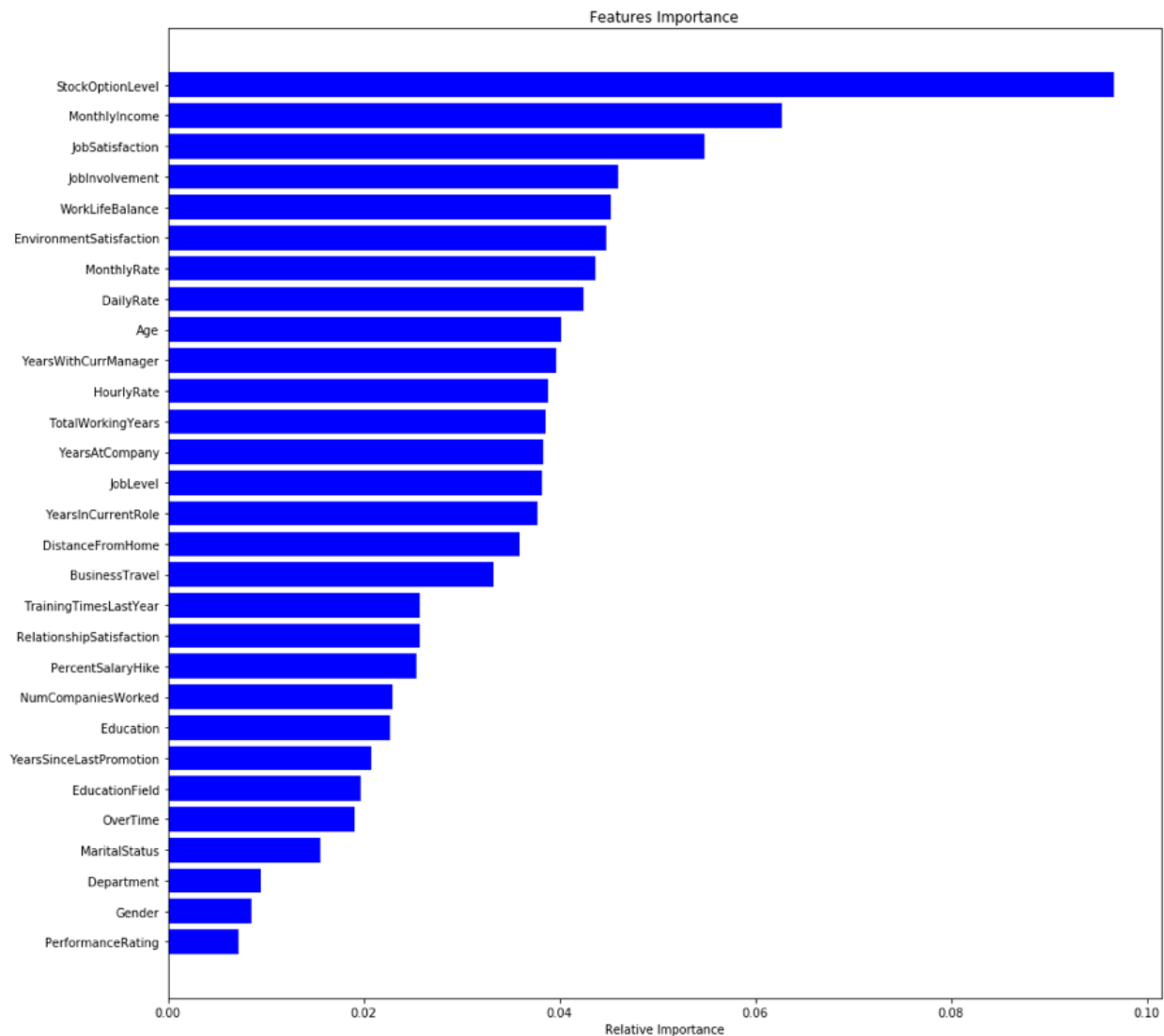
	Model name	Train Accuracy	Test Accuracy	Precision	Recall	AUC
0	Logistic Regression	0.8960	0.8821	0.720930	0.438620	0.702094
4	SVC best estimator	0.9349	0.8776	0.707317	0.408451	0.688009
7	Random Forest Classifier 0	1.0000	0.8526	0.666667	0.169014	0.576399
3	SVC 0	0.9018	0.8503	0.692308	0.126761	0.557975
6	Decision Tree Classifier best estimator	0.8581	0.8458	0.615385	0.112676	0.549581
8	Random Forest Classifier best estimator	0.9456	0.8435	0.550000	0.154930	0.565303
1	K-nearest neighbors 0	0.8678	0.8413	0.529412	0.126761	0.552569
2	K-nearest neighbors best estimator	0.8678	0.8413	0.529412	0.126761	0.552569
5	Decision Tree Classifier 0	1.0000	0.7732	0.347368	0.464789	0.648611

Logistic Regression is also good algorithm for interpretability.

KEY FINDINGS AND INSIGHTS

- ➔ Summary Key Findings and Insights, which walks your reader through the main drivers of your model and insights from your data derived from your classifier model.

So, the key feature that explain attrition are the following.



Knowing that the HR department can implement retention strategies in order to avoid unnecessary turnover and associated costs.

SUGGESTIONS FOR NEXT STEPS IN ANALYZING THIS DATA

- ➔ Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model after adding specific data features that may help you achieve a better explanation or a better prediction.

Additional actions can be taken for preprocessing:

- Feature selection.
- Additional feature engineering

I could have implemented data sampling techniques to improve the performance of algorithms.

I could have improved regularization on certain models.

I could have better defined the optimal threshold tuning for classification.

It could be good to get additional data as the dataset is quite small and this is a bit difficult when working on an imbalanced dataset.

APPENDIX: CODE

The code for this project is available on GITHUB:

<https://github.com/Olivier-FONTAINE/IBM-Machine-Learning-professional-certificate/blob/main/03-CLA-Employee%20attrition.ipynb>