Project Report
Programming for Data Science
BUAN 6340.502

Employee Retention Model

By: Neel Gandhi, Sherif Elnagdy, Mayank Yadav, Lucia Cha

Project Description

Software companies are increasingly employing data scientists to build retention models because the IT industry is undergoing a paradigm shift from “Exit Interviews” to a more proactive approach. Employee retention is important because it is costly to the company to train and groom the employee to be corporate-ready. There are potential consequences involved when employees quit, such as excessive job responsibility their coworkers must absorb, insecure employees, and time/cost involved in recruiting, hiring, and training a new employee.

Dataset

Provided by Infinite Computing Systems, Inc (Cedar Rapids, IA, US)

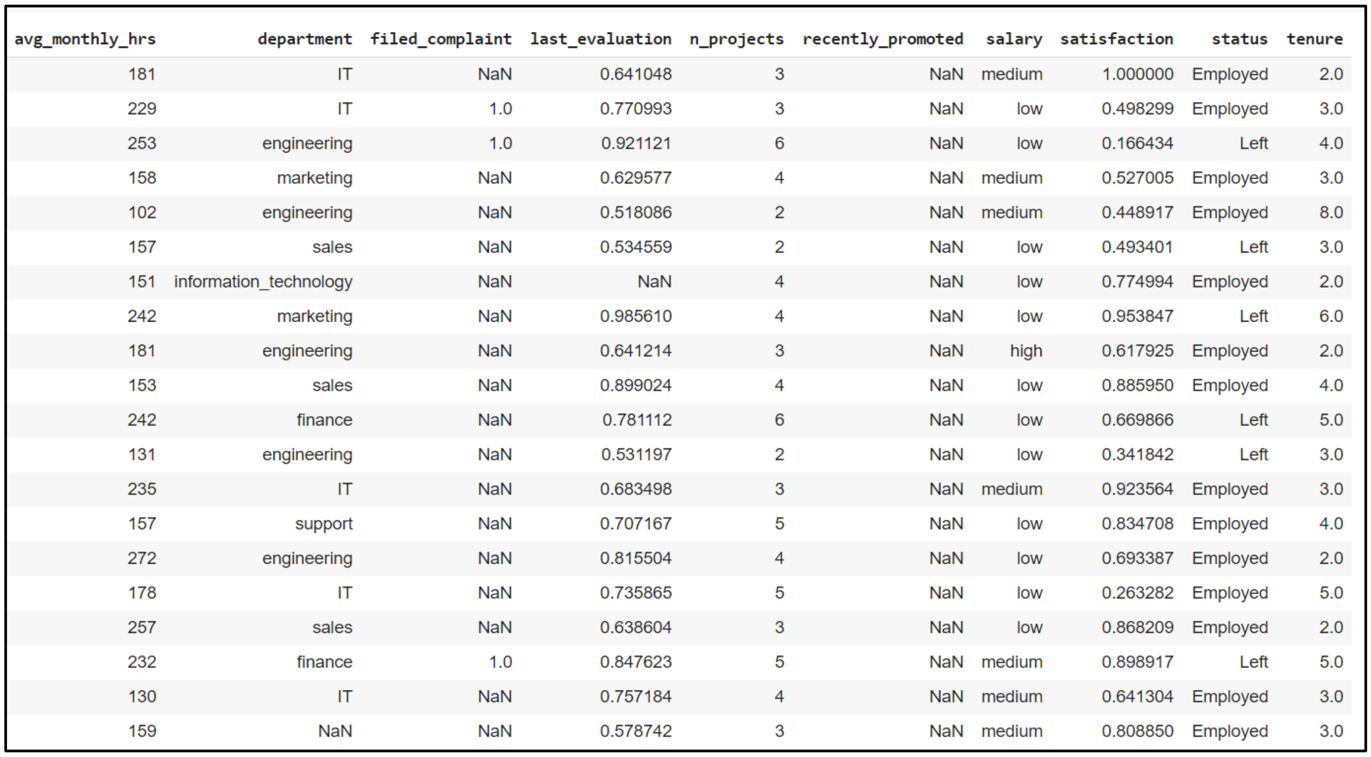
Description: 14249 rows, 10 columns

Data Dictionary:

1. **Status:** current employment status
2. **Department:** Department employee belongs to
3. **Salary:** Salary level relative to the rest of their department
4. **Tenure:** Number of years relative to the rest of their department
5. **Recently_Promoted: [categorical variable]** Promoted in the last 3 years
6. **N_Projects:** Number of projects the employee is stacked on
7. **Avg_Monthly_Hrs:** Average number of hours worked per month
8. **Satisfaction:** Score for employee’s satisfaction with the company
9. **Last_Evaluation:** Score for the most recent evaluation of employee
10. **Filed_Complaint: [categorical variable]** Formal complaint filed by employee in the last 3 years

Data Manipulation/Data Cleaning

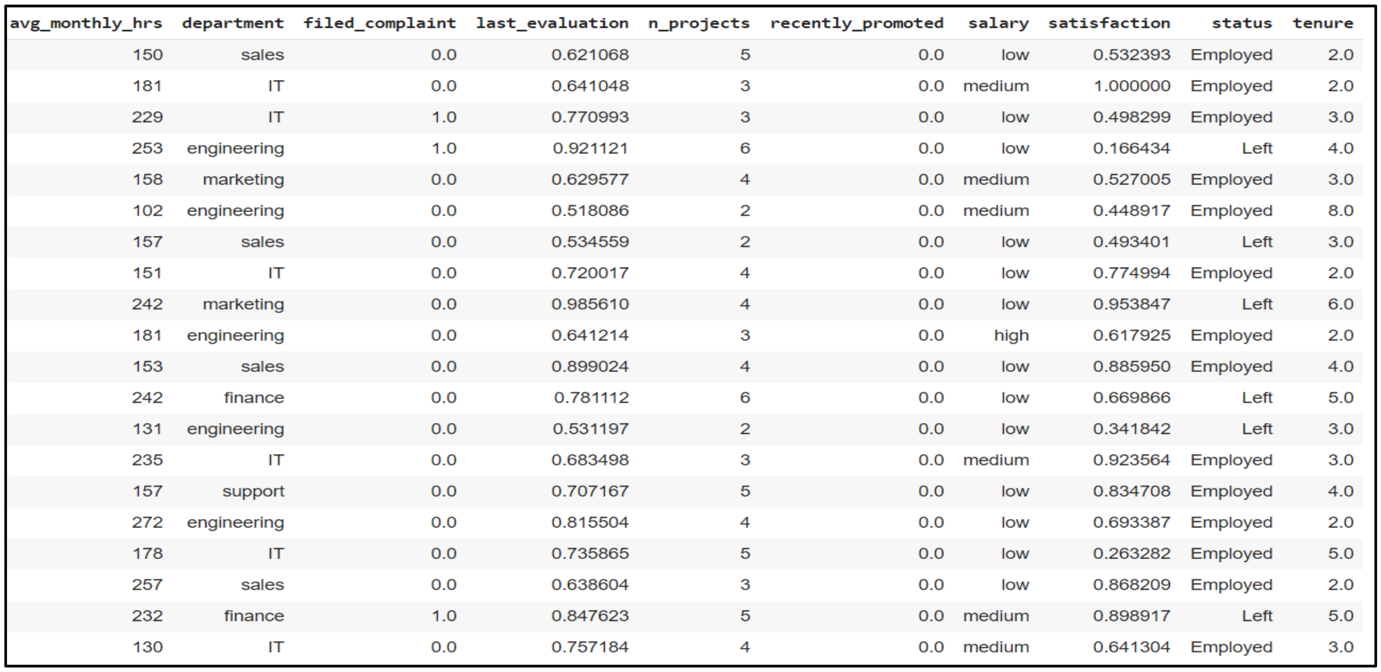
[Raw Data]



avg_monthly_hrs	department	filed_complaint	last_evaluation	n_projects	recently_promoted	salary	satisfaction	status	tenure
181	IT	NaN	0.641048	3	NaN	medium	1.000000	Employed	2.0
229	IT	1.0	0.770993	3	NaN	low	0.498299	Employed	3.0
253	engineering	1.0	0.921121	6	NaN	low	0.166434	Left	4.0
158	marketing	NaN	0.629577	4	NaN	medium	0.527005	Employed	3.0
102	engineering	NaN	0.518086	2	NaN	medium	0.448917	Employed	8.0
157	sales	NaN	0.534559	2	NaN	low	0.493401	Left	3.0
151	information_technology	NaN	NaN	4	NaN	low	0.774994	Employed	2.0
242	marketing	NaN	0.985610	4	NaN	low	0.953847	Left	6.0
181	engineering	NaN	0.641214	3	NaN	high	0.617925	Employed	2.0
153	sales	NaN	0.899024	4	NaN	low	0.885950	Employed	4.0
242	finance	NaN	0.781112	6	NaN	low	0.669866	Left	5.0
131	engineering	NaN	0.531197	2	NaN	low	0.341842	Left	3.0
235	IT	NaN	0.683498	3	NaN	medium	0.923564	Employed	3.0
157	support	NaN	0.707167	5	NaN	low	0.834708	Employed	4.0
272	engineering	NaN	0.815504	4	NaN	low	0.693387	Employed	2.0
178	IT	NaN	0.735865	5	NaN	low	0.263282	Employed	5.0
257	sales	NaN	0.638604	3	NaN	low	0.868209	Employed	2.0
232	finance	1.0	0.847623	5	NaN	medium	0.898917	Left	5.0
130	IT	NaN	0.757184	4	NaN	medium	0.641304	Employed	3.0
159	NaN	NaN	0.578742	3	NaN	medium	0.808850	Employed	3.0

1. *Imputed missing values in **filed_complaint** and **recently_promoted** with 0 to make them discrete and numerical for analysis*
2. *Dropped rows with missing values for **department**, **tenure**, and **satisfaction***
3. *Changed **department** from 'information_technology' to 'IT', redundant name for same department, 209 rows*
4. *Imputed missing values for **last_evaluation** by the respective department's mean value for last_evaluation, 1284 rows. Imputation was done because dropping these rows would result in significant data loss.*

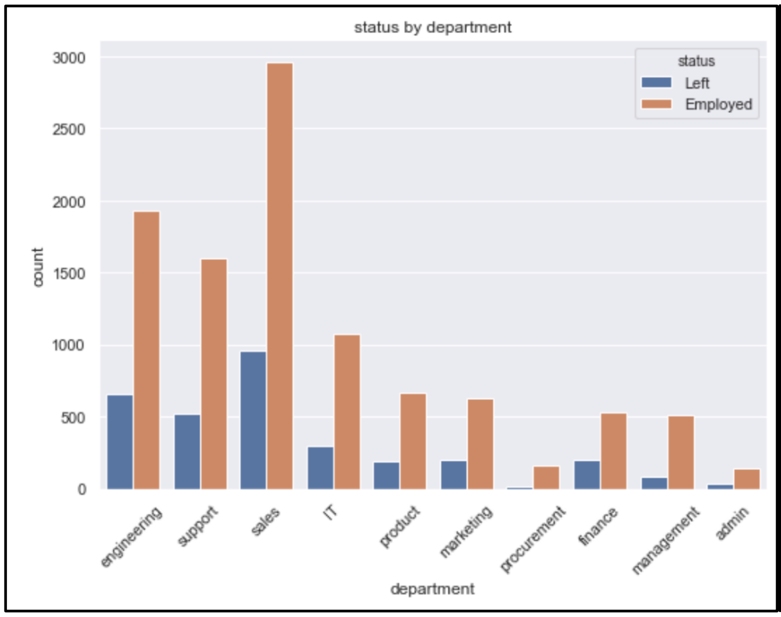
[Cleaned Data]

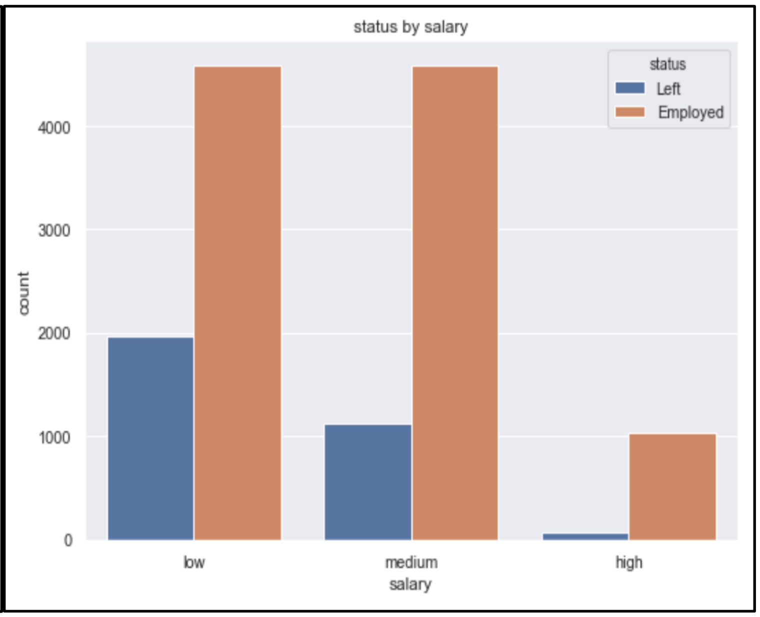


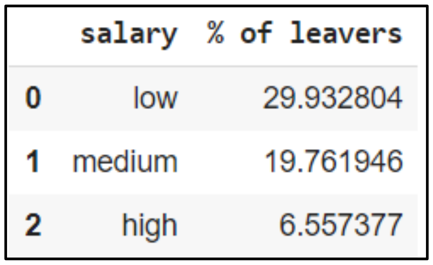
avg_monthly_hrs	department	filed_complaint	last_evaluation	n_projects	recently_promoted	salary	satisfaction	status	tenure
150	sales	0.0	0.621068	5	0.0	low	0.532393	Employed	2.0
181	IT	0.0	0.641048	3	0.0	medium	1.000000	Employed	2.0
229	IT	1.0	0.770993	3	0.0	low	0.498299	Employed	3.0
253	engineering	1.0	0.921121	6	0.0	low	0.166434	Left	4.0
158	marketing	0.0	0.629577	4	0.0	medium	0.527005	Employed	3.0
102	engineering	0.0	0.518086	2	0.0	medium	0.448917	Employed	8.0
157	sales	0.0	0.534559	2	0.0	low	0.493401	Left	3.0
151	IT	0.0	0.720017	4	0.0	low	0.774994	Employed	2.0
242	marketing	0.0	0.985610	4	0.0	low	0.953847	Left	6.0
181	engineering	0.0	0.641214	3	0.0	high	0.617925	Employed	2.0
153	sales	0.0	0.899024	4	0.0	low	0.885950	Employed	4.0
242	finance	0.0	0.781112	6	0.0	low	0.669866	Left	5.0
131	engineering	0.0	0.531197	2	0.0	low	0.341842	Left	3.0
235	IT	0.0	0.683498	3	0.0	medium	0.923564	Employed	3.0
157	support	0.0	0.707167	5	0.0	low	0.834708	Employed	4.0
272	engineering	0.0	0.815504	4	0.0	low	0.693387	Employed	2.0
178	IT	0.0	0.735865	5	0.0	low	0.263282	Employed	5.0
257	sales	0.0	0.638604	3	0.0	low	0.868209	Employed	2.0
232	finance	1.0	0.847623	5	0.0	medium	0.898917	Left	5.0
130	IT	0.0	0.757184	4	0.0	medium	0.641304	Employed	3.0

Exploratory Data Analysis

1. Exploring the **status** of employees by department and **salary**



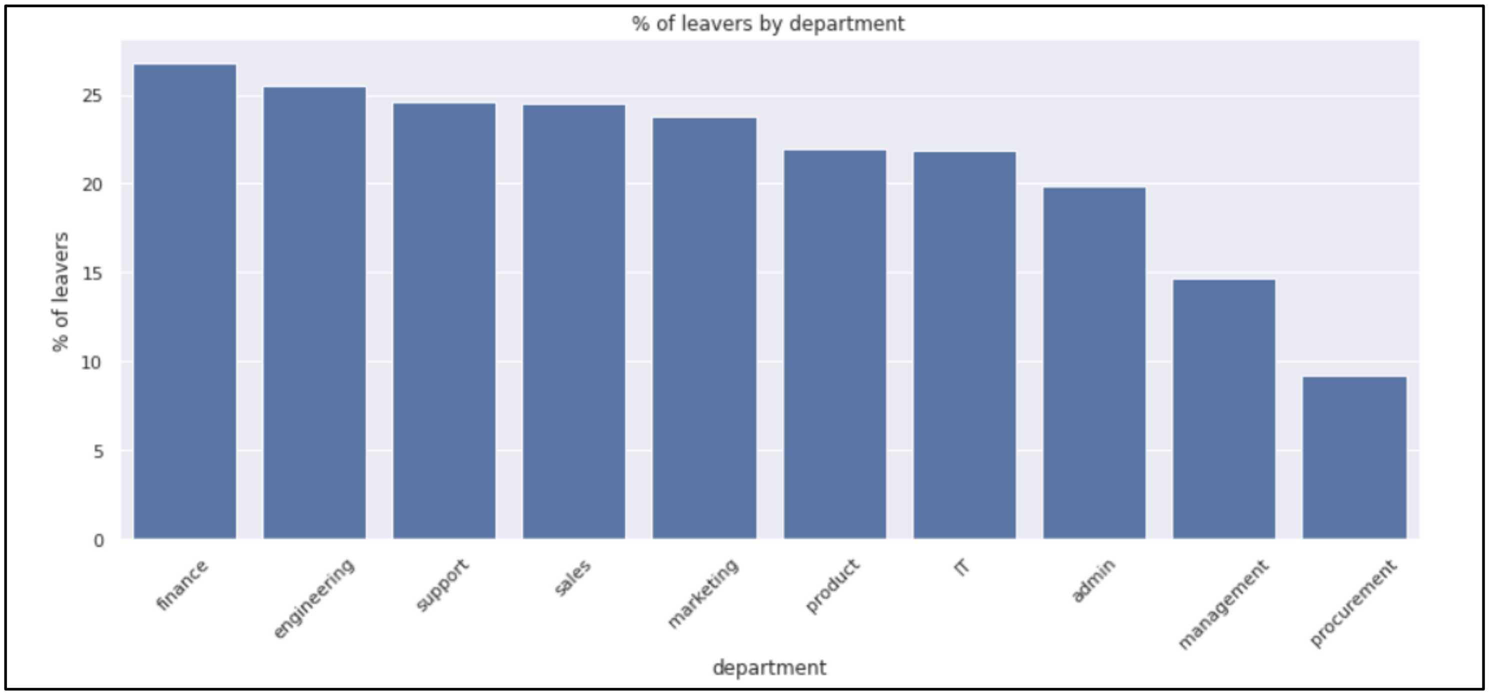




	salary	% of leavers
0	low	29.932804
1	medium	19.761946
2	high	6.557377

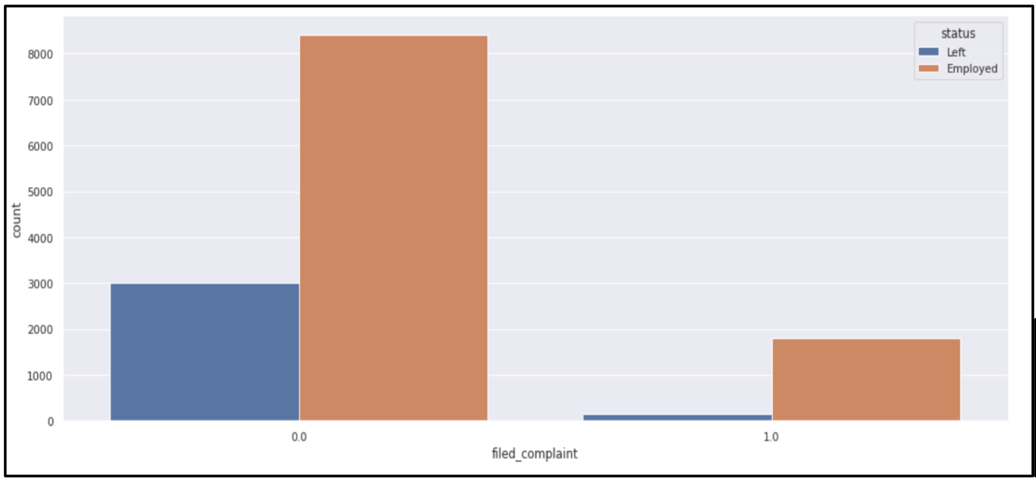
From the "status by department" bar chart, we observed that the 'sales' department had the highest number of employees that left. As can be analyzed intuitively, we see that people with relatively low salary left the company which made up about 30% of employees.

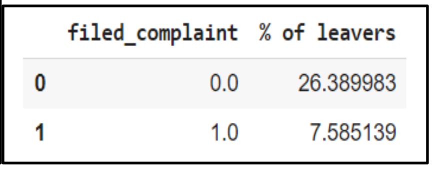
2. Exploring the percentage of employees who left **by department**



It is necessary that the company take a look into the policies and work culture for the finance department because it has the highest percentage of employees who left. It might also help to see what the procurement department is doing to prevent employees from leaving.

3. Exploring by **Filed_Complaint**

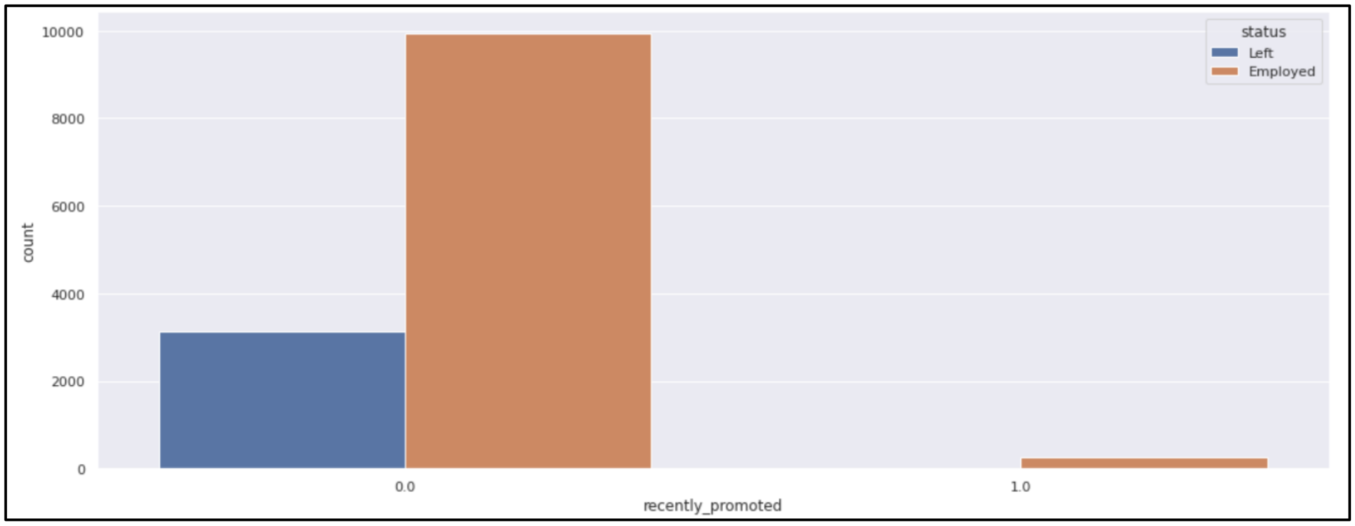


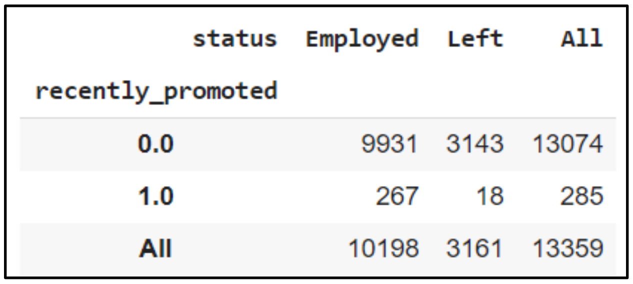


	filed_complaint	% of leavers
0	0.0	26.389983
1	1.0	7.585139

Here, we observe that employees who have filed a formal complaint in the last 3 years have more retention than the people that did not. It could be that complaint with the company or coworkers factors less in people's decision to leave the company

4. Exploring by **Recently_Promoted**

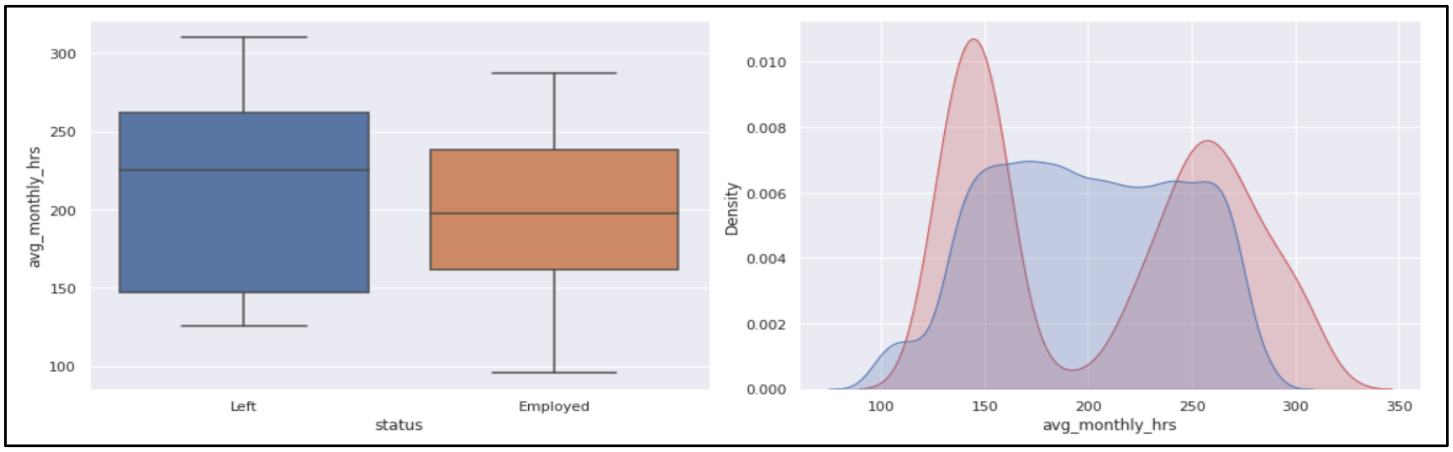




status	Employed	Left	All
recently_promoted			
0.0	9931	3143	13074
1.0	267	18	285
All	10198	3161	13359

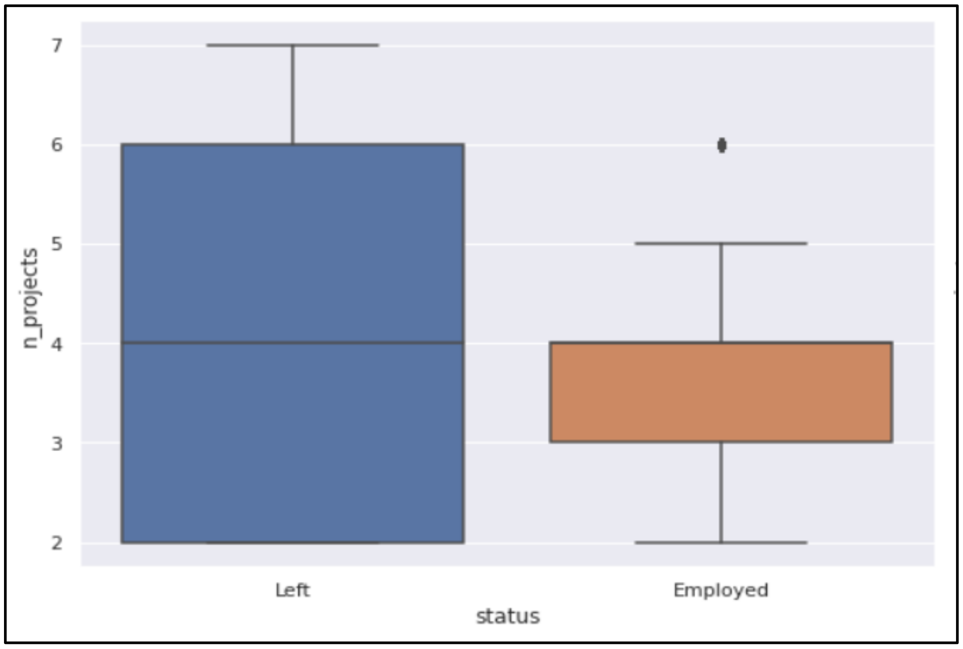
As expected, employees who have been recently promoted in the last 3 years have a very high retention rate. The total number of employees not recently promoted was 13,074 and 3143 of them left the company. That is 24%! It is safe to say, promotions play a critical role in retaining employees.

5. Exploring by **Avg_Monthly_Hrs**



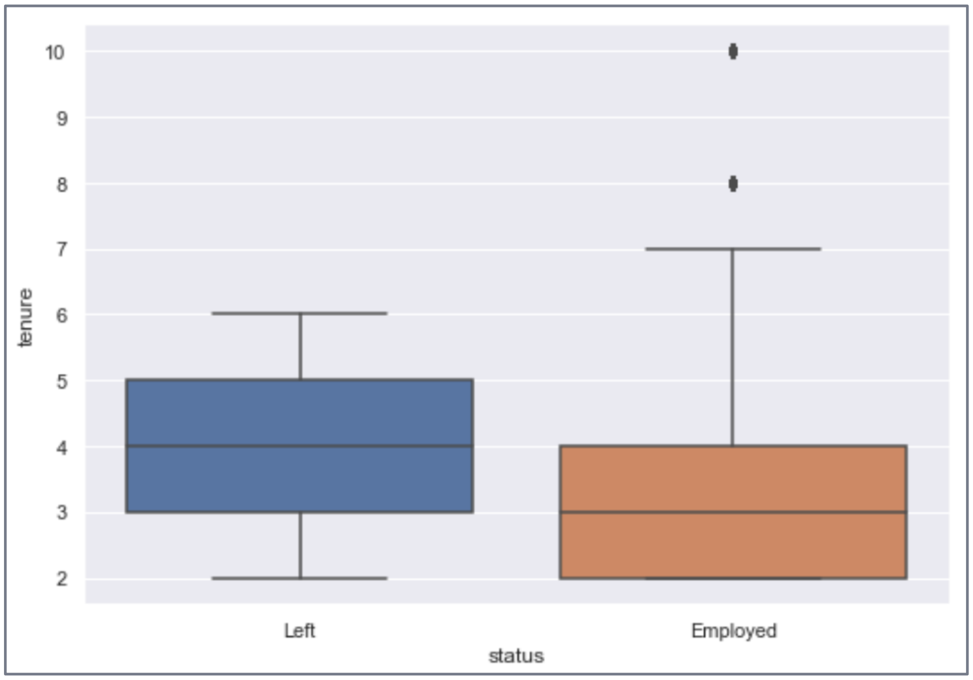
We observe that the employees who left, on an average worked more number of hours per month than the employees still with the company. The kdeplot (right) explains the distribution further--people that left either worked around 150 hours or around 260 hours per month. We can see this distribution is quite polarized when compared to the hours worked among employees that are still with the company.

6. Exploring by **N_Projects**



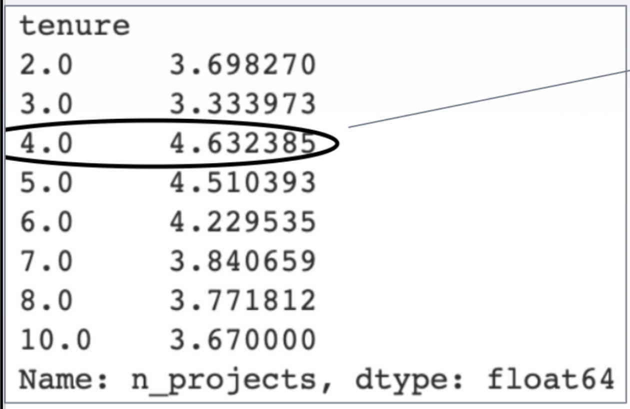
We observe that the employees who left had on an average a greater number of projects than people who are still employed. This correlates with the "avg_monthly_hrs" chart (number 5). The employees with more projects probably were the ones who put in the extra hours of work, eventually resulting in them leaving the company.

7. Exploring by **Tenure**

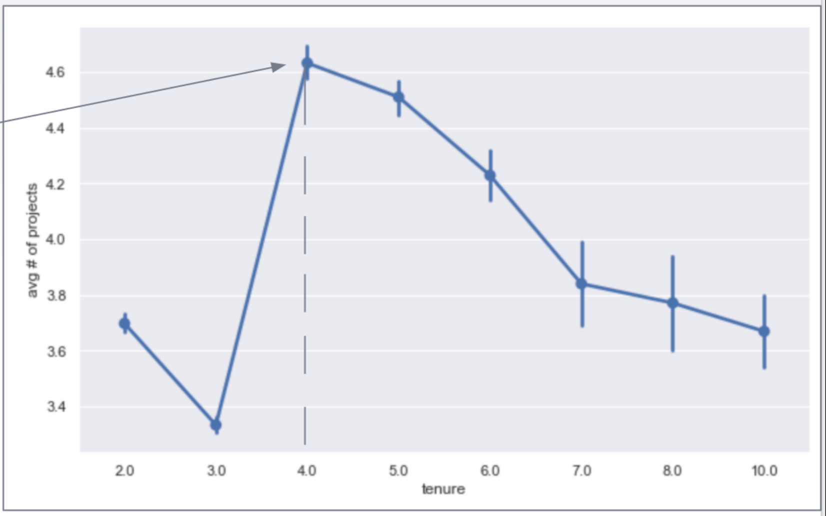


This histogram shows that the employees who left stayed with the company for about 4 years before leaving.

8. Exploring by **Tenure** and **N_Projects**

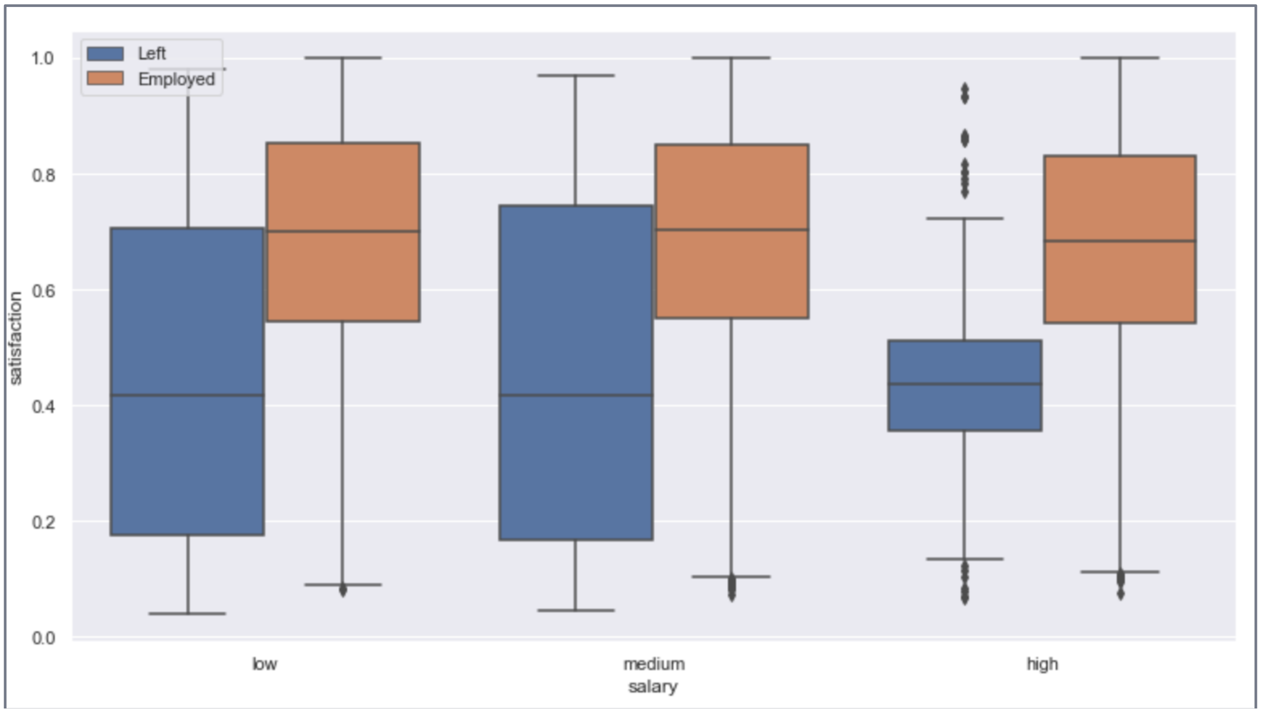


```
tenure
2.0      3.698270
3.0      3.333973
4.0      4.632385
5.0      4.510393
6.0      4.229535
7.0      3.840659
8.0      3.771812
10.0     3.670000
Name: n_projects, dtype: float64
```



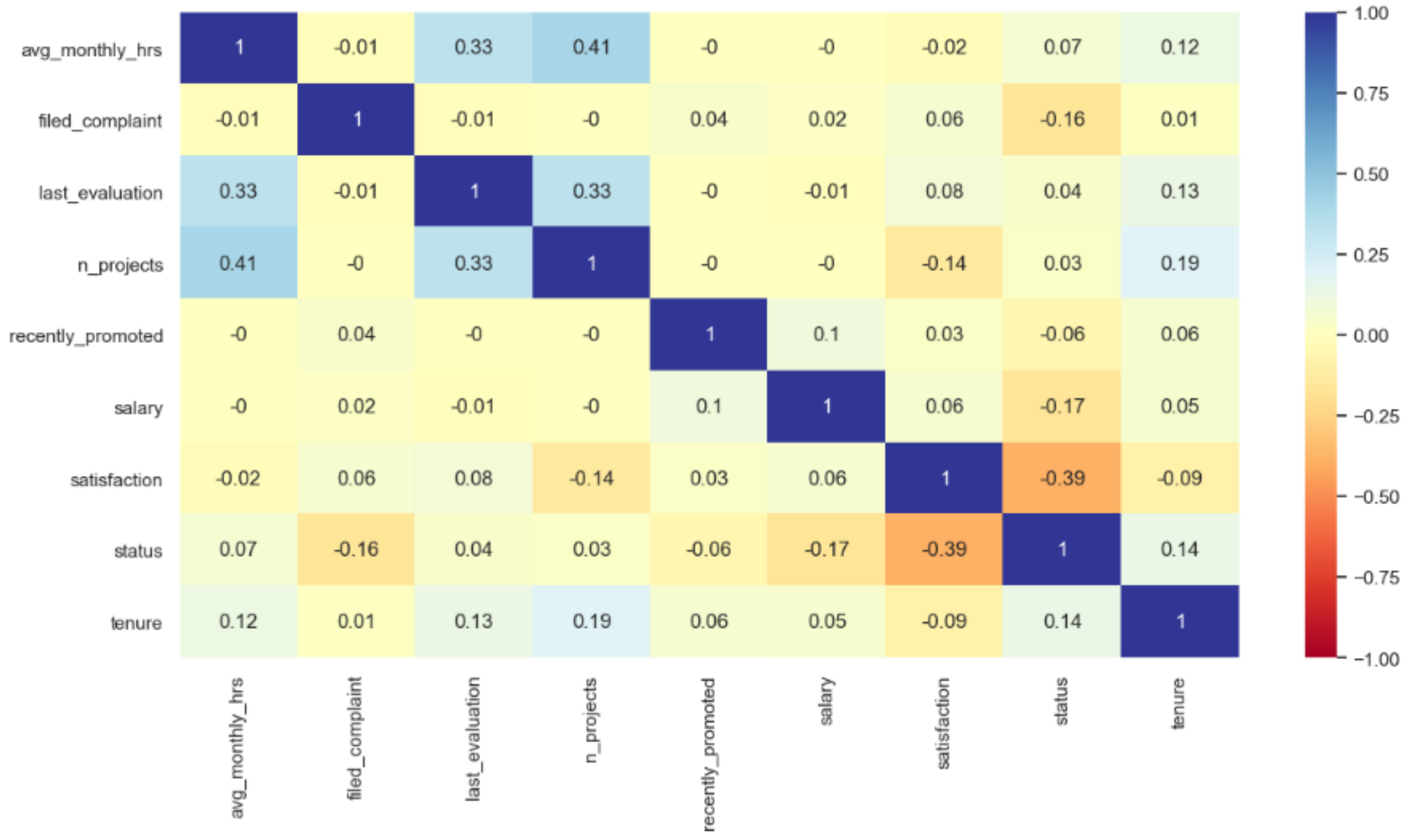
We can make a connection that around the 4 year mark employees have the most projects. If we look back at number 7, employees leave around the 4 year mark. There seems to be a correlation with the number of projects given, number of hours worked and tenure.

9. Exploring by **Satisfaction** and **Salary**



There is a relation between salary and satisfaction. There are a lot more employees who left when salary was low compared to when salary was high. The median satisfaction level was overall lower for those employees who left regardless of the salary level. Satisfaction level and salary level was generally higher for the employees who stayed.

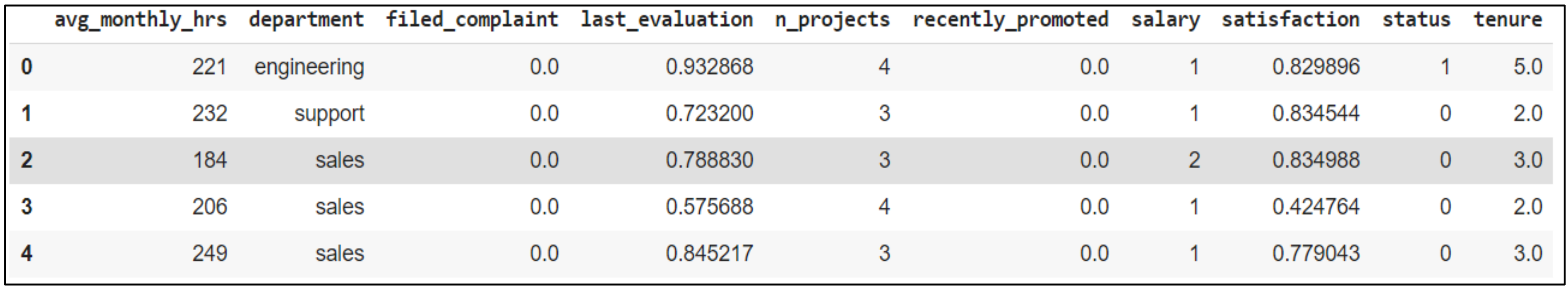
10. Bivariate analysis



As we can see, the target has slightly negative correlation with satisfaction. Also**, n_projects** and **last evaluation** are positively correlated with **avg_monthly** hours. Please note that the target variable was mapped to 0 (did not leave) and 1 (left).

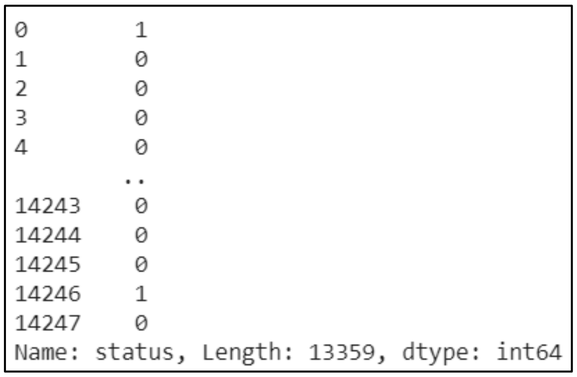
Data Preprocessing

X



	avg_monthly_hrs	department	filed_complaint	last_evaluation	n_projects	recently_promoted	salary	satisfaction	status	tenure
0	221	engineering	0.0	0.932868	4	0.0	1	0.829896	1	5.0
1	232	support	0.0	0.723200	3	0.0	1	0.834544	0	2.0
2	184	sales	0.0	0.788830	3	0.0	2	0.834988	0	3.0
3	206	sales	0.0	0.575688	4	0.0	1	0.424764	0	2.0
4	249	sales	0.0	0.845217	3	0.0	1	0.779043	0	3.0

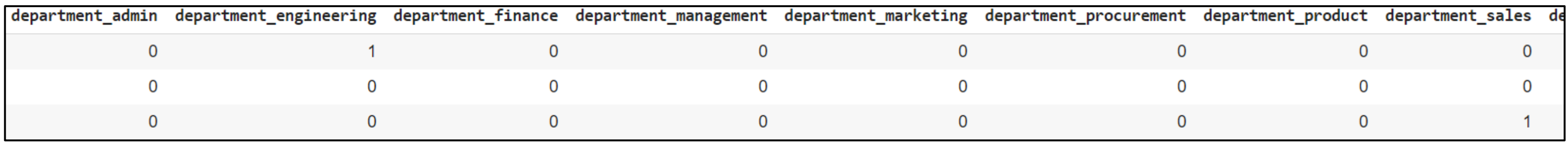
Y (variable of interest)



```
0        1
1        0
2        0
3        0
4        0
        ..
14243    0
14244    0
14245    0
14246    1
14247    0
Name: status, Length: 13359, dtype: int64
```

As is with typical partitioning of data before feeding it to models, we partitioned our data into training and test. Our data consists of numerical as well as categorical variables; to make sure we provide interpretable data to models for analysis we standardised X(to get rid of scaling issues) and created dummy variables since these models do not work on categorical variables.

Dummy variables for department (**X**)



department_admin	department_engineering	department_finance	department_management	department_marketing	department_procurement	department_product	department_sales	de
0	1	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	1	

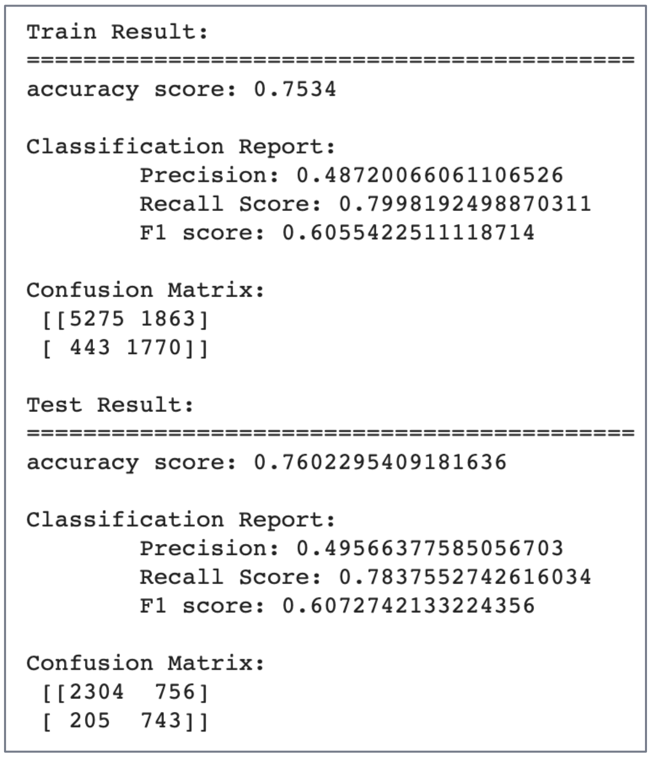
Data Analysis - Machine Learning

After cleaning and exploring the dataset, we applied machine learning models to predict which employees would leave based on given employee data. Since our target variable (status) is binary, (if an employee left or is still mployed) this is a classification problem. We used classification models, such as Logistic Regression, Support Vector achine, Decision Tree, Random Forest, and Gradient Boosting to predict if an employee would leave the company or tay. Since our target variable is not balanced, we decided that another metric besides accuracy was needed to evaluate our model. We could use precision if the cost of false positive is high or recall if the cost of false negative is high. We ecided to focus on recall as we want to be able to predict all of workers who are about to quit. We used GridsearchCV and RandomsearchCV(for Randomforest) to tune the hyperparameters of the different algorithms used in the analysis. Since the dataset is imbalanced, we used AUC score to evaluate different models.

Logistic Regression

The first model we ran was the Logistic Regression model. Before tuning the parameters, our Logistic egression model's test accuracy was 0.79 and F1 score was 0.45. After running our initial Logistic Regression Model, e tuned some of the hyperparameters, such as C, regularization penalty, solver, and class weight.

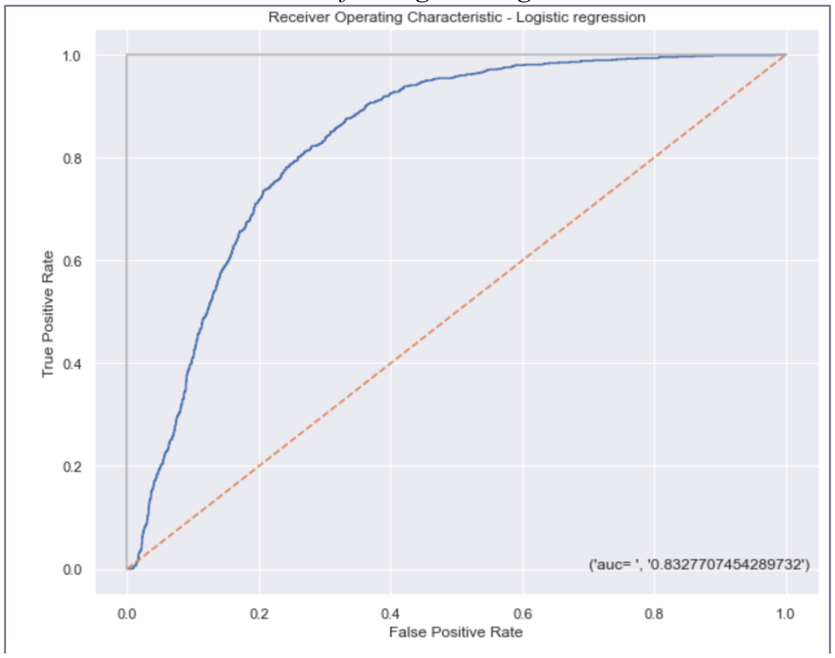
Train/Test results after Hyperparameter tuning



```
Train Result:
===========================================
accuracy score: 0.7534

Classification Report:
        Precision: 0.48720066061106526
        Recall Score: 0.7998192498870311
        F1 score: 0.6055422511118714

Confusion Matrix:
 [[5275 1863]
 [ 443 1770]]

Test Result:
===========================================
accuracy score: 0.7602295409181636

Classification Report:
        Precision: 0.49566377585056703
        Recall Score: 0.7837552742616034
        F1 score: 0.6072742133224356

Confusion Matrix:
 [[2304  756]
 [ 205  743]]
```

Next, we plotted the ROC curve of our model, which plots the True Positive Rate on the y-axis and False Positive Rate on the x-axis. The area under the curve, which measures the ability of a classifier to distinguish between classes, is 0.83.

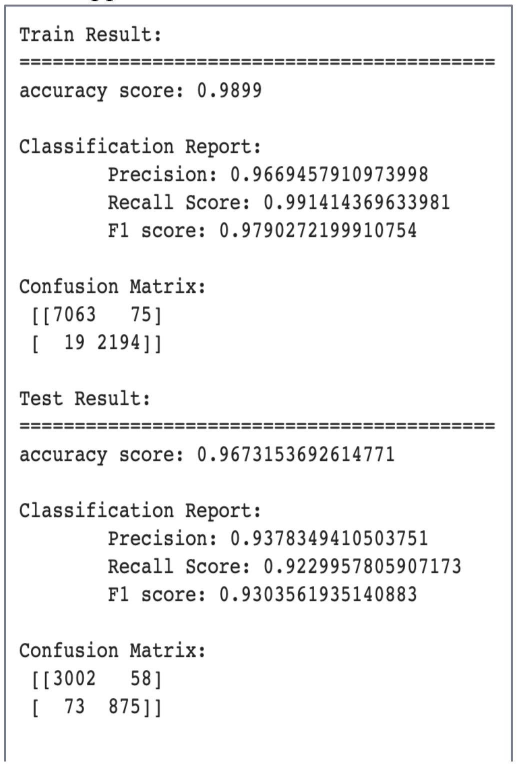
Roc Curve for Logistic Regression



Support Vector Machine

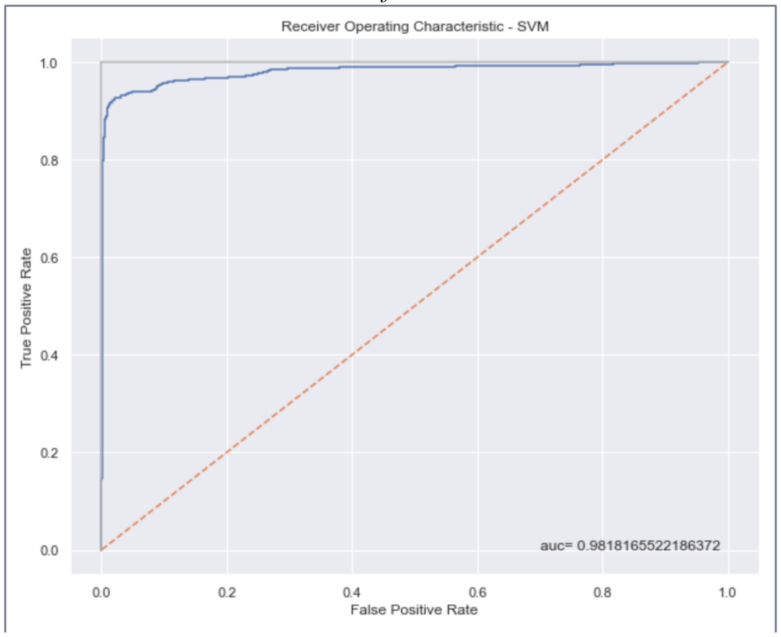
The next model we built was the Support Vector Machine. The hyperparameters we tuned for the Support Vector Machine were C, gamma, kernel, and class weight. Our best SVM model consisted of C=1,gamma=1, kernel=rbf, and class_weight=balanced.

Support Vector Machine Results



```
Train Result:
============================================
accuracy score: 0.9899

Classification Report:
        Precision: 0.9669457910973998
        Recall Score: 0.991414369633981
        F1 score: 0.9790272199910754

Confusion Matrix:
 [[7063   75]
 [  19 2194]]

Test Result:
============================================
accuracy score: 0.9673153692614771

Classification Report:
        Precision: 0.9378349410503751
        Recall Score: 0.9229957805907173
        F1 score: 0.9303561935140883

Confusion Matrix:
 [[3002   58]
 [  73  875]]
```

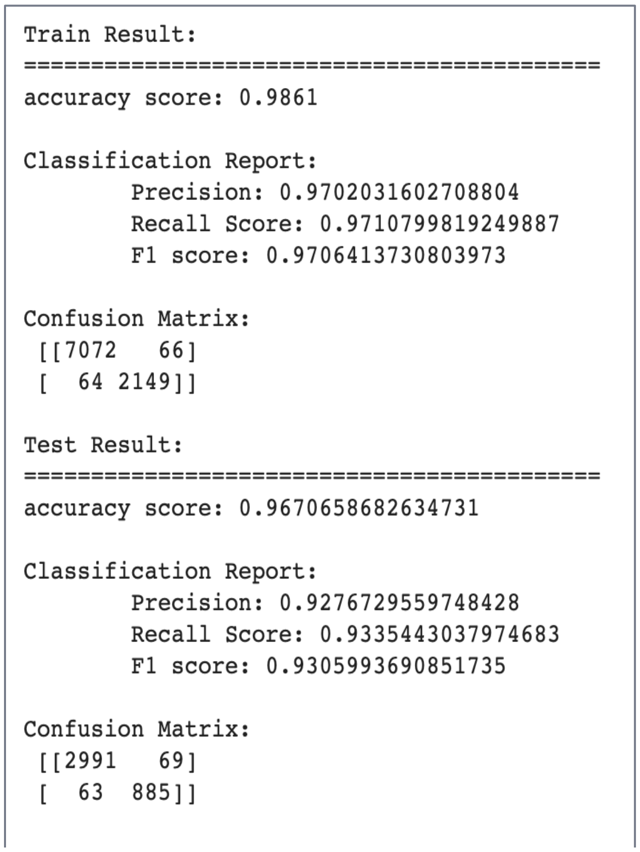
Roc Curve of SVM model



Decision Tree

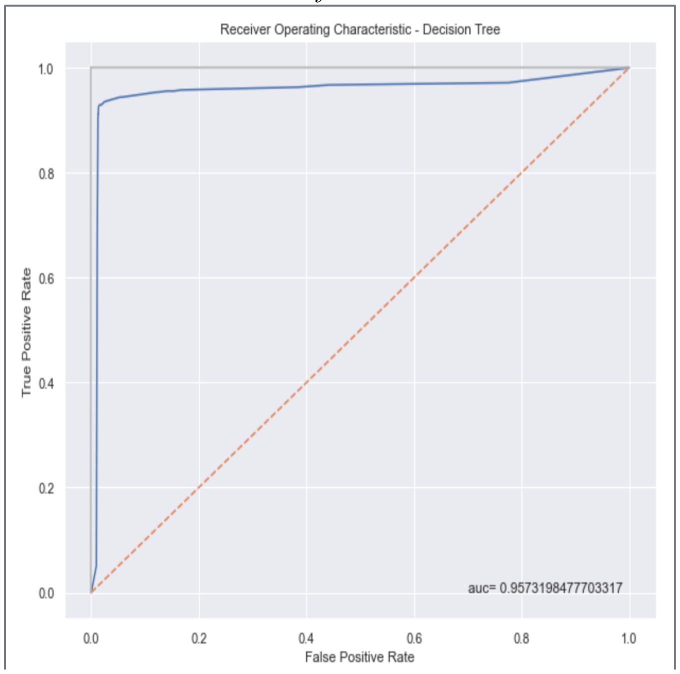
The next model we built was the Decision Tree. The hyperparameters we tuned for the Decision Tree were criterion, splitter, max_depth, min_samples_split, min_samples_leaf, class_weight. Our best Decision Tree model consisted of criterion=gini, max_depth=10, min_samples_split=2, min_samples_leaf=1.

Decision Tree Results



```
Train Result:
===========================================
accuracy score: 0.9861

Classification Report:
        Precision: 0.9702031602708804
        Recall Score: 0.9710799819249887
        F1 score: 0.9706413730803973

Confusion Matrix:
 [[7072   66]
 [  64 2149]]

Test Result:
===========================================
accuracy score: 0.9670658682634731

Classification Report:
        Precision: 0.9276729559748428
        Recall Score: 0.9335443037974683
        F1 score: 0.9305993690851735

Confusion Matrix:
 [[2991   69]
 [  63  885]]
```

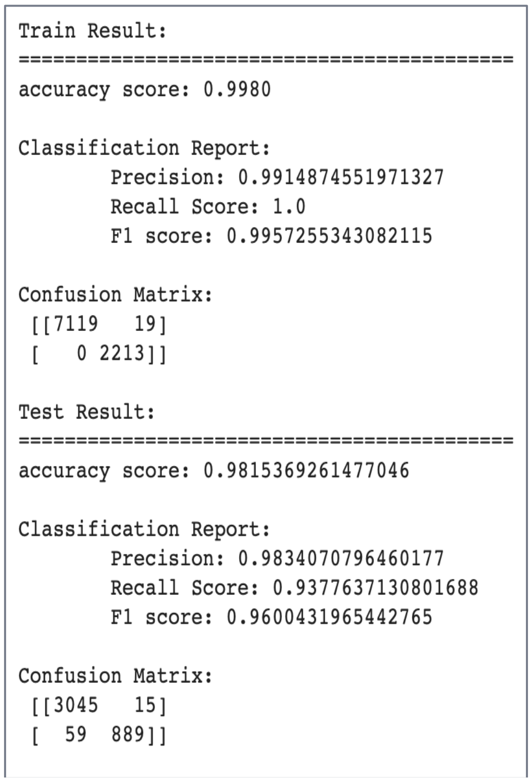
Roc Curve of Decision Tree



Random Forest

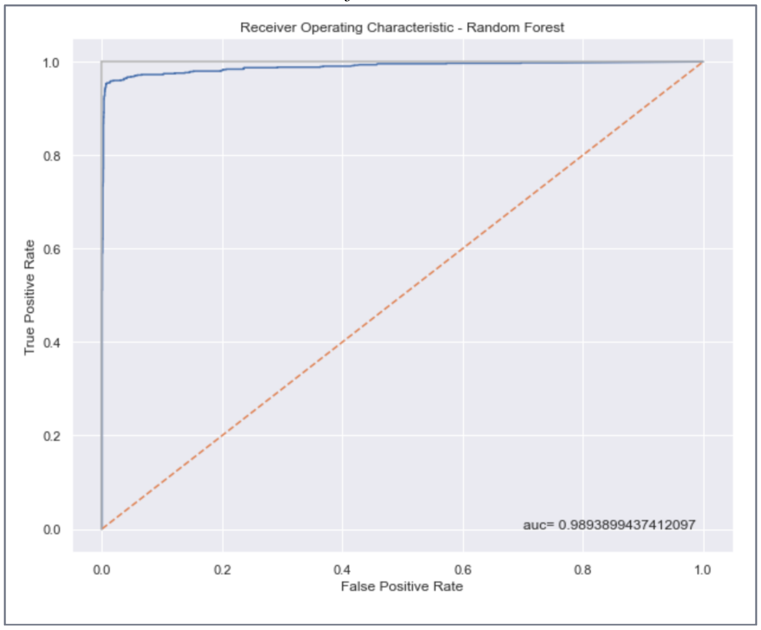
The next model we built was the Random Forest. The hyperparameters we tuned for the Random Forest were C, gamma, kernel, and class weight. Our best SVM model consisted of C=1,gamma=1, kernel=rbf, and class_weight=balanced.

Random Forest Results



```
Train Result:
============================================
accuracy score: 0.9980

Classification Report:
        Precision: 0.9914874551971327
        Recall Score: 1.0
        F1 score: 0.9957255343082115

Confusion Matrix:
 [[7119   19]
 [   0 2213]]

Test Result:
============================================
accuracy score: 0.9815369261477046

Classification Report:
        Precision: 0.9834070796460177
        Recall Score: 0.9377637130801688
        F1 score: 0.9600431965442765

Confusion Matrix:
 [[3045   15]
 [  59  889]]
```

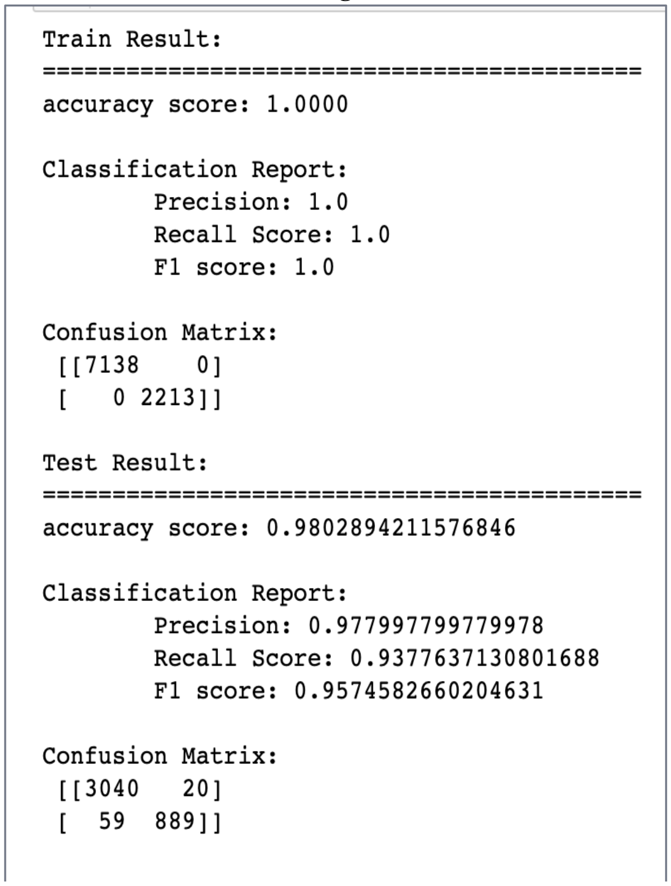
Roc curve of Random Forest



Gradient Boosting

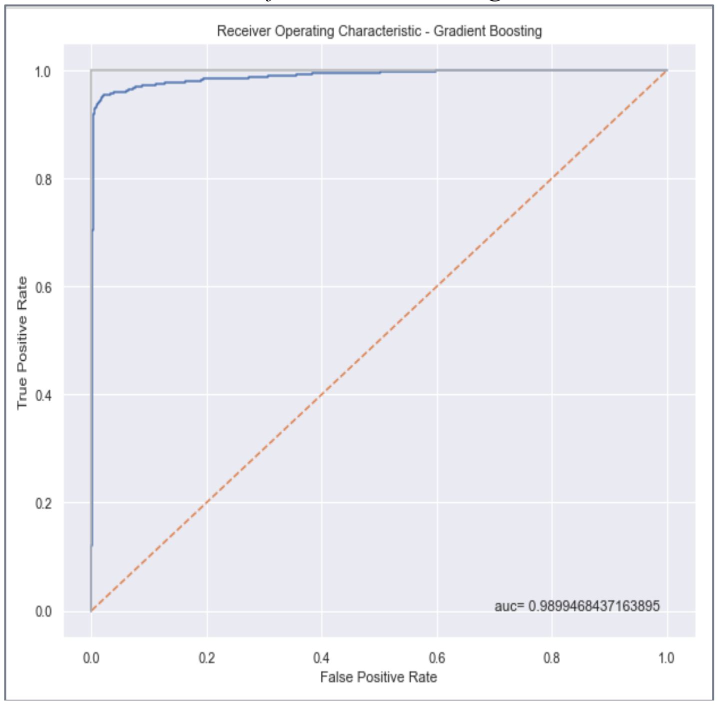
The next model we built was the Gradient Boosting. The hyperparameters we tuned for the Gradient Boosting were learning_rate, max_depth, min_samples_split, and class n_estimators. Our best Gradient Boosting model consisted of learning_rate=0.1, max_depth=8, min_samples_split=6, and n_estimator=300.

Gradient Boosting Model Results

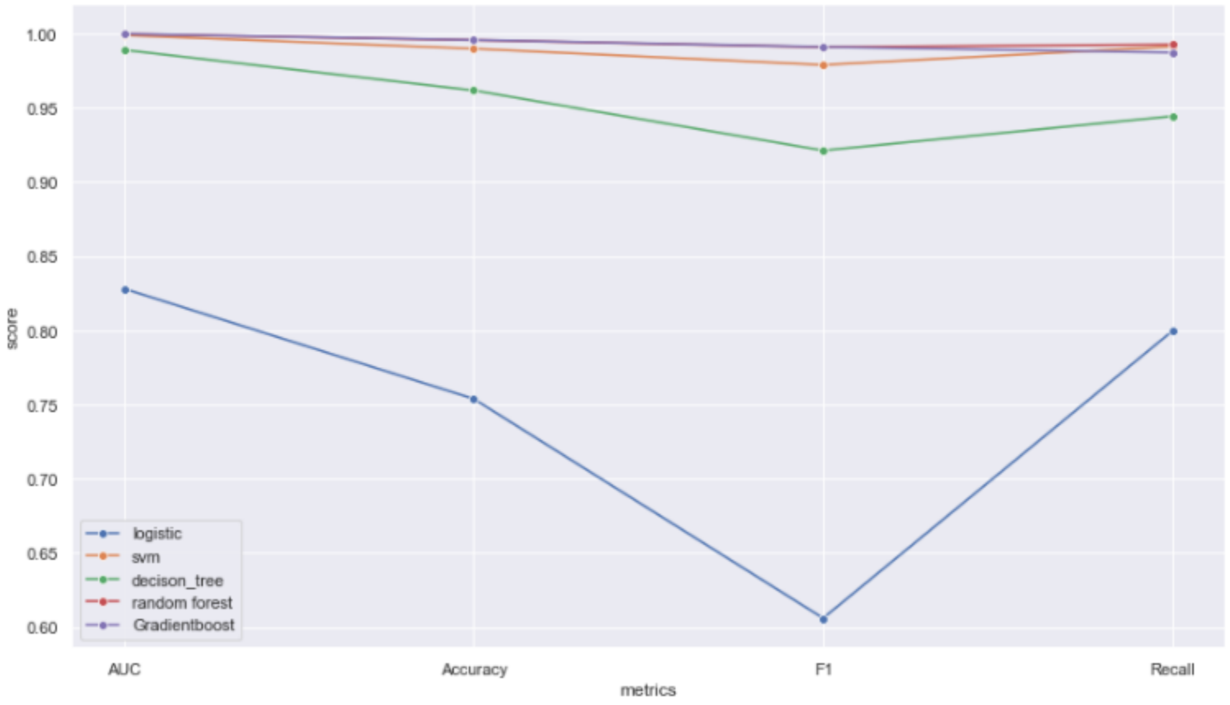


```
Train Result:
============================================
accuracy score: 1.0000

Classification Report:
        Precision: 1.0
        Recall Score: 1.0
        F1 score: 1.0

Confusion Matrix:
 [[7138    0]
 [   0 2213]]

Test Result:
============================================
accuracy score: 0.9802894211576846

Classification Report:
        Precision: 0.977997799779978
        Recall Score: 0.9377637130801688
        F1 score: 0.9574582660204631

Confusion Matrix:
 [[3040   20]
 [  59  889]]
```

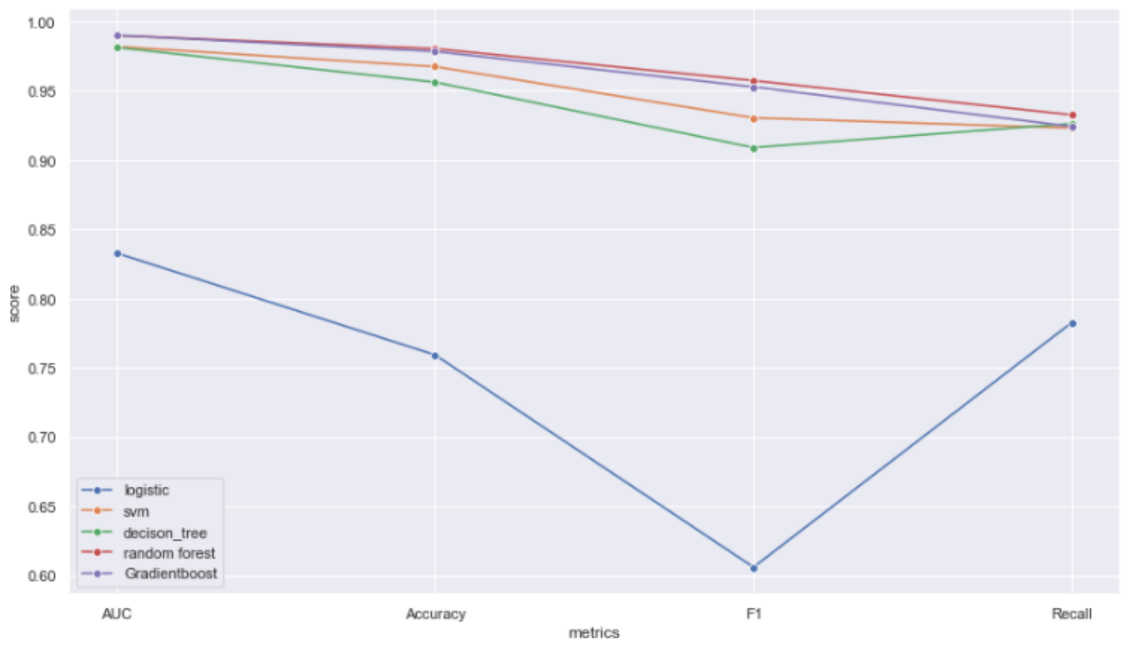
Roc curve of Gradient Boosting model



Models evaluation (train dataset)



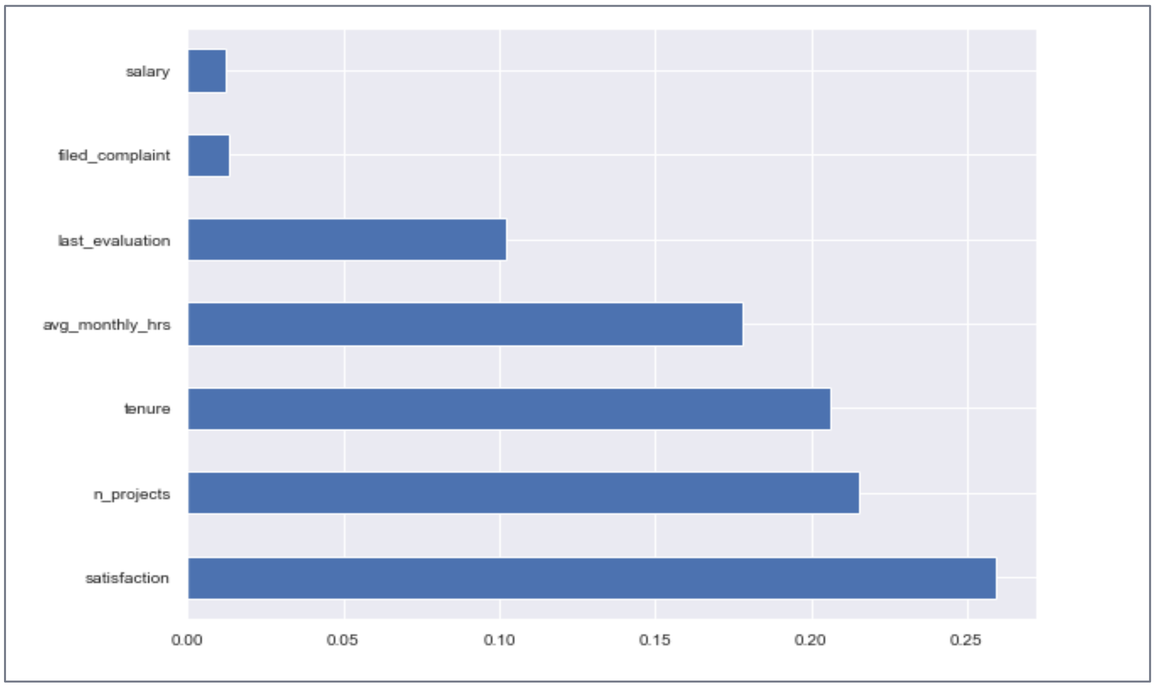
Model evaluation (test dataset)



The above two graphs represent the evaluation of different models used on both test and train datasets. The best model to analyze the data on employee retention is the Random Forest with the hyperparameters: n_estimators: 100, min_samples_split:6, min_samples_leaf: 1, max_features:'sqrt', max_depth:20, criterion:'entropy' . As shown above, the Random Forest model scored the highest on AUC, accuracy, F1 and Recall. In addition, there is not a big difference on how the Random Forest model performs on the training set than on the test set, which means the model is not likely overfitting.

Feature Importance

Looking at the feature importance graph, we can suggest the company to focus on increasing employee satisfaction in order to retain more employees. The next feature importance is the number of projects. As data analysts, we can suggest that there may be a correlation between the number of projects and employee satisfaction.


salary
filed_complaint
last_evaluation
avg_monthly_hrs
tenure
n_projects
satisfaction