Assignment 6:

Final Report

Prepared by: Jacqueline James

2022-04-21

Problem Statement

In 2021, Acorn released its strategic vision for 2026. The plan aims to leverage data driven insights to optimize business performance in an increasingly competitive environment.

Along with 5 other objectives, attrition has been identified as a key priority area. Thus, the current report will focus on employee attrition.

Attrition may refer either to long-term vacancies or to positions that have been eliminated. Reasons for attrition may vary but can include low unemployment levels, toxic workplace, workplace restructuring, among other factors.

Not only does attrition carry with it directly quantifiable costs (*i.e.* resources needed to offboard and onboard employees), but it is also associated with 'soft' costs such as diminished employee moral, poor reputation within the industry, compromised future performance, loss of human capital etc. In fact, a well documented negative relationship exists between organizational performance and turnover, as well as between labour productivity and turnover (Hancok *et.al., 2013,* (Winnea, Marescauxb, Selsc, Beverend, & Vanormelingen, 2019).

Ultimately, Acorn aims to reduce its turnover rate to the industry average within 1 year. To do this Acorn would like to predict which employees are at risk of leaving and to understand the drivers of attrition.

In assessing the success of this project, the average length of employment, voluntary/involuntary turnover rates and the new hire retention rate will be key metrics of success.

Key Stakeholders

HR leadership team, HR business partners, business leaders, and other analytics teams are the main stakeholders for this project. Their role in this project, estimated influence, importance, needs and concerns are outlined in figure 1.1 below.

Figure 1.1: Stakeholder Analysis

Key Stakeholder	Role	Estimated Project Influence	Estimated Project Importance	Needs	Concerns
HR leadership team	Advocate			Regular progress updates	Privacy concerns surrounding data collection and usage
HR business partners	Educate			Access to results, including personal identifiers Method to communicate results clearly	Privacy concerns surrounding data collection and usage
Business leaders	Advocate			Links to organizational performance financial measures of success	Financial concerns related to project viability
Other analytics Teams	Partner			Access to data without sensitive information Opportunities to link with other analytics projects	Compatibility with other analytics projects

Section 2: Research, Methodology and Ethics

As the knowledge, skills and abilities (*i.e.*, human capital) of an employee increase, so to does the cost of attrition (Hancock, Bosco, McDaniel, & Pierce, 2013; Singh, *et al.,* 2012). Therefore, the current analysis will focus on employees who have had a significant amount of firm specific human capital invested in them.

Firm specific human capital was calculated as the average between an individuals' job level, number of training sessions attended over the last year, their performance rating, as well as the number of years spent at Acorn. Human capital scores falling in the 50th percentile and above were included in the analysis, as this was deemed a significant investment.

2.1 Methodology

Several analytic models will be used in the current analysis (as opposed to relying purely on descriptive analytics). The advantage of predictive modeling is that it illustrates how the various drivers of attrition interact with one another to influence an employee's decision to leave the organization in the future. Put simply, predictive modeling provides an idea of cause and effect.

Dataset: IBM's Employee Attrition dataset will be used for the current analysis. The data was created in 2019 by IBM. It was retrieved in February 2022. This dataset includes 1470 entries with 35 features relating to demographic information, financial factors, job details, tenure, satisfaction ratings and more (please see Appendix A for more details about the data).

Preprocessing: There are several features that will be removed from the analysis as they do not provide any value. 'Employee count', 'Over 18' and 'Standard Hours' are all constant values. This lack of variance will not benefit the model. Further, employee number is a unique identifier for each employee and will similarly not benefit the model.

2.1.1 Exploratory Data Analysis

Feature creation: In addition to the 35 features included in the dataset, new features were also created. These were; training compare and income groups. Both of these variables aim to quantify an employee's perception of performance in relation to their peer groups. This was motivated by Singh, *et al. (2012)* who suggested that those who are underperforming in relation to their peers would be more likely to leave the organization as compared to those who are overperforming in relation to their peers.

'Training Compare' focuses on the amount of training an employee has received as compared to that of their peers. In this case, peer group is defined by one's department. Thus, the feature is calculated by subtracting the amount of training times attended by an employee within the past year from the department average.

Income groups measures an employee's income as compared to their peer groups. In this instance, a peer group is defined by one's department and job level. Thus, the feature is calculated by subtracting the individual's income from the average income of those in the same department and job level.

Features Condensed: Several categorical variables were condensed into fewer categories; this was done to simplify the final model and to address rare levels. 9 job roles, were condensed into 3, 5 Job levels were condensed into 3, and 6 Education Fields were condensed into 4 (please see Appendix B for more information as to how these variables were created).

Transformations: Monthly income was log transformed. This was done to account for the diminishing returns to income. It is likely that an increase in income would convince an employee to stay with Acorn but only up to a certain point. After that, an additional unit increase of income would not be enough to encourage an employee to stay. Log transformations would account for these diminishing returns.

Past Literature: Driven by the rise of people analytics, past work on attrition analysis is readily available. After a scan of the literature,

several drivers of attrition have been identified. These features can be grouped in to the 5 broad categories shown below;

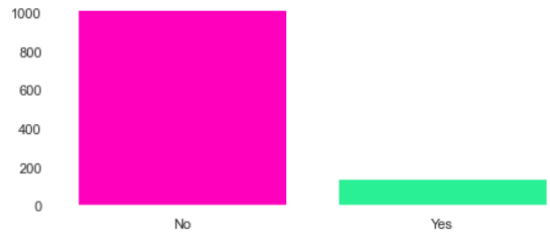
1. *Financial Factors:* Compensation, salary compared to one's peer group, the length of time since last promotion and the length of time since last salary change (Vimoli & Modi, 2021; Singh, et al.).
2. *Work Experience*: Number of companies worked, total working years and total years with one's current manager (I. Setiawan, 2020).
3. *Demographic*: Age, education level and marital status (Ray & Sanyal, 2019; I. Setiawan, 2020)
4. *Satisfaction*: environment satisfaction and job satisfaction (I Setiawan, 2020).
5. *Factors specific to the job*: Working overtime and traveling for business (I Setiawan, 2020).

2.1.2 Current Insights

The section below will outline key insights discovered from an initial analysis of the data. The following three questions will be answered; 1) how many employees left Acorn in 2021, 2) who left the organization and 3) why did they leave the organization?

1) How Many Employees Left Acorn in 2021? As seen in figure 2.1A, a minority of employees left the organization representing 237, or 19.22%, of employees in total. This translates to 140, or 13.83%, of employees with a 'significant amount' of internal human capital investment.

Figure 2.1A Attrition levels



2)Who left the organization? As seen in figure 2.1B below, men left the organization at a slightly higher rate than women (13.10% vs. 10.83% respectively).

Figure 2.1B Attrition levels by gender

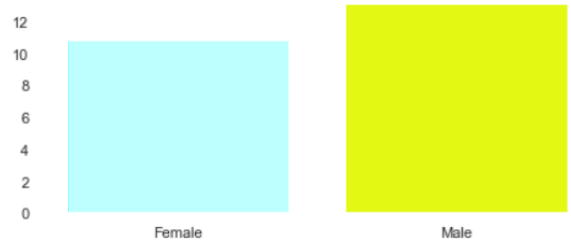


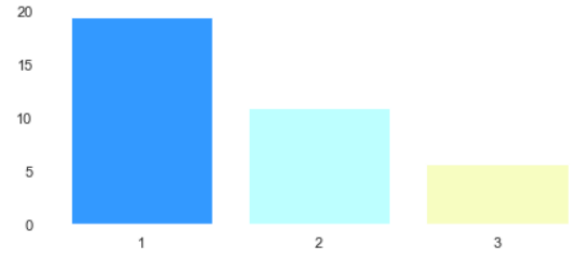
Figure 2.1C below, displays the attrition rate for each department. Out of the 359 employees who

work in the Sales department, 59 of those employees left the organization. This represents the highest attrition rate in Acorn.

Figure 2.1C: Attrition Rate by Department

As seen in Figure 2.1D below, the majority of employees held entry level positions. With 52 out of the 267 employees who held entry level jobs in Acorn leaving the organization. This contrasts with the 78 out of 632 employees with mid-level jobs and 10 out of 165 employees with high-level jobs.

Figure 2.1D : Attrition rate by department

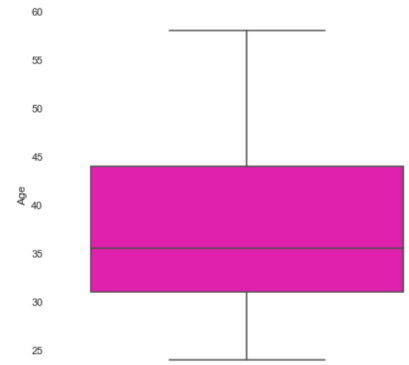


As seen in Figure 2.1E on the right, the majority of employees that left the organization occupied 'science' focused roles with 53 out of 347 employees leaving the organization. This contrasts with the 73 out of the 654 employees who occupied 'management and HR' roles and left the organization, and with the 14 out of the 151 employees who occupied 'representative' roles who left the organization.

Figure 2.1E : Attrition by Department

As seen in Figure2.1F below, the average age of attrition is 36 Years old.

Figure 2.1F : Age of Attrition

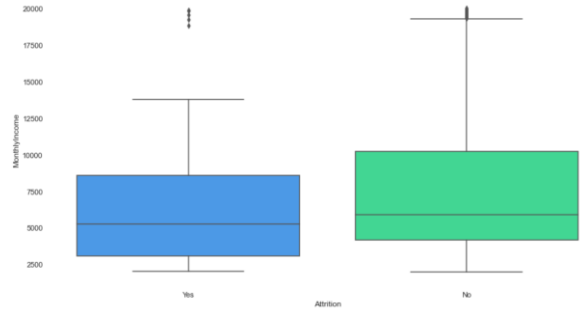


3)Why Did employees leave the organization?

Financial Factors: employees who left Acorn in 2021 earned, on average, $1,380.46 less as compared to their peers that did not leave the organization ($6,269.71 vs. $7,650.17

respectively). This can be seen in Figure 2.1G below.

Figure 2.1G : Average income by Attrition Status



Satisfaction: Regardless of attrition status, most employees rated their level of satisfaction highly. To understand the differences between employees who left Acorn and those that did not, each measure of satisfaction (environment satisfaction, relationship satisfaction and job satisfaction) will be considered separately and their associated relative scores analysed.

As seen in Figure 2.1H on the right, out of the 140 employees that left the organization, a greater proportion of employees rated their level of environment satisfaction as 'low' when compared to their peers that did not leave (14.46 vs. 6.35, respectively). Likewise, out of those who left Acorn, a greater proportion rated their level of environment satisfaction as 'good' when compared to those that stayed (16.87 vs. 14.86 respectively).

Figure 2.1H: Environment Satisfaction

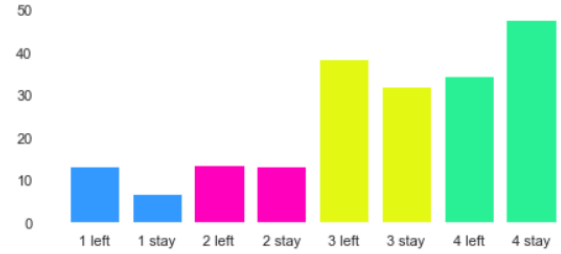
A similar trend can be seen in terms of relationship satisfaction depicted in Figure 2.1I. Out of those who left the organization, a greater proportion of individuals rated their level of relationship satisfaction as 'low' when compared to those who stayed (9.62% vs. 6.62% respectively). However, the ratings for 'good', 'excellent' and 'outstanding' were similar for each group.

Figure 2.1I: Relationship Satisfaction

In relation to job satisfaction, as seen in Figure 2.1J below, out of those that left the organization, a greater proportion of individuals rated their level of job satisfaction as 'low' when

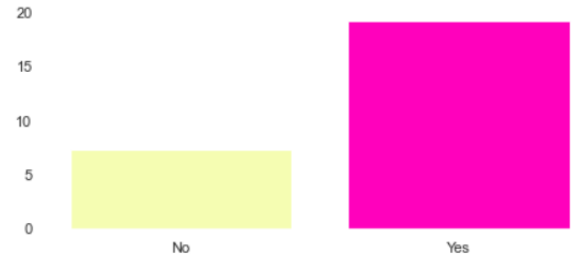
compared to their counterparts who stayed (13.39 vs. 6.70). Surprisingly, employees also rated their level of job satisfaction as 'good' at higher rates as compared to individuals who stayed (38.39 vs. 32.17).

Figure 2.1J : Job Satisfaction



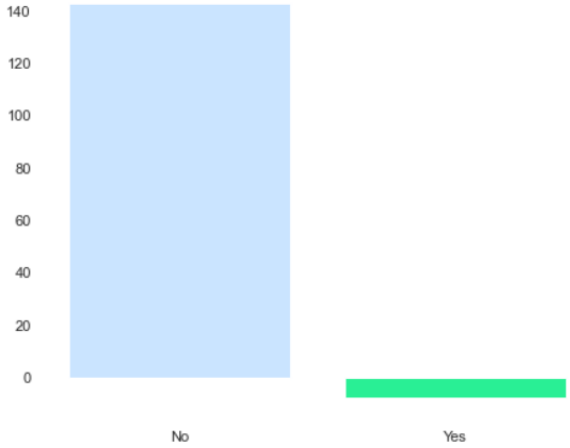
Factors specific to the job: As seen in figure 2.1K below, employees who left Acorn were more likely to work overtime

Figure 2.1K: Overtime by Attrition



As seen in Figure2.1L on the righ, employees that left Acorn received less training on average than their peers.

Figure 2.1L: Training Times Last Year Compared to Average by Attrition Level



Section 3: Model Creation

This section will outline the model creation process. The analysis was done with python 3.8 and the full code will be available in Appendix C.

The most significant challenge in modeling these data is the sever class imbalance (as mentioned previously, there are 1012 employees that did not leave the organization vs 140 who did). This lack of data will prevent the models from learning about this group and thus will make it difficult for the model to accurately predict which employees will leave the organization. To help correct for this problem, the data was re-sampled using a combination of the Synthetic Minority Over-Sampling Technique(SMOTE) along with Tomeck chains. This has the effect of up-sampling the minority class and down-sampling the majority class. Ultimately, this has the effect of reducing false negatives (predicting an employee will stay within the organization while they really intend to leave) while increasing false positives (predicting an employee will leave while they intend to stay) (Please see Appendix C- Section 6 for the implementation) .

3.1 Models Tested

Five models were fitted to the data i) Logistic Regression, ii) Random Forest Classifier, iii) Gradient Boosted Classifier iv) AdaBoost and v) Support Vector Machines. All models were tuned using Grid Search to find the optimal parameters. Please see Appendix C Section 5 for more information regarding the method used to tune the models.

3.2 Evaluation Metrics

I) F1-Score(macro average): The Macro average F1-Score is comprised of two measures i) precision and ii) recall. Precision asks, out of the employees who were identified as leaving the organization, how many actually left? While Recall asks, out of the number of employees who

Figure 3.3A: Model Performance

Model	*Precision*	*Recall*	*F1 (Macro)*	*F1 (Minority)*	*Kappa*
Linear Regression	**0.79**	0.77	**0.78**	**0.63**	**0.560**
Random Forest	0.65	0.71	0.67	0.48	0.344
Gradient Boosted	0.40	0.41	0.40	0.00	-0.195
AdaBoost	**0.79**	0.77	**0.78**	**0.63**	**0.560**
Support Vector Machine	0.74	**0.79**	0.76	0.61	0.514

left the organization, how many were identified as leaving?

The F1- Score summarizes precision and recall in one metric by calculating the harmonic mean between precision and recall.

Since the data is unbalanced (more employees stayed with the organization rather than left) the macro average is useful, as it gives each class an equal weighting. The macro average will compute the score for each class separately before taking the average.

II) Kappa: The Kappa statistics compares the model's accuracy as compared to a baseline of a model that assigns classes randomly. It is a measure that is commonly used with imbalanced data.

iii) Confusion matrix: provides a summary of how well the model predicted outcomes vs. their actual result. This summarizes how well the model does at predicting the minority class and will be used in combination with the other scores in order to gain a better sense of how the model predicts attrition.

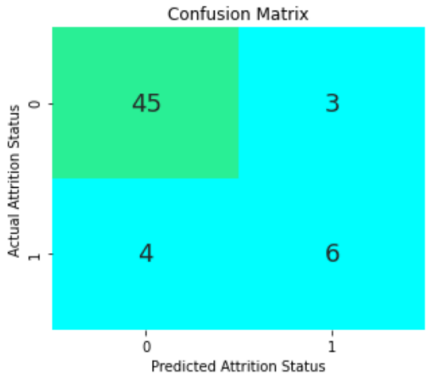
3.3: Results

As seen in Figure 3.3A above, logistic regression and AdaBoost are the top performing models. Both perform well when considering the macro average f-score, but when considering the f-score for the minority class and the kappa

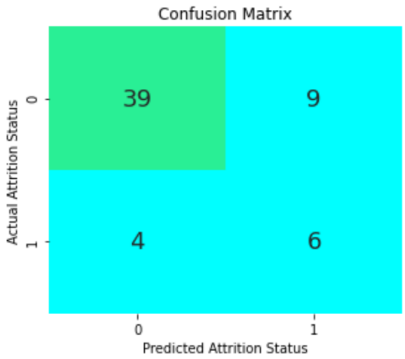
statistic performance decreases. This is again due to the sever class imbalance. Consulting the confusion matrixes in figure 3.3 below can help in understanding how well the model is able to accurately classify the minority case.

Figure 3.3: Confusion Matrix by Model

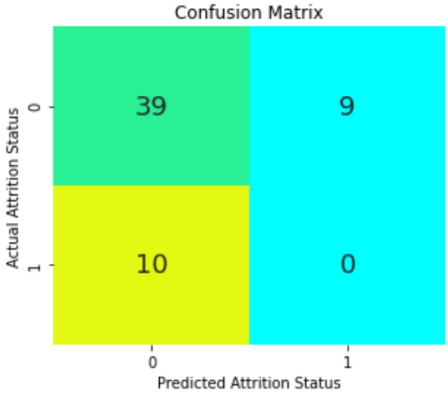
Linear Regression: 60%



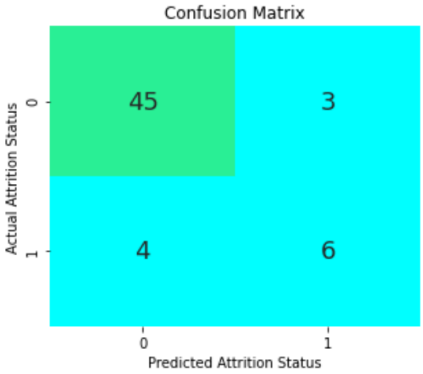
Random Forest: 60%



Gradient Boosting: 0%



Ada Boost: 60%



Support Vector Machine: 70%

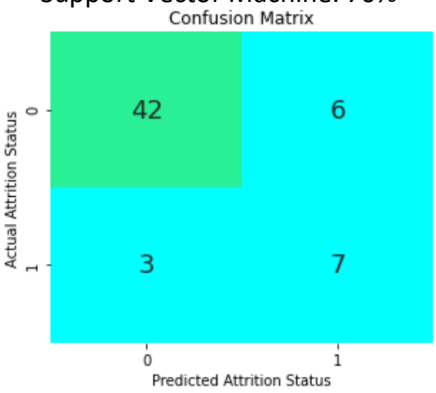


Figure 3.4 : Cross Validation

	Model	Min	Max	Mean
Kappa	*Logistic Regression*	-0.01	**0.63**	**0.35**
	Ada regressor	**0.02**	0.38	0.24
	Support Vector Machine	**0.02**	0.54	0.31
Precision	*Logistic Regression*	0.53	**0.80**	**0.69**
	Ada regressor	**0.56**	0.71	0.63
	Support Vector Machine	0.52	0.70	0.63
Recall	*Logistic Regression*	0.53	0.78	0.68
	Ada regressor	**0.57**	0.72	0.65
	Support Vector Machine	0.54	**0.80**	**0.69**
F1-score(minority)	*Logistic Regression*	0.19	**0.64**	**0.46**
	Ada regressor	**0.21**	0.54	0.37
	Support Vector Machine	**0.21**	0.54	**0.46**
F1-score(macro)	*Logistic Regression*	**0.53**	**0.80**	**0.69**
	Ada regressor	0.51	0.73	0.64
	Support Vector Machine	0.51	0.73	0.64

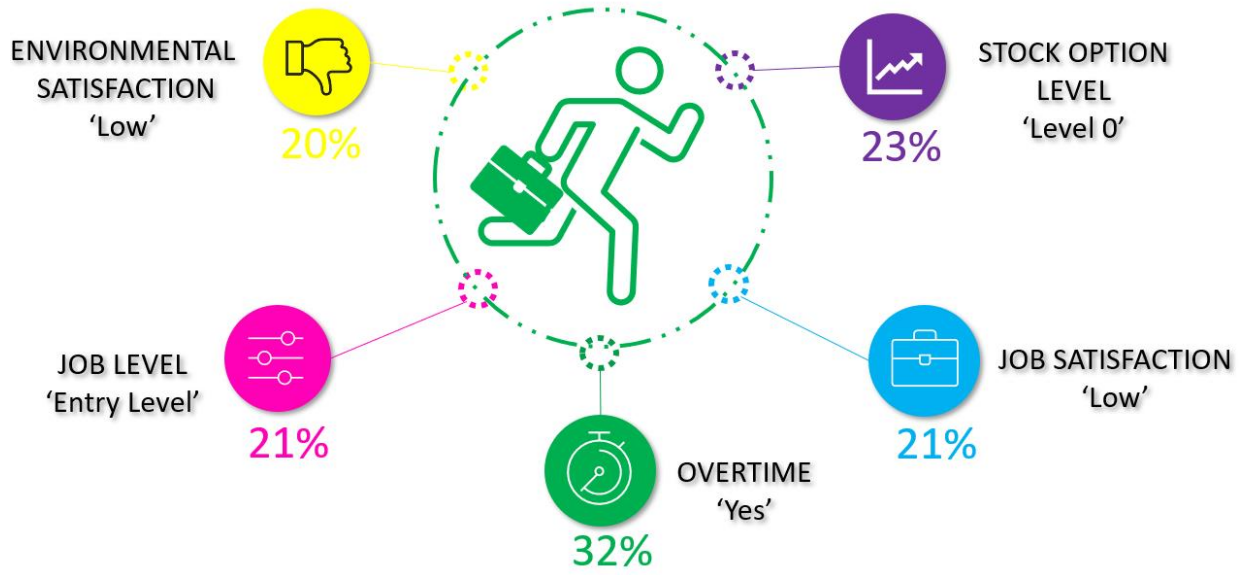
Gradient Boosting performs the worst, accurately predicting 0% of the employees that left the organization. Support Vector Machine is the best performing model, accurately predicting 70% of employees who left. While the rest of the models accurately predict 60% of employees who intend left.

Logistic regression, adaboost, and support vector machine are the top performing models.

iv) Cross Validation: To choose between the top 3 performing models, a 5 fold, K-Folds cross validation method was used. This provides some insight into how the model may preform with new data. The key metrics are Kappa and the F1-score for the minority class. As seen in Figure 3.4, all three models have the potential to perform poorly. Logistic regression achieves higher scores at maximum and performs the best on average. However, due to the variance in terms of results, all models need to be improved before final implementation.

v) Model Selection: When selecting a model for implementation, it is important to consider its interpretability. A key aspect of the current analysis on attrition is to develop anti-attrition programs, this requires a firm understanding of the drivers of attrition at Acorn. A major advantage of logistic regression is that its

Figure 3.4: Drivers of Attrition



features are readily interpretable. Thus, logistic regression will be implemented.

3.4 Feature Importance

Understanding the drivers of attrition will help HR practitioners to develop employee retention programs.

As seen in figure 3.4A, holding all else constant, an environmental satisfaction score of 'low' is associated with a 20% increase in the likelihood of attrition; similarly, a job satisfaction level of 'low' increases the likelihood of attrition by 21%. Working overtime is associated with a 32% increase in the likelihood of attrition and having no stock options is associated with a 23% increase in the likelihood of attrition. Thus, HR practitioners may consider improving environmental satisfaction, and job satisfaction; offering stock option levels and limiting working overtime in order to retain employees.

3.5 Dashboard

The final key deliverable for HR practitioners is a method to communicate the results of this project clearly. To this end, a dashboard will be created that will provide HR practitioners with an overview of existing and predicted patterns in attrition (as illustrated in Appendix D figure D1) and allow managers to identify specific employees at risk of leaving Acorn (as illustrated in Appendix D figure D2). Anti-attrition programs can then be targeted towards this group.

Section 4: Discussion/Conclusion

Attrition has been flagged as a priority area for Acorn. With the goal of reducing attrition levels to industry standard within one year. The current report lays the groundwork for further work around predicting attrition.

5 models have been tested; logistic regression, random forest regressor, gradient booster, support vector machine and adaboost. Logistic regressor performed the best in terms of accurately predicting which employees will leave the organization. However, it's performance is not sufficient – with some instances the model performing worse than a guess.

To improve performance, more data needs to be collected about employees that left the organization. This will allow the model to learn more about this group. For example, IBM can predict attrition in the next 6 months with a 95% accuracy rate. However, they have found that peer groups are defined as different things to different people (e.g. some employees considered former classmates rather than employees in the same department as their peers). This suggests that more complex relationships between features should be assessed. However, a balance should be maintained between gathering useful data and maintaining privacy.

In the interim, HR practitioners may consider improving environmental satisfaction, and job satisfaction; offering stock option levels and limiting working overtime. In order to reduce attrition levels, however, HR practitioners should be aware that the link between these factors and attrition intent may be tenuous.

In order to communicate the results of this project, a dashboard will be created that will provide HR practitioners with an overview of existing and predicted patterns in attrition and also allow managers to identify specific employees at risk of leaving Acorn.

Finally, users should note that attrition will not always be controllable as personal factors often influence an employees decision to leave the organization.

Appendix A – Data

Feature Name	*Description*	*Levels*	*Data Type*
Age	The age of the employee	N/A	Integer
Attrition	If the employee left IBM	Yes No	Categorical
BusinessTravel	If the employee travels for work	Non-Travel Travel_Frequently Travel_Rarely	Categorical
DailyRate	Wage for a single day	N/A	Integer
Department	Which department the employee is from	Human Resources Sales Research & Development	Categorical
DistanceFromHome	The distance from home to work	N/A	Integer
Education	Level of education	1 'Below College' 2 'College' 3 'Bachelor' 4 'Master' 5 'Doctor'	Integer
EducationField	Field of education	Human Resources Other Technical Degree Marketing Medical Life Sciences	Categorical
EmployeeCount	Number of employees	All entries are 1	Integer
EmployeeNumber	Employee Identification number	N/A	Integer
EnvironmentSatisfaction	Perceived level of satisfaction with the work environment	1 'Low' 2 'Medium' 3 'High' 4 'Very High'	Integer
Gender	Gender of employee	Female Male	Categorical
HourlyRate	Wage per hour	N/A	Integer
JobInvolvement	Level of involvement in job	1 'Low' 2 'Medium' 3 'High' 4 'Very High'	Integer
JobLevel[1]	Job title categories	1 'Entry-level staff' 2 'Senior staff' 3 'First-level Mgt' 4 'Middle Mgt' 5 'Senior Mgt'	Integer
JobRole	role	Human Resources Research Director Sales Representative Manager Healthcare Representative Manufacturing Director Laboratory Technician Research Scientist Sales Executive	Categorical
JobSatisfaction	How satisfied employees are with their job	1 'Low' 2 'Medium' 3 'High' 4 'Very High'	Integer

Feature Name	*Description*	*Levels*	*Data Type*
MaritalStatus	Marital status	Divorced Single Married	Categorical
MonthlyIncome	Amount of money earned per month	N/A	Integer
MonthlyRate*	Fixed salary per monthly	N/A	Integer
NumCompaniesWorked*	Number of companies worked at prior to joining IBM	N/A	Integer
Over18	If the employee is over 18	Y	Categorical
OverTime	If the employee works overtime	Yes, No	Categorical
PercentSalaryHike	Percentage increase in salary	N/A	Integer
PerformanceRating	Employee performance rating	1 'Low' 2 'Good' 3 'Excellent' 4 'Outstanding'	Integer
RelationshipSatisfaction	Level of satisfaction with their relationship	1 'Low' 2 'Medium' 3 'High' 4 'Very High'	Integer
StandardHours	Standard working hours	All 80	Integer
StockOptionLevel*	approval levels limiting access to options trading	0 'non' 1 'Covered Calls & Cash-Secured Puts' 2 'Long Options' 3 'Option Spreads'	Integer
TotalWorkingYears*	How many years employee has worked across all employers	N/A	Integer
TrainingTimesLastYear	Hours of training last year	N/A	Integer
WorkLifeBalance	Perceived level of work-life balance	1 'Bad' 2 'Good' 3 'Better' 4 'Best'	Integer
YearsAtCompany	Number of years at IBM	N/A	Integer
YearsInCurrentRole	Number of years in current role	N/A	Integer
YearsSinceLastPromotion	Number of years since last promotion	N/A	Integer
YearsWithCurrManager	Number of years with current manager	N/A	Integer

Appendix B

Job Role was condensed from 9 levels into 3. The new levels are Management_Hr, Representative and Science, they were condensed as follows;

New Levels	Managment_HR	Representative	Sci
Old Levels	'Manager' Manufacturing Director' 'Research Director' 'Sales Executive' 'Human Resources'	'Healthcare Representative' 'Sales Representative'	'Laboratory Technician' 'Research Scientist'

5 Job Levels were condensed into 3. The new levels are 1= Entry, 2=Mid and 3= Top. They were condensed as follows

New Levels	1	2	3
Old Levels	1	2 3	4 5

Education Level, condense to managment_Hr, Representative and Science

New Levels	'Buiss'	'Sci'	'Other'	'teck'
Old Levels	'Human Resources' 'Marketing'	'Life Sciences' 'Medical'	'Other'	'Technical Degree'

Appendix C – Code Presentation

Prepared by: Jacqueline James

Date: 2022-03-21

Kernel: Python 3.8

Relevant packages & versions: imblearn 0.9.0, scikit-learn 1.0.2, scipy 1.8.0, numpy: default, pandas: default

Section 1: Installing Packages and Uploading data

```
# Updating & Installing Packages
!pip install imblearn
!pip update imblearn
!pip update scikit-learn
!pip install kmodes
!pip install scipy --upgrade
!pip install category_encoders


# load Data
import pandas as pd
import numpy as np

ndf=pd.read_csv('IBM_onlytop2.csv')
```

Section 2: Feature Creation

```
# Creating Training Compare
ndf['Training_Compare']= 2

ndf['Training_Compare'] =
np.where(ndf['Department'] == 'Human
Resources',
ndf['TrainingTimesLastYear']-2.56,
ndf['Training_Compare'])
ndf['Training_Compare'] =
np.where(ndf['Department'] == 'Research
& Development',
ndf['TrainingTimesLastYear']-2.79,
ndf['Training_Compare'])
ndf['Training_Compare'] =
np.where(ndf['Department'] == 'Sales',
ndf['TrainingTimesLastYear']-2.85,
ndf['Training_Compare'])

# Job Role, condense to managment_Hr,
Representative and Science
ndf['Job_Role'] = 2

ndf['Job_Role'] =
np.where(ndf['JobRole'] == 'Manager',
'Managment_HR', ndf['Job_Role'])
ndf['Job_Role'] =
np.where(ndf['JobRole'] ==
'Manufacturing Director',
'Managment_HR', ndf['Job_Role'])
ndf['Job_Role'] =
np.where(ndf['JobRole'] == 'Research
Director', 'Managment_HR',
ndf['Job_Role'])
ndf['Job_Role'] =
np.where(ndf['JobRole'] == 'Sales
Executive', 'Managment_HR',
ndf['Job_Role'])
ndf['Job_Role'] =
np.where(ndf['JobRole'] == 'Human
Resources', 'Managment_HR',
ndf['Job_Role'])

ndf['Job_Role'] =
np.where(ndf['JobRole'] == 'Healthcare
Representative', 'Representative',
ndf['Job_Role'])
ndf['Job_Role'] =
np.where(ndf['JobRole'] == 'Sales
Representative', 'Representative',
ndf['Job_Role'])

ndf['Job_Role'] =
np.where(ndf['JobRole'] == 'Laboratory
Technician', 'Sci', ndf['Job_Role'])
ndf['Job_Role'] =
np.where(ndf['JobRole'] == 'Research
Scientist', 'Sci', ndf['Job_Role'])

# Job Level 1= Entry, 2=Mid and 3= Top
ndf['Job_Level'] = 2

ndf['Job_Level'] =
np.where(ndf['JobLevel'] == 1, 1,
ndf['Job_Level'])
ndf['Job_Level'] =
np.where(ndf['JobLevel'] == 2, 2,
ndf['Job_Level'])
ndf['Job_Level'] =
np.where(ndf['JobLevel'] == 3, 2,
ndf['Job_Level'])
```

```
ndf['Job_Level'] =
np.where(ndf['JobLevel'] == 4, 3,
ndf['Job_Level'])
ndf['Job_Level'] =
np.where(ndf['JobLevel'] == 5, 3,
ndf['Job_Level'])

# Job Role, condense to managment_Hr,
Representative and Science
ndf['Education_Field'] = 2

ndf['Education_Field'] =
np.where(ndf['EducationField'] == 'Human
Resources', 'Buiss',
ndf['Education_Field'])
ndf['Education_Field'] =
np.where(ndf['EducationField'] == 'Life
Sciences', 'Sci',
ndf['Education_Field'])
ndf['Education_Field'] =
np.where(ndf['EducationField'] ==
'Marketing', 'Buiss',
ndf['Education_Field'])
ndf['Education_Field'] =
np.where(ndf['EducationField'] ==
'Medical', 'Sci',
ndf['Education_Field'])
ndf['Education_Field'] =
np.where(ndf['EducationField'] ==
'Other', 'Other',
ndf['Education_Field'])
ndf['Education_Field'] =
np.where(ndf['EducationField'] ==
'Technical Degree', 'teck',
ndf['Education_Field'])

# Income_Group Comparison between
average income by department and Job
Level
ndf['Income_Groups']= 2

ndf['Income_Groups'] =
np.where((ndf['Department'] == 'Human
Resources')&(ndf['JobLevel'] == 1),
ndf['MonthlyIncome']-2733,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] == 'Human
Resources')&(ndf['JobLevel'] == 2),
ndf['MonthlyIncome']-5563,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] == 'Human
Resources')&(ndf['JobLevel'] == 3),
ndf['MonthlyIncome']-9623,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] == 'Human
Resources')&(ndf['JobLevel'] == 4),
ndf['MonthlyIncome']-16148,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] == 'Human
Resources')&(ndf['JobLevel'] == 5),
ndf['MonthlyIncome']-19198,
ndf['Income_Groups'])

ndf['Income_Groups'] =
np.where((ndf['Department'] == 'Research
& Development')&(ndf['JobLevel'] == 1),
ndf['MonthlyIncome']-2840,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] == 'Research
& Development')&(ndf['JobLevel'] == 2),
ndf['MonthlyIncome']-5291,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] == 'Research
& Development')&(ndf['JobLevel'] == 3),
ndf['MonthlyIncome']-10170,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] == 'Research
& Development')&(ndf['JobLevel'] == 4),
ndf['MonthlyIncome']-15634,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] == 'Research
& Development')&(ndf['JobLevel'] == 5),
ndf['MonthlyIncome']-19219,
ndf['Income_Groups'])

ndf['Income_Groups'] =
np.where((ndf['Department'] ==
'Sales')&(ndf['JobLevel'] == 1),
ndf['MonthlyIncome']-2507,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] ==
'Sales')&(ndf['JobLevel'] == 2),
ndf['MonthlyIncome']-5746,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] ==
'Sales')&(ndf['JobLevel'] == 3),
```

```
ndf['MonthlyIncome']-9282,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] ==
'Sales')&(ndf['JobLevel'] == 4),
ndf['MonthlyIncome']-15166,
ndf['Income_Groups'])
ndf['Income_Groups'] =
np.where((ndf['Department'] ==
'Sales')&(ndf['JobLevel'] == 5),
ndf['MonthlyIncome']-19088,
ndf['Income_Groups'])
```

In [5]:

```
ndf['MonthlyIncome_log']=np.log(ndf['Mon
thlyIncome'])
```

Section 3: Defining X and Y variables

```
X=ndf[['Age','Job_Level','MonthlyIncome_
log',
'StockOptionLevel','YearsAtCompany','Yea
rsWithCurrManager',
'OverTime','Education','MaritalStatus',
'BusinessTravel','Department','Relations
hipSatisfaction',
'EnvironmentSatisfaction','JobSatisfacti
on','Training_Compare',
'JobRole','Education_Field']]

# Defining the target variable
y=ndf[['Attrition']]
```

Section 4: Defining Models

```
from sklearn.linear_model import
LogisticRegression
from sklearn.tree import
DecisionTreeClassifier
from sklearn.ensemble import
RandomForestClassifier,GradientBoostingC
lassifier, AdaBoostClassifier
from sklearn import svm

#lr=LogisticRegression(C=0.01,
solver='liblinear',random_state=30)
lr=LogisticRegression(C=0.01,
solver='lbfgs',random_state=35)
rm=RandomForestClassifier(max_depth= 3,
max_features= 25,
min_samples_leaf=35, min_samples_split=
2,

n_estimators=35, random_state=30)
clf=GradientBoostingClassifier(learning_
rate=100,

max_depth=100,n_estimators= 10,

random_state=25)
ABC =
AdaBoostClassifier(n_estimators=34)
svm= svm.SVC(C=0.001, gamma=0.01,
             kernel='linear',
probability=True)
```

Section 5: Tuning Models

```
# Finding Best Parameters
from sklearn.model_selection import
GridSearchCV

model=svm # Change the Model as needed
param = {"n_estimators":
[10,15,20,25,30,35,40,45,50,100,150,500,
1000]}
folds = 10
search = GridSearchCV(model, param, cv =
folds)
search.fit(X_train_final ,
y_train_final)
print(search.best_estimator_)
```

Section 6: Classification Reports, Kappa Scores and Confusion Matrix for Each Model

```
from sklearn.model_selection import
StratifiedKFold
from sklearn.metrics import
f1_score,precision_score,recall_score
from sklearn.preprocessing import
LabelEncoder
import category_encoders as ce
from imblearn.combine import SMOTETomek
from sklearn.model_selection import
cross_val_score
from sklearn.metrics import
classification_report
from sklearn.metrics import f1_score
```

```
from sklearn.metrics import
cohen_kappa_score
from sklearn.model_selection import
train_test_split
import matplotlib.pyplot as plt
import seaborn as sns

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.05,
random_state=42)

encoder =
ce.OneHotEncoder(handle_missing='value',
handle_unknown='value',use_cat_names=Tru
e)
encoder.fit(X_train)
X_train_final =
encoder.transform(X_train)
X_test_final = encoder.transform(X_test)
label_encoder = LabelEncoder()
label_encoder.fit(y_train)
y_train_final =
label_encoder.transform(y_train)
y_test_final =
label_encoder.transform(y_test)

sme= SMOTETomek(random_state=2)
X_train_final, y_train_final =
sme.fit_resample(X_train_final,
y_train_final)

for model, name in
zip([lr,rm,clf,ABC,svm],
['lr','rm','clf','ABC','svm']):
    model.fit(X_train_final,
y_train_final)
    y_pred= model.predict(X_test_final)
    score= f1_score(y_test_final,
y_pred, average='macro')

kappa=cohen_kappa_score(y_test_final,
y_pred)
    cf=
classification_report(y_test_final,
y_pred)
    cm = confusion_matrix(y_test_final,
y_pred)
    print(name, cf)
    print(name, 'Kappa:', kappa)
    print(name, cm)
```

Section 7: K-folds Cross Validation, Classification Reports for each Model

```
from sklearn.model_selection import
StratifiedKFold
from sklearn.metrics import
f1_score,precision_score,recall_score
from sklearn.preprocessing import
LabelEncoder
import category_encoders as ce
from imblearn.combine import SMOTETomek
from sklearn.model_selection import
cross_val_score
from sklearn.metrics import
classification_report
from sklearn.metrics import f1_score
from sklearn.metrics import
cohen_kappa_score

#encoder=ce.BaseNEncoder
KFold= StratifiedKFold(n_splits=5,
random_state=10, shuffle=True)

# Convert DF to Array for K-folds
X=X5
X=np.array(X)
y=np.array(y)

# Start K Folds
for model, name in
zip([lr,rm,clf,ABC,svm],
['lr','rm','clf','ABC','svm']):
    for fold, (train_index, test_index)
in enumerate(KFold.split(X,y), 1):
        X_train, X_test =
X[train_index], X[test_index]
        y_train, y_test =
y[train_index], y[test_index]
        # Set Parameters for encoders
        encoder =
ce.OneHotEncoder(handle_missing='value',

handle_unknown='value',

use_cat_names=True)
        encoder.fit(X_train)
        X_train_final =
encoder.transform(X_train)
        X_test_final =
encoder.transform(X_test)
        label_encoder = LabelEncoder()
```

```
        label_encoder.fit(y_train)
        y_train_final =
label_encoder.transform(y_train)
        y_test_final =
label_encoder.transform(y_test)

        sme= SMOTETomek(random_state=2)
        X_train_final, y_train_final =
sme.fit_resample(X_train_final,
y_train_final)
        model.fit(X_train_final,
y_train_final)
        y_pred=
model.predict(X_test_final)
        score= f1_score(y_test_final,
y_pred, average='macro')

kappa=cohen_kappa_score(y_test_final,
y_pred)
        cf=
classification_report(y_test_final,
y_pred)

        print(name, cf)
```

Section 8: Confusion Matrix Visualization

```
model= svm
model.fit(X_train_final, y_train_final)
y_pred= model.predict(X_test_final)
cm = confusion_matrix(y_test_final,
y_pred)
print(model)
ax=plt.subplot()
sns.heatmap(cm,annot=True,
cmap=['#00FFFF','#E3F913','#FF01BC','#29
EF95'],annot_kws={'size':18})

ax.set_xlabel('Predicted Attrition
Status')
ax.set_ylabel('Actual Attrition Status')
ax.set_title('Confusion Matrix')
```

Section 9: Feature Importance

Logistic Regression - Normalized

For feature names

In [55]:

```
import numpy as np
import matplotlib.pyplot as plt
import sklearn as sk
from sklearn.preprocessing import
normalize

from sklearn.model_selection import
StratifiedKFold
from sklearn.metrics import
f1_score,precision_score,recall_score
from sklearn.preprocessing import
LabelEncoder
import category_encoders as ce
from imblearn.combine import SMOTETomek
from sklearn.model_selection import
cross_val_score
from sklearn.metrics import
classification_report
from sklearn.metrics import f1_score
from sklearn.linear_model import
LogisticRegression
from sklearn.tree import
DecisionTreeClassifier
from sklearn.ensemble import
RandomForestClassifier,GradientBoostingC
lassifier, AdaBoostClassifier
from sklearn import svm

svm= svm.SVC(C=0.001, gamma=0.01,
             kernel='linear',
probability=True)

#encoder=ce.BaseNEncoder
KFold= StratifiedKFold(n_splits=10,
random_state=10, shuffle=True)

# Convert DF to Array for K-folds
X=np.array(X)
y=np.array(y)

# Start K Folds
for model, name in zip([lr], ['lr']):
    for fold, (train_index, test_index)
in enumerate(KFold.split(X, y)):
        X_train, X_test =
X[train_index], X[test_index]
        y_train, y_test =
y[train_index], y[test_index]
        # Set Parameters for encoders
        encoder =
ce.OneHotEncoder(handle_missing='value',

handle_unknown='value',

use_cat_names=True)
```

```
        encoder.fit(X_train)
        X_train_final =
encoder.transform(X_train)
        X_test_final =
encoder.transform(X_test)
        label_encoder = LabelEncoder()
        label_encoder.fit(y_train)
        y_train_final =
label_encoder.transform(y_train)
        y_test_final =
label_encoder.transform(y_test)

        sme= SMOTETomek(random_state=42)
        X_train_final, y_train_final =
sme.fit_resample(X_train_final,
y_train_final)

        model.fit(X_train_final,
y_train_final)
        y_pred=
model.predict(X_test_final)

# get featurenames
feature_names=X_test_final.columns
print(feature_names)
```

Feature Importance

```
import numpy as np
import matplotlib.pyplot as plt
import sklearn as sk
from sklearn.preprocessing import
normalize

from sklearn.model_selection import
StratifiedKFold
from sklearn.metrics import
f1_score,precision_score,recall_score
from sklearn.preprocessing import
LabelEncoder
import category_encoders as ce
from imblearn.combine import SMOTETomek
from sklearn.model_selection import
cross_val_score
from sklearn.metrics import
classification_report
from sklearn.metrics import f1_score
from sklearn.linear_model import
LogisticRegression
from sklearn.tree import
DecisionTreeClassifier
from sklearn.ensemble import
RandomForestClassifier,GradientBoostingC
lassifier, AdaBoostClassifier
from sklearn import svm

svm= svm.SVC(C=0.001, gamma=0.01,
             kernel='linear',
probability=True)

#encoder=ce.BaseNEncoder
KFold= StratifiedKFold(n_splits=10,
random_state=10, shuffle=True)

# Convert DF to Array for K-folds
X=np.array(X)
y=np.array(y)

# Start K Folds
for model, name in zip([lr], ['lr']):
    for fold, (train_index, test_index)
in enumerate(KFold.split(X, y)):
        X_train, X_test =
X[train_index], X[test_index]
        y_train, y_test =
y[train_index], y[test_index]
        # Set Parameters for encoders
        encoder =
ce.OneHotEncoder(handle_missing='value',

handle_unknown='value',

use_cat_names=True)
        encoder.fit(X_train)
        X_train_final =
encoder.transform(X_train)
        X_test_final =
encoder.transform(X_test)
        label_encoder = LabelEncoder()
        label_encoder.fit(y_train)
        y_train_final =
label_encoder.transform(y_train)
        y_test_final =
label_encoder.transform(y_test)

X_train_final=sk.preprocessing.normalize
(X_train_final, norm='l2', axis=1,
copy=True, return_norm=False)

X_test_final=sk.preprocessing.normalize(
X_test_final, norm='l2', axis=1,
copy=True, return_norm=False)

        sme= SMOTETomek(random_state=42)
        X_train_final, y_train_final =
sme.fit_resample(X_train_final,
y_train_final)
```

```
        model.fit(X_train_final,
y_train_final)
        y_pred=
model.predict(X_test_final)

# get importance
importance = lr.coef_
for x in importance:
    for i,v in
enumerate(zip(feature_names, x)):
        print(i,v)
```

Appendix D – Dashboard

The full dashboard is available herehttps://public.tableau.com/app/profile/jacqueline2330/viz/INFO70042Assignment4Finaljjames/Trends

Figure D1: Current & Predicted trends in Attrition

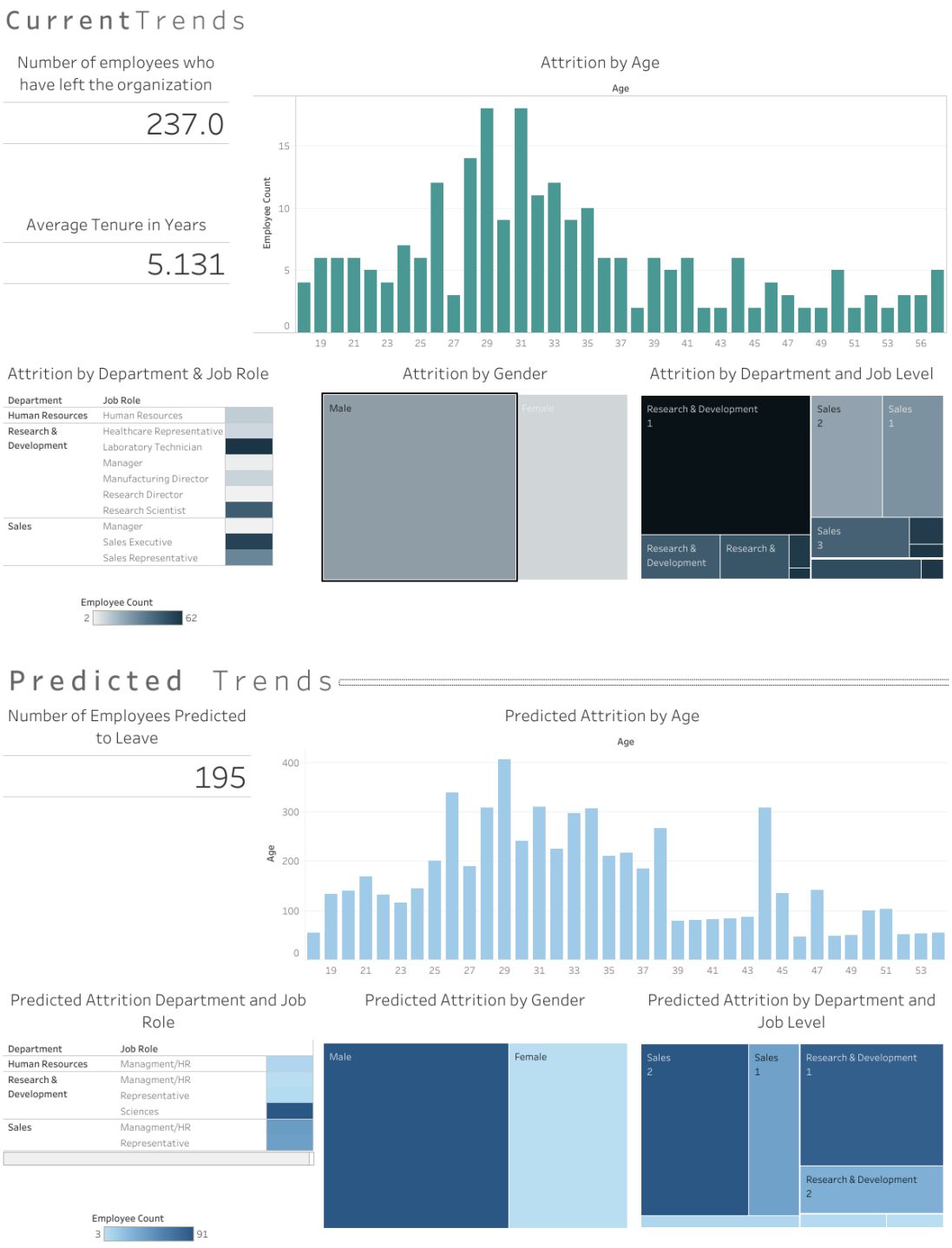
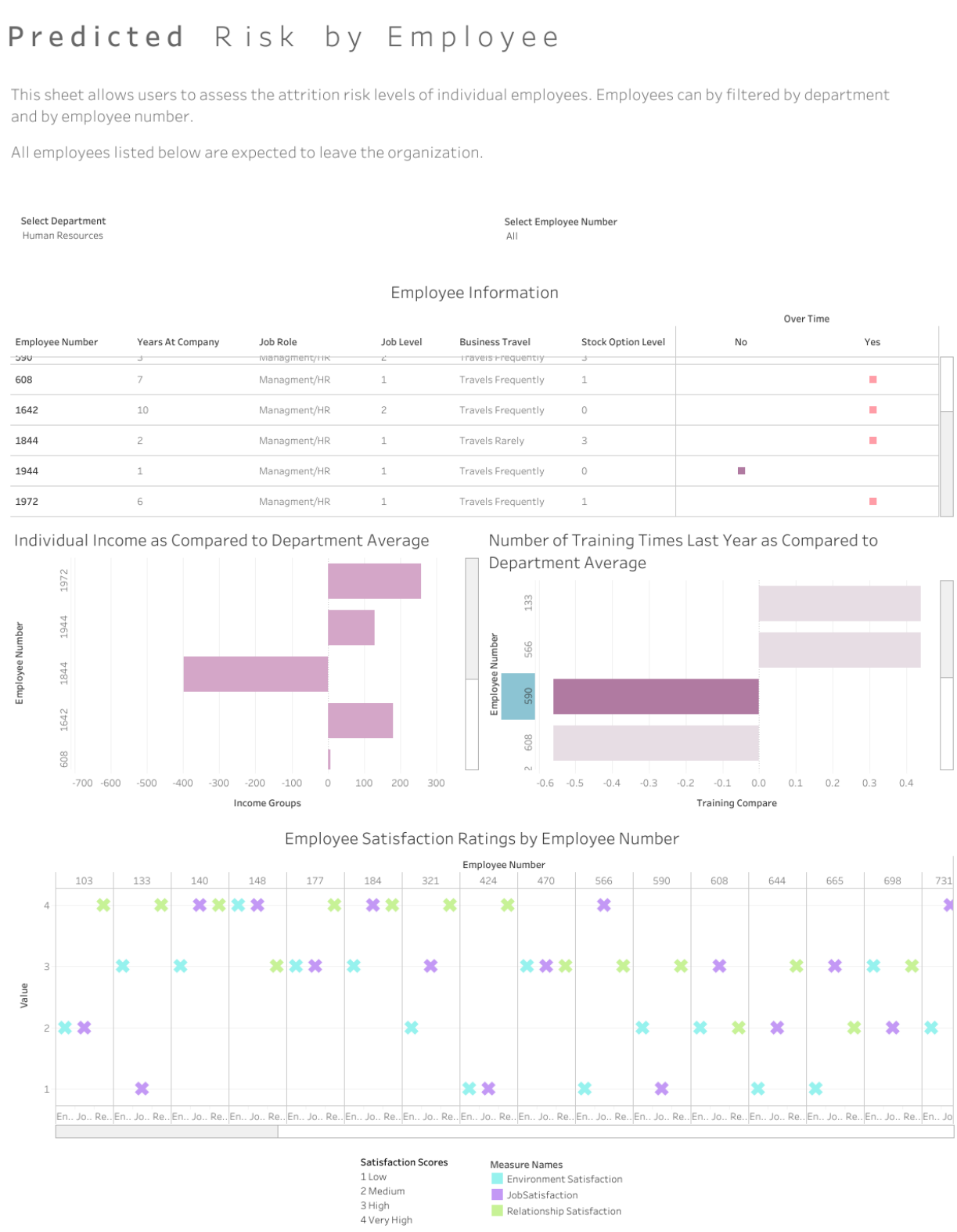


Figure D2: Dashboard for predicting Employee Attrition Risk



Works Cited

Data. Employee Attrition Data. Open Database Liscence. Kaggle. Avalible https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

Hancock, J. I., Bosco, F. A., McDaniel, K. R., & Pierce, C. A. (2013). Meta-Analytic Review of Employee Turnover. *Journal of Managment, 39*(3), 573-603. doi:DOI: 10.1177/0149206311424943

I Setiawan*, S. S. (2020). HR analytics: Employee attrition analysis using logistic Regression. *Materials Science and Engineering*.

Ray, A. N., & Sanyal, J. (2019). Machine Learning Based Attrition Prediction. *Global Conference for Advancement in Technology.* Bangalore, India.

Singh, M., Varshney, K. R., Wang, J., Gill, A. R., Faur, P. I., & Ezry, R. (n.d.). An Analytics Approach for Proactively Combating Voluntary Attrition of Employees. *2012 IEEE 12th International Conference on Data Mining Workshops.* IEEE Computer Society.

Vimoli, M., & Modi, S. (2021). Employee Attrition System Using Tree Based Ensemble Method. *International Conference on Communication, Computing and Industry*.

Winnea, S. D., Marescauxb, E., Selsc, L., Beverend, I. V., & Vanormelingen, S. (2019). The impact of employee turnover and turnover volatility on labor productivity: a flexible non-linear approach. *30*, pp. 3049-3079. The Internati onal Journal of Human Resource Management.

Acorn