Project Report

March 10, 2022

Team Seven

Harpreet Kaur Setia

Sara Tanabe

Talia Zalesne

Christopher Cannon

Poom Nichayapun

Business Understanding

[Company] is a leading data storage technology company focused on excellence and innovation in the space. According to recent financial statements, [Company] had a successful 2021 fiscal year, with around 11.48B in revenue. This was an almost 13% growth YoY from 2020. While they compete in a fairly saturated market, with top competitors like Western Digital, Micron Technology, and NETGEAR, their goal is to differentiate themselves by "reducing the cost and complexity of data storage". [Company] operates with over 42,000 employees across 10 offices in 8 different countries globally. With a Glassdoor rating of 3.8, employees seem generally happy with their jobs; however, the job market is extremely competitive, especially when it comes to recruiting and retaining great engineers. As a consulting group, we hope to help them anticipate and improve the current 6-7% turnover rate.

Problem Statement

At [Company], what drives employee turnover among the departments and what is the cost associated with this turnover? We are aiming to help [Company] predict employee churn in a way that allows them to mitigate it and/or anticipate it to allow them to begin the hiring process early and therefore reduce the cost of turnover. Employee churn is defined as the overall rate at which employees turnover (employees leave and new ones enter). In the case of [Company], they are interested very specifically in voluntary termination - as in an employee choosing to leave without being directed to by [Company].

Overarching Objectives

As consultants for [Company], we have three main objectives:

- First, evaluate the data provided to us by [Company] in order to begin understanding what types of employees churn. This will involve running analysis on the data and generating some descriptive statistics which will help determine which features may be valuable in a few ways. First, we will look at how much data exists in each column. We will also look at how much variability each feature has - if the entire column contains the same value it probably isn't helpful/if it contains too many values it may be useful to bin the data. Finally, we will explore how the raw data of each feature relates to our target - employee status (active/voluntarily inactive).
- Once we have gained a better understanding of the raw data and cleaned it up in the way we deem best, we will be able to move on to our second objective which will involve building models to make predictions surrounding future churn. To meet this objective, we will use Python to build and tune various different types of models to see which provides the best and most interpretable results. We plan to experiment mainly with decision trees, logistic regression, and neural networks.
- This all leads us to our final and main objective - to help [Company] with the issue of employee turnover. We plan to approach this from two angles. First, to help [Company] predict who will

churn and what factors led to this decision. This will allow us to make recommendations to [Company] on actions they may be able to take to reduce this issue. We also hope to help [Company] anticipate inevitable churn which will allow them to begin the hiring process early and in turn mitigate some of the cost associated with employees choosing to leave their company.

Project Management

Team Members

- Talia Zalesne is getting her MS in Business Analytics with a focus in Decision Sciences. Her background is in mathematics in the form of an undergraduate degree from The University of Colorado, Boulder and years spent teaching math of all levels. She excels in logical reasoning and therefore is proficient in coding despite her brief background in the subject. She loves hiking, skiing, and everything outdoors Colorado has to offer!

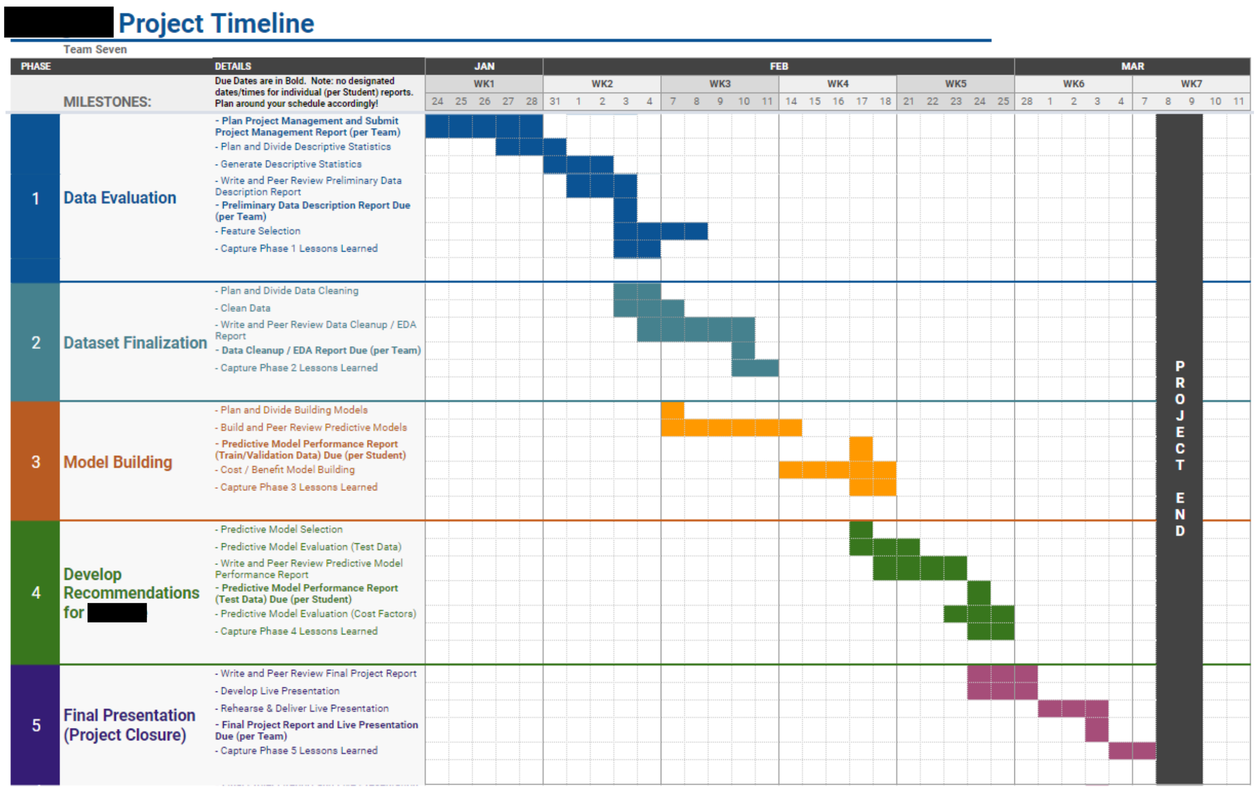
- Christopher Cannon (Chris) is getting his MS in Business Analytics after a 1 year stint as a Business Analyst because the experience was so positive. Before that brief experience, Chris accrued over 8 years of experience in project and program management as a Commissioned Officer in the US Army. He has a daughter named Molly so on weekends you'll find them at places like the childrens' museum or the zoo!

- Sara Tanabe is currently pursuing a MBA/MS in Business Analytics Dual Degree. After spending 6 years in the ad tech space, most recently as a Director of Account Management, she realized that her favorite part of the job was telling stories with data. She is excited to shift mindsets away from client management as she takes the next step as a data analyst at Visa in June. In her free time she enjoys traveling, eating delicious food, and hanging out with her friends, family, and dog, Henry.

- Poom Nichayapun is currently pursuing an MBA/MS in Business Analytics. Before MBA at Leeds, he gained more than 5 years of software development and business intelligence experiences from an e-commerce company and other software companies in his hometown: Bangkok, Thailand. Besides professional experiences, he, like other Coloradans, enjoys singing karaoke, cooking and outdoor activities such as skiing, hiking, trail running and occasionally training for marathon or ultramarathon events.

- Harpreet Kaur Setia is pursuing an MS degree in Business Analytics with Decision Science as an elective. She has a bachelor's degree in Engineering majoring in Information Technology. Her professional career began as a software developer with a leading MNC in India, where she contributed for 3.5 years. During this time, she got to briefly work with unstructured data, and that's what inspired her to delve deeper into the data industry. She also enjoys exploring the outdoors via biking and hiking and is a keen admirer of art.

Project Timeline

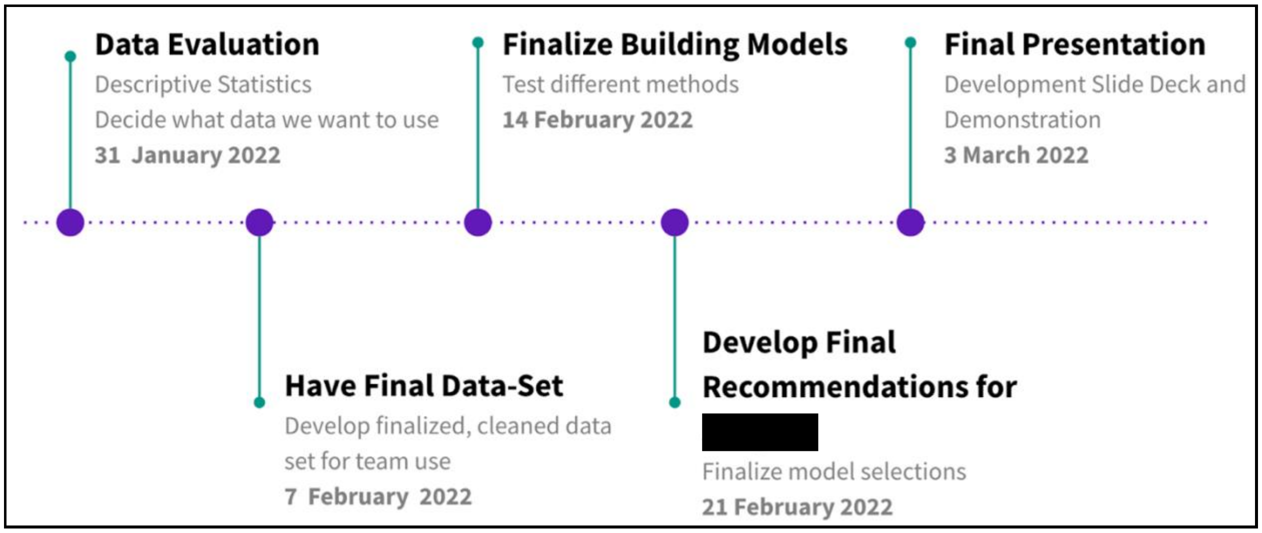


[redacted] Project Timeline

Team Seven

PHASE	MILESTONES:	DETAILS
		Due Dates are in Bold. Note: no designated dates/times for individual (per Student) reports. Plan around your schedule accordingly!
1	Data Evaluation	- **Plan Project Management and Submit Project Management Report (per Team)** - Plan and Divide Descriptive Statistics - Generate Descriptive Statistics - Write and Peer Review Preliminary Data Description Report - **Preliminary Data Description Report Due (per Team)** - Feature Selection - Capture Phase 1 Lessons Learned
2	Dataset Finalization	- Plan and Divide Data Cleaning - Clean Data - Write and Peer Review Data Cleanup / EDA Report - **Data Cleanup / EDA Report Due (per Team)** - Capture Phase 2 Lessons Learned
3	Model Building	- Plan and Divide Building Models - Build and Peer Review Predictive Models - **Predictive Model Performance Report (Train/Validation Data) Due (per Student)** - Cost / Benefit Model Building - Capture Phase 3 Lessons Learned
4	Develop Recommendations for [redacted]	- Predictive Model Selection - Predictive Model Evaluation (Test Data) - Write and Peer Review Predictive Model Performance Report - **Predictive Model Performance Report (Test Data) Due (per Student)** - Predictive Model Evaluation (Cost Factors) - Capture Phase 4 Lessons Learned
5	Final Presentation (Project Closure)	- Write and Peer Review Final Project Report - Develop Live Presentation - Rehearse & Deliver Live Presentation - **Final Project Report and Live Presentation Due (per Team)** - Capture Phase 5 Lessons Learned

Project Outcomes (Milestones)



Data Evaluation
Descriptive Statistics
Decide what data we want to use
31 January 2022

Have Final Data-Set
Develop finalized, cleaned data set for team use
7 February 2022

Finalize Building Models
Test different methods
14 February 2022

Develop Final Recommendations for [redacted]
Finalize model selections
21 February 2022

Final Presentation
Development Slide Deck and Demonstration
3 March 2022

Risks and Challenges

Risk/Challenge Description	Solution
Team Schedule	Weekly pre-scheduled meetings
Short Timeframe	Making sure there is no scope creep
Varying Levels of Expertise Amongst Team	Work towards each others strengths
Data Clarity	Cleaning Data and working off of one finalized dataset
Lack of Business Understanding	Researching [Company] and attending office hours to ask questions
Code Conflict	GitHub Moderator
Data Privacy and Security	Be mindful of how we are sharing data

Project Strategy

To accomplish the project objectives, our team has implemented the following strategies to foster collaboration and solidify team dynamics as we believe that, besides technical skills, strong team collaboration is another key aspect that would lead us to the success of the project.

Team Philosophy

As a team we are a learning organization, and as such have designated explicitly that we will capture lessons learned from our first project at the end of each phase so that we can prevent repeating our missteps in the second project

Communication

We regard this project as a software development project. Thus, we would strictly follow Agile Methodologies to efficiently manage the project in an attempt to deliver the product that aligns with user requirements that constantly change over time. To achieve this, we would conduct a weekly stand-up meeting every Monday to receive updates on the development and status of the project from each member. Slack channel would be another communication approach to broadcast or announce anything related to the project.

Model Work Division

To conform the data strategy prevalent among data-driven companies, we would implement predictive models based on one cleaned unified data set as a single source of truth (SSOT) to ensure that the predictive results, regardless of different algorithms, would align in the same direction. Once we have a stable and cleaned dataset, each team member will incorporate this dataset for model implementation and testing. To prevent code conflict, each person will create a new GitHub branch and work locally on his/her branch. During model implementation, he/she will need to test and evaluate the performance of the model with various hyperparameter configurations to achieve the best performance of his/her responsible model, which will be compared with those of other team members.

Eventually, one team member, possibly Poom, will be the Github moderator who performs CI/CD integration to ensure that all codes, once fully integrated, would function normally as an integral unit.

Data Visualization Work Division

Individually, each team member will be responsible for visualization based on their models. However, as a team, we will mutually develop a brand of the combination of visualizations we wish to appear when delivering the project.

Presentation Work Division

For the presentation, we will brainstorm for the format and the structure of presentations as a whole, while maintaining wiggle room for creativity for the section that each specific team member is responsible for.

PM Report Key Takeaways

- [Company] has been a worldwide leader for over 40 years and manufactures data storage technology and solutions for customers across the globe. With 42,000 employees working out of offices in 8 different countries, [Company] reported revenues of $11.48B in 2021, up 18% compared to the prior calendar year.
- The company is looking for business solutions to improve their current turnover rate of 7%. As consultants to the company, we aim to predict employee churn such that they are able to reduce their costs associated with turnovers.
- We are looking to explore and evaluate data provided to us by [Company] to discover first-hand descriptive statistics, which will give an insight on the attributes that may be valuable to analyze what kind of employees churn.
- We strategized on our approach to encourage collaboration and communication on progressions and bottlenecks. We marked down certain milestones throughout the project development stage to precisely follow the timeline and yield accurate deliverables.
- As a team, we will follow Agile methodology principles to efficiently manage the project with weekly sprints and stand-up meetings. We will also be using GitHub to better manage collaborations from all team members and moderate CI/CD integration.
- We will be building multiple predictive models to make projections on future churn. We aim to present our recommendations to [Company] within 6-7 weeks so that they can reduce this issue and also mitigate some of the cost associated with employees choosing to leave their organization.

Preliminary Data Analysis

Metadata

Within the 22.5MB excel spreadsheet data file provided to us by [Company], there are 85448 rows, each representing an existing or former employee, and 46 different features. With this vast amount of data, we realize that we will need to do a lot of data cleaning. To begin our analysis process, we have chosen to organize the different features into categories to help us understand variables that are related or possibly repetitive. These chosen categories include: Job Type, Length of Employment, Employee Age, Employee Demographic Information, Employee Location, Employee Education, Compensation, and Employee Performance. Examples of this can be found later in this report. Additionally, we recognized that only 8 of these features are numerical, 3 of them are Date/Time, and the remaining 35 are categorical. As the majority of the data is categorical, we will need to take this into consideration when choosing the appropriate predictive models to utilize. In the following, we will delve into more specific findings from our data exploration and process.

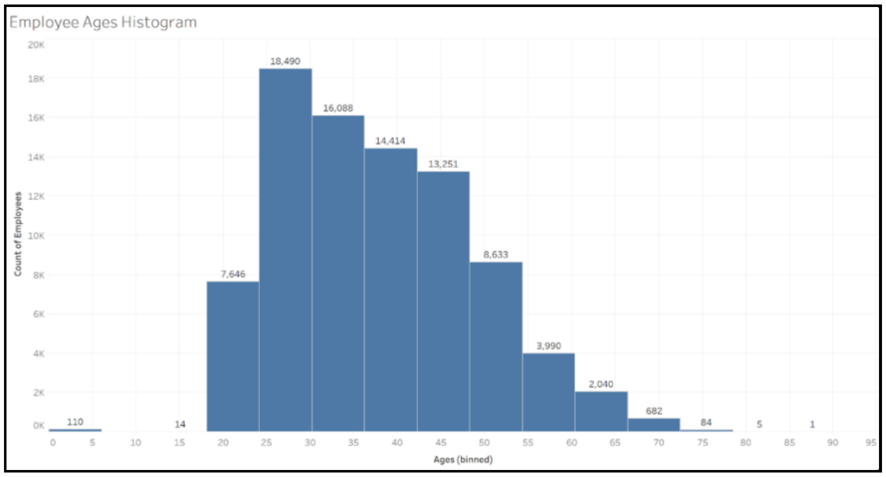
Data Analysis Tools/Software

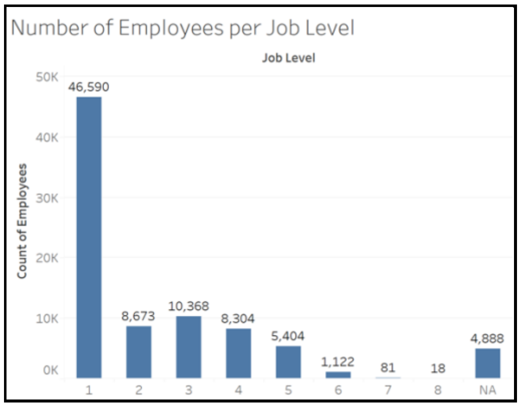
Similarly to our stance for the overall project, we feel it is best for code combination purposes to stick to one coding language, Python. We will be using GitHub to organize our individual work and our GitHub Moderator, Poom, will be responsible for combining everything at the end. We used Python to run describe(), summary() along with additional visualization libraries, namely MatPlotlib and Seaborn, to help us gain an overview of the dataset visually and conceptually. This allows us to be able to strategize our cleanup and preprocessing plan for this data set as the next phase of this project. In preparation for feature selection, we ran correlation tests and looked at various scatter plots to help us further understand which features are highly correlated. Besides Python, we leveraged Tableau to help us visualize relationships we consider important and complement what trends/insights Python visualization may have overlooked.

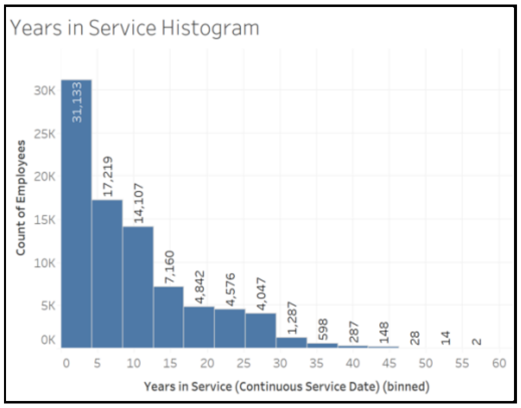
Data Exploration and Visualizations

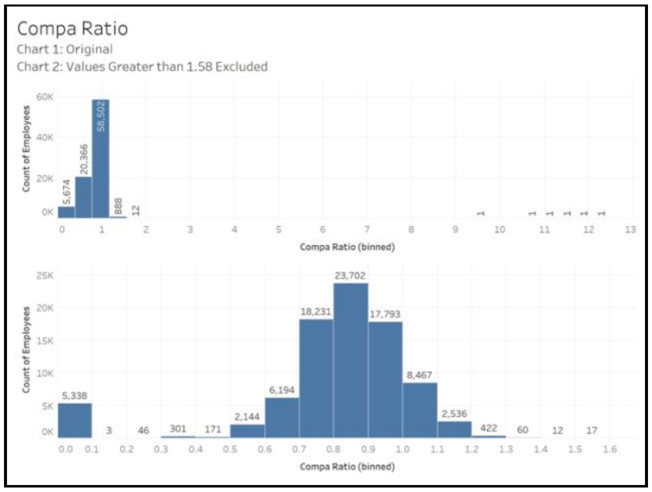
Shape of the Data - Histograms

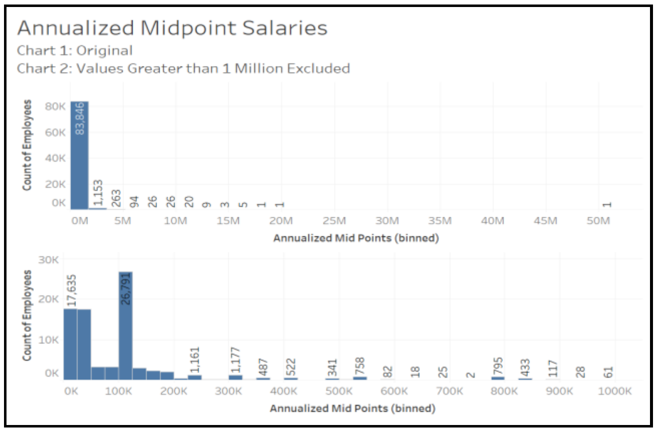
Our team produced histograms for each of the six numeric features. Three of these features produced legible results without manipulation (*Ages*, *Job Level*, and *Years in Service (Continuous Service Date)*). These histograms are displayed first. Then, the other three features required removal of various values to make the resulting charts legible. How the feature representations were adjusted, as well as views of their charts before and after adjustment, are provided in the following.

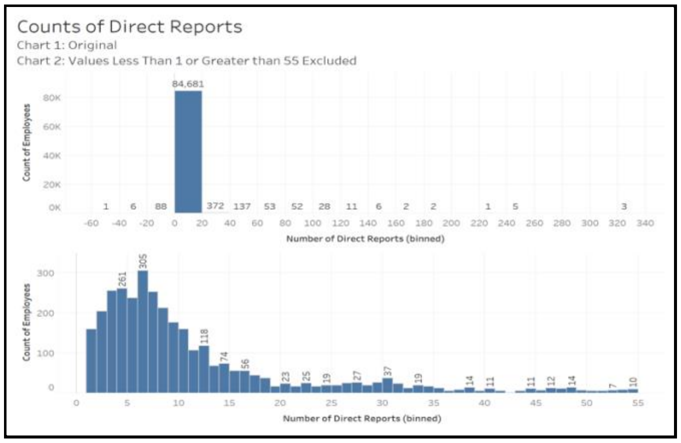

Employee Ages Histogram
Count of Employees
Ages (binned)
110
14
7,646
18,490
16,088
14,414
13,251
8,633
3,990
2,040
682
84
5
1


Number of Employees per Job Level
Job Level
Count of Employees
1: 46,590
2: 8,673
3: 10,368
4: 8,304
5: 5,404
6: 1,122
7: 81
8: 18
NA: 4,888


Years in Service Histogram
Count of Employees
Years in Service (Continuous Service Date) (binned)
31,133
17,219
14,107
7,160
4,842
4,576
4,047
1,287
598
287
148
28
14
2


Compa Ratio
Chart 1: Original
Chart 2: Values Greater than 1.58 Excluded
Count of Employees
Compa Ratio (binned)
5,674
20,366
58,502
888
12
5,338
3
46
301
171
2,144
6,194
18,231
23,702
17,793
8,467
2,536
422
60
12
17

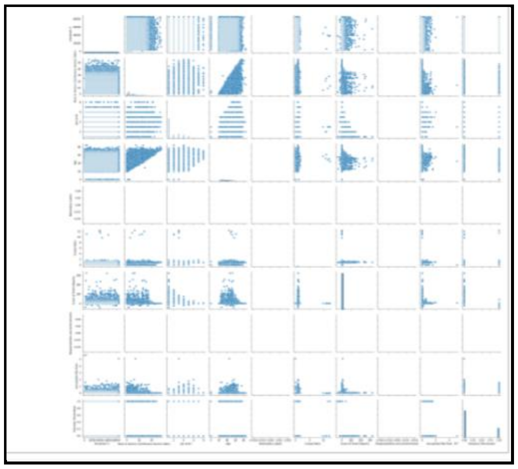

Annualized Midpoint Salaries
Chart 1: Original
Chart 2: Values Greater than 1 Million Excluded
Count of Employees
Annualized Mid Points (binned)
83,846
1,153
263
94
26
26
20
9
3
5
1
1
1
17,635
26,791
1,161
1,177
487
522
341
758
82
18
25
2
795
433
117
28
61


Counts of Direct Reports
Chart 1: Original
Chart 2: Values Less Than 1 or Greater than 55 Excluded
Count of Employees
Number of Direct Reports (binned)
1
6
88
84,681
372
137
53
52
28
11
6
2
2
1
5
3
261
305
118
74
56
23
25
19
27
37
19
14
11
11
12
14
7
10

Scatterplots

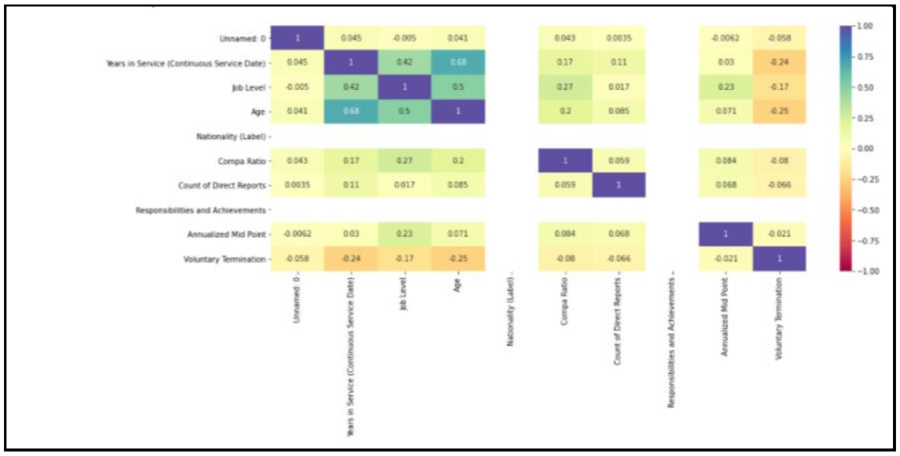
Since [Company] is interested in voluntary termination, we created an additional boolean column "Voluntary Termination" for those ex-employees that voluntarily resigned (coded as 'TRUE' and vice versa). Then we observe the correlation between this newly created "Voluntary Termination" and other 8 numeric columns.

Initially, we use pairplot to see the overview for the scatterplot for each pair of "Voluntary Termination" vs a numeric column. As seen in the pairplot below, most pairs do not display any apparent positive or negative correlation to "volunteer termination". Hence, the better choice to visualize the correlation would be correlation heat map as this plot can display numeric values.



Correlation Heat Map

Besides visualizing correlation between each pair of "Voluntary Termination" and any numeric predictors, we adopt Correlation Heat Map to quantify these correlation numbers.



As illustrated above, all corresponding variables show negative correlation to Voluntary Termination. Nevertheless, None of these correlations are significantly strong. The strongest correlation columns to “Voluntary Termination” are “Years in Service (Continuous Service Date)” and “Age”. Based on this discovery, we can infer that the longer [Company] employees work there, the less likely they are to leave the company. This inference would also resemble the relationship between age and voluntary termination as “Years in Service” and “Age” forms a strong positive relationship as high as 0.68.

Presumably, this trend may tie to the generational mindset as “Age” is directly related to “Generation”.

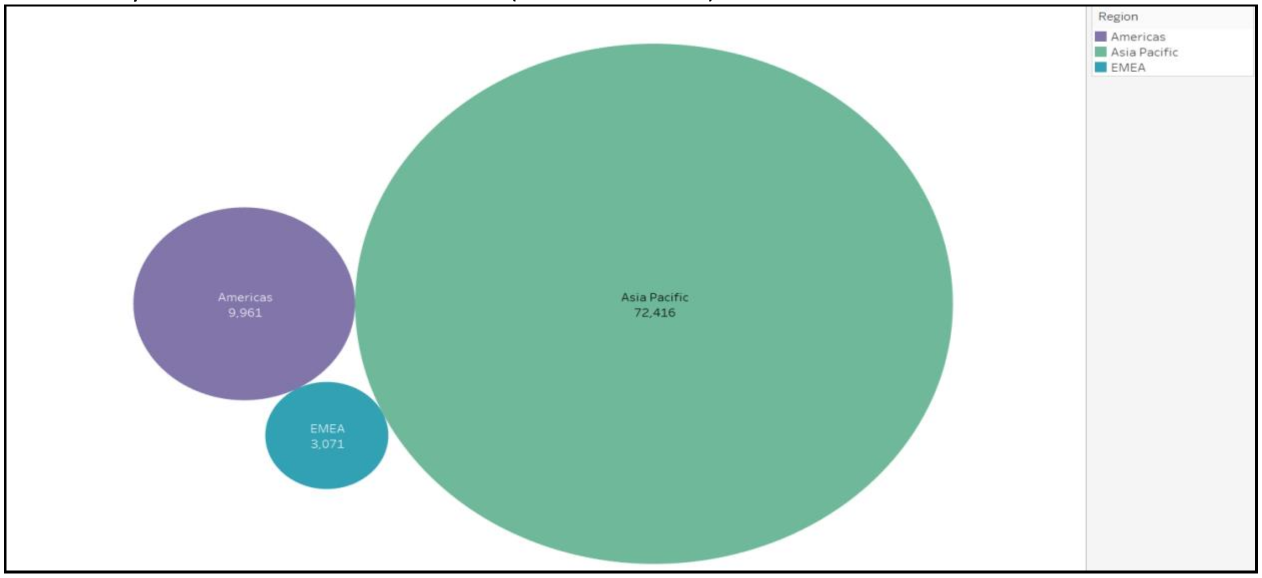
Visualizations based on demographics and job groups

Regions:

[Company] employees are spread across three regions worldwide:

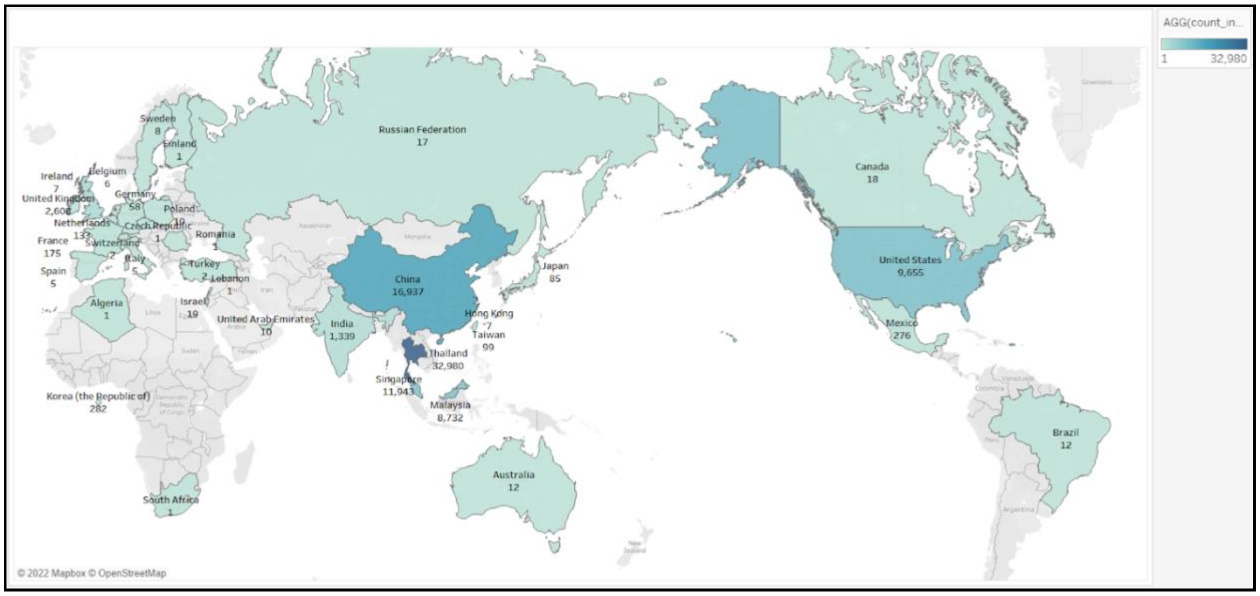
1. Asia Pacific,
2. The Americas, and
3. Europe, the Middle East and Africa (EMEA)

On exploring the metrics of these three regions, we inferred that Asia Pacific employees the largest staff followed by The Americas and then EMEA (as shown below):



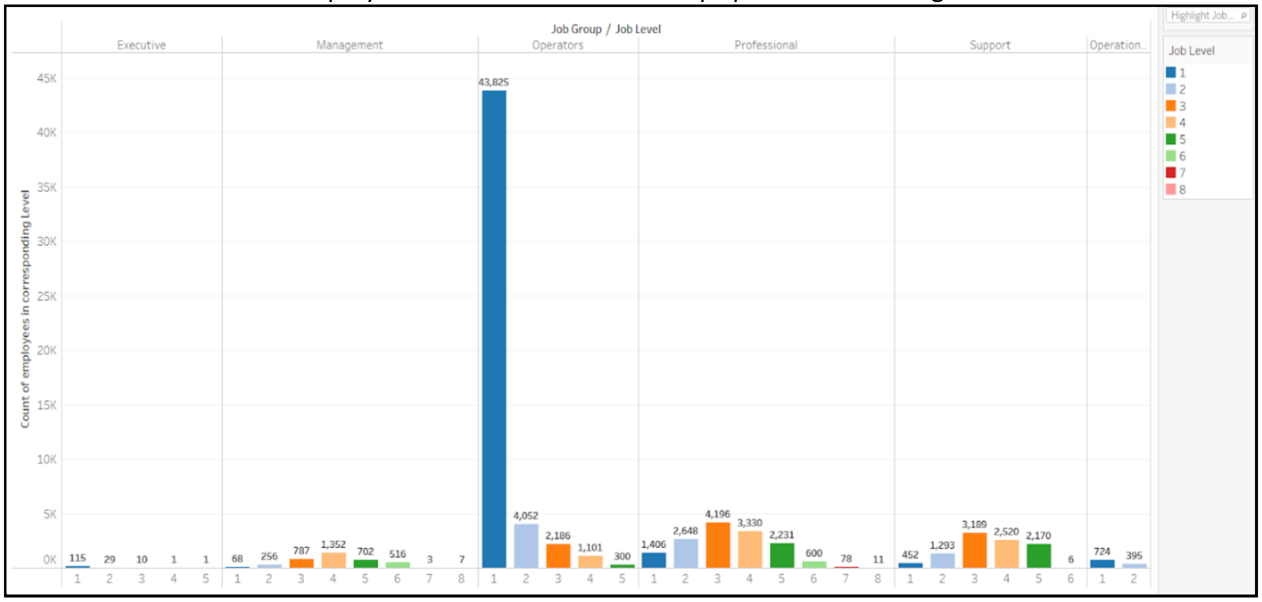
Countries:

[Company] employees are spread across 35 different countries across the globe. Amongst these countries, the highest concentration is at Thailand with 32,980 employees:



Job Level:

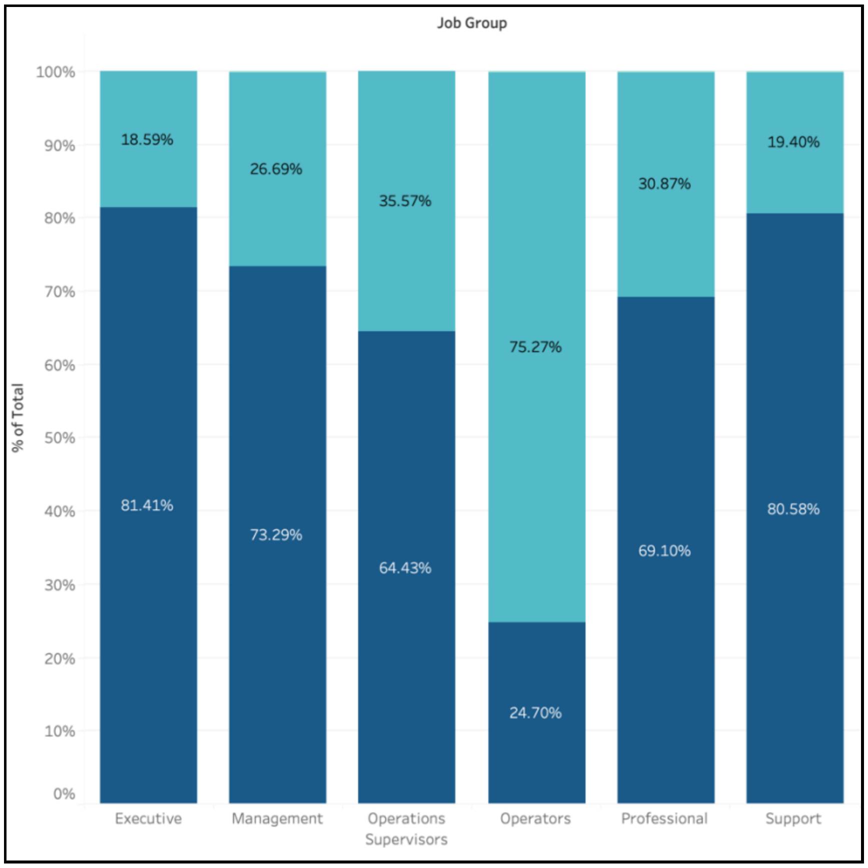
There is a distribution of 8 employee levels across all members at the [Company] corporation. Here, we show the count of each employee level across all Job Groups present in the organization:



≈

Gender Distribution:

Something we found interesting was the gender distribution weighing heavily male among leadership positions at [Company]. This will be something we dive further into throughout our analysis.



Descriptive Statistics

Numeric Data

The data has only 6 numeric columns (technically, initial analysis shows 8 numerical columns but two of them are populated with only NULL values). Below is a table of descriptive statistics for each column and our initial thoughts about each. Note, we defined the number of outliers based on first normalizing each column using [X - mean(X)]/std(X) and then counting the number of occurrences with values greater than 3 (or less than -3).

Column	Null Counts	Mean	Std. Dev	Max	Min	Number of Outliers
Years in Service	0	9.65	8.50	55.00	0.00	575
Job_Level	4888	-	-	8.00	1.00	-
Age	0	37.93	11.01	85.00	0.00	319
Compa Ratio	0	0.80	0.26	12.29	0.00	5346
Count of Direct Reports	0	0.62	5.82	324.00	-44.00	697
Annualized Midpoint	0	178316.13	592924.22	51120000.0	0.00	1356

Years in Service: We definitely think the amount of time an employee has been with the company will be an indicator of churn. Based on the mean and standard deviation, an outlier would be a person who has been with the company longer than 35.15 years.

Job Level: The job level column takes the values 1-8 and defines "the level of activities, duties and responsibilities and the nature of the work to be performed". Since these are really factors and not integers, descriptive statistics such as mean and standard deviation are not meaningful. In order to use this column, it will be necessary to combine it with the job category for each employee as a new feature. It also appears that the "Compensation Grade" column is already a combination of these two features.

Age: Age will almost certainly be a useful predictor, it looks like the average employee is around 38. Based on the data dictionary, it appears this value is relatively up to date - not just the age at which they were hired.

Compa Ratio: This is expected to be one of our most important features. Compa Ratio essentially tells what the employee makes in relation to the corresponding midpoint (where 1 means they make the midpoint in their position, and values greater than 1 indicate they make more than others, etc.). This feature essentially fixes all of the issues outlined below in the annualized midpoint column. Based on the mean and standard deviation, an outlier in this column would be anyone with a compa ratio higher than 1.58 or less than .02.

Count of Direct Reports: We are not sure what this column refers to. The closest explanation from the provided data dictionary is the "Organization Depth" category with the description "the number of layers within a leader's organization from CEO to individual contributor regardless of management level. Layers start with CEO at layer 0, the CEO's direct report is layer 1, etc." but this does not seem to match this column since the minimum value is -44.

Annualized Midpoint: Though the descriptive statistics are listed above, this column will probably not be very informative. In order to use it at all, we would have to convert all of the various currencies into USD. However, even then it may not be informative to compare salaries across different countries because there are still the factors of differing costs of living, etc.

Categorical Data

In this section we outline key characteristics of the categorical features of our dataset. These characteristics of each feature are described below.

Count of Unique Values: The count in this column represents the number of unique elements (observations) contained in that feature. For example, there are 3 possible Employee Statuses as indicated by the '3' in this column for that feature. And those possible statuses are: 'Terminated', 'Active', and 'Unpaid Leave'.

Team Description: Due to the large number of features we organized each feature into 1 of 10 descriptive categories. This enabled our team to better understand and "bin" various features into functions, and potentially identify candidate features for removal. Removal because these features may be redundant. The categories we established are: Compensation, Employee Age, Employee Demographics, Employee Education, Employee Location, Employee Performance, Index, Job Type, Length of Employment, and Termination Reason.

Count of Nulls: the count of null elements (observations) within each feature. A raw count of null values may not be easy to interpret, and so our team also represents the nulls as a percent of total observations in the next column.

Percent of Values are Null: the percent of the overall dataset those null values in the previous column represent. As mentioned previously, organizing null values in this way may be easier to appreciate the true level of null counts for a given feature.

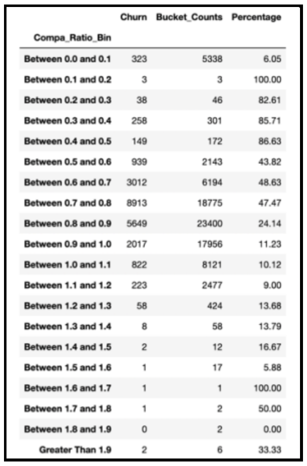
This information is summarized as a table and the full table is available in *Appendix A: Complete Categorical Data Description*. Below we provide a sample of the table (first five features) for brevity because there are 35 categorical features.

Feature	**Count of Unique Values**	**Team Description**	**Count of Nulls**	**% of Values are Null**
Employee Status	3	Job Type	0	0.0%
Employee Type	3	Job Type	0	0.0%
Job Category (Job Class)	13	Job Type	4883	5.7%
Job Group	6	Job Type	4883	5.7%
Job Family (Job Class)	176	Job Type	4883	5.7%

Data on Churn

After completing our initial data exploration, we wanted to look into some of the features as they are related to churn. We created a simple "Churn" column that maps to 1 if "Event Reason Name" is "Voluntary Termination" and 0 otherwise. By that definition, our dataset contains 22,419 employees who churned and 63,029 who are either still active or were terminated for other reasons. These are some tables from our findings:

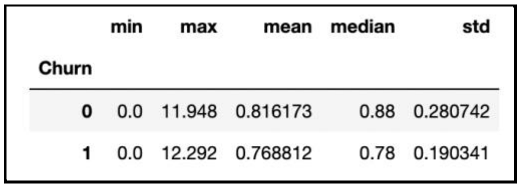
*Note the format for most tables is as follows: Churn column (number of people in that bucket who churned), Bucket_Counts (total number of instances in that bucket), Percentage (percent of people in that bucket who churned)



	Churn	Bucket_Counts	Percentage
Compa_Ratio_Bin			
Between 0.0 and 0.1	323	5338	6.05
Between 0.1 and 0.2	3	3	100.00
Between 0.2 and 0.3	38	46	82.61
Between 0.3 and 0.4	258	301	85.71
Between 0.4 and 0.5	149	172	86.63
Between 0.5 and 0.6	939	2143	43.82
Between 0.6 and 0.7	3012	6194	48.63
Between 0.7 and 0.8	8913	18775	47.47
Between 0.8 and 0.9	5649	23400	24.14
Between 0.9 and 1.0	2017	17956	11.23
Between 1.0 and 1.1	822	8121	10.12
Between 1.1 and 1.2	223	2477	9.00
Between 1.2 and 1.3	58	424	13.68
Between 1.3 and 1.4	8	58	13.79
Between 1.4 and 1.5	2	12	16.67
Between 1.5 and 1.6	1	17	5.88
Between 1.6 and 1.7	1	1	100.00
Between 1.7 and 1.8	1	2	50.00
Between 1.8 and 1.9	0	2	0.00
Greater Than 1.9	2	6	33.33

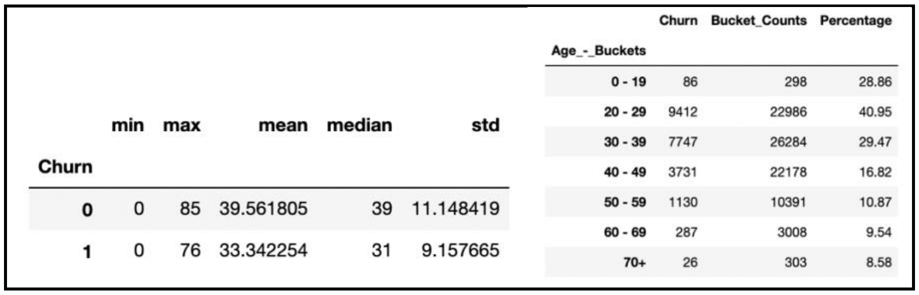
Compa Ratio Statistics:

In the first table, we see that the mean compa ratio among employees who have not churned is .82 vs. .77 among employees who did churn. In the second table, we see a clear general trend that the lower the compa ratio, the higher the percentage of employees that churned (the later buckets in the table start to not have enough instances for the percentages to be meaningful).



	min	max	mean	median	std
Churn					
0	0.0	11.948	0.816173	0.88	0.280742
1	0.0	12.292	0.768812	0.78	0.190341

Age Statistics:

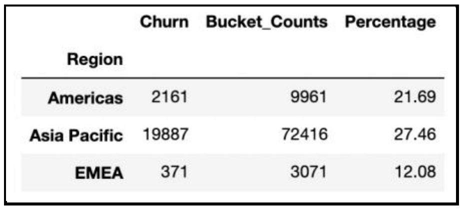


	min	max	mean	median	std
Churn					
0	0	85	39.561805	39	11.148419
1	0	76	33.342254	31	9.157665

	Churn	Bucket_Counts	Percentage
Age_-_Buckets			
0 - 19	86	298	28.86
20 - 29	9412	22986	40.95
30 - 39	7747	26284	29.47
40 - 49	3731	22178	16.82
50 - 59	1130	10391	10.87
60 - 69	287	3008	9.54
70+	26	303	8.58

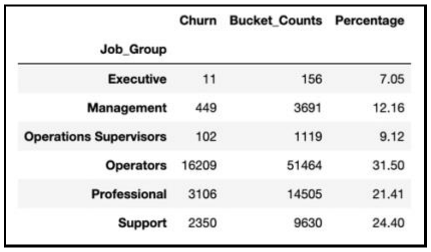
In the first table, we see that the mean age among employees who have not churned is 39 vs. 31 among employees who did churn. In the second table, we see a clear general trend that the lower the employee age, the higher the percentage of employees that churned.

Region Statistics:



	Churn	Bucket_Counts	Percentage
Region			
Americas	2161	9961	21.69
Asia Pacific	19887	72416	27.46
EMEA	371	3071	12.08

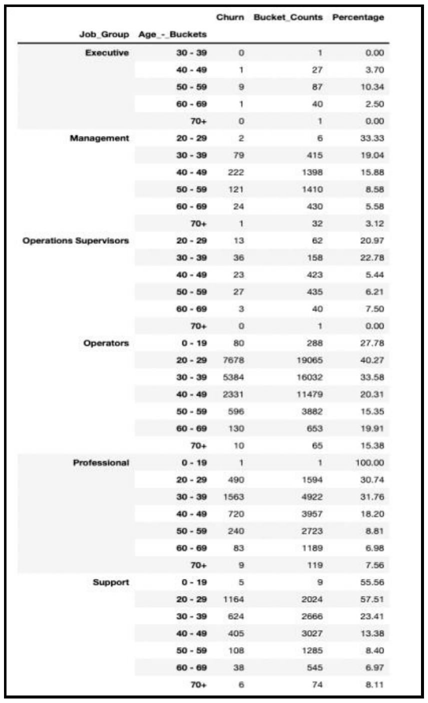
Job Group Statistics:



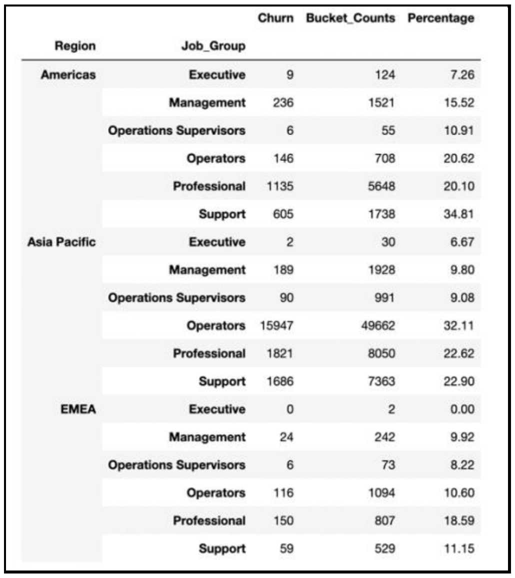
	Churn	Bucket_Counts	Percentage
Job_Group			
Executive	11	156	7.05
Management	449	3691	12.16
Operations Supervisors	102	1119	9.12
Operators	16209	51464	31.50
Professional	3106	14505	21.41
Support	2350	9630	24.40

As can be expected, there is the highest percent of churn among the operators and lowest among the executives. It is important to remember, however, that an executive churning is much more costly than an operator.

Combinations:



Job_Group	Age_-_Buckets	Churn	Bucket_Counts	Percentage
Executive	30 - 39	0	1	0.00
	40 - 49	1	27	3.70
	50 - 59	9	87	10.34
	60 - 69	1	40	2.50
	70+	0	1	0.00
Management	20 - 29	2	6	33.33
	30 - 39	79	415	19.04
	40 - 49	222	1398	15.88
	50 - 59	121	1410	8.58
	60 - 69	24	430	5.58
	70+	1	32	3.12
Operations Supervisors	20 - 29	13	62	20.97
	30 - 39	36	158	22.78
	40 - 49	23	423	5.44
	50 - 59	27	435	6.21
	60 - 69	3	40	7.50
	70+	0	1	0.00
Operators	0 - 19	80	288	27.78
	20 - 29	7678	19065	40.27
	30 - 39	5384	16032	33.58
	40 - 49	2331	11479	20.31
	50 - 59	596	3882	15.35
	60 - 69	130	653	19.91
	70+	10	65	15.38
Professional	0 - 19	1	1	100.00
	20 - 29	490	1594	30.74
	30 - 39	1563	4922	31.76
	40 - 49	720	3957	18.20
	50 - 59	240	2723	8.81
	60 - 69	83	1189	6.98
	70+	9	119	7.56
Support	0 - 19	5	9	55.56
	20 - 29	1164	2024	57.51
	30 - 39	624	2666	23.41
	40 - 49	405	3027	13.38
	50 - 59	108	1285	8.40
	60 - 69	38	545	6.97
	70+	6	74	8.11



Region	Job_Group	Churn	Bucket_Counts	Percentage
Americas	Executive	9	124	7.26
	Management	236	1521	15.52
	Operations Supervisors	6	55	10.91
	Operators	146	708	20.62
	Professional	1135	5648	20.10
	Support	605	1738	34.81
Asia Pacific	Executive	2	30	6.67
	Management	189	1928	9.80
	Operations Supervisors	90	991	9.08
	Operators	15947	49662	32.11
	Professional	1821	8050	22.62
	Support	1686	7363	22.90
EMEA	Executive	0	2	0.00
	Management	24	242	9.92
	Operations Supervisors	6	73	8.22
	Operators	116	1094	10.60
	Professional	150	807	18.59
	Support	59	529	11.15

Here we have run a similar analysis but the first table is broken up first by job group and then further by age buckets and the second table is broken up first by region and then by job group.

Data Issues

- **Repetitive nature of some columns and how we will choose to incorporate the information. For example,**
    - Length of employment features: We need to decide how precise we want to be. The options will be to use the 4 buckets laid out in the column "Length of Service - Buckets", to use the number of years from this column, or to create a new column of the number of days of employment calculated based on the hire date columns and the termination date column/today's date if they are still listed as active.
    - Age features: The first option on how to incorporate age would be to use the "Generation" column which essentially bins the ages into 4 categories. We are leaning away from incorporating this because the spread of the data within this column is not very good. The next option is to use the "Age Bins" column which seems to do a pretty good job of dividing the values (by decade). Finally, we could use the straight "Age" column as a continuous predictor.
    - The "Location Name" column is repeated exactly.

    - There appears to be a bit of a hierarchy where we have “Job Group” and then we build on that with “Job Category” and then we also have “Job Level”. However, expanding on this we then have “Compensation Grade” which seems to combine “Job Category” with “Job Level” AND we have “Comp Grade Profile” which appears to combine the previous two things with “Location Name”. Overall, these will be highly correlated/repetitive columns and it will be necessary to decide exactly how to use them.

- **Outliers**
    - Relating to the outliers in the compa-ratio column, it is worth noting that there are 6 employees with values of 9 and higher. All of these employees are terminated and all of them have a job level of either 2 or 3. They have mostly been with the company for a comparatively long time but it is worth asking [Company] if these values are intentional or potential mistakes. It is also worth noting that almost all employees with a compa-ratio less than 0.02 have a NULL value for their job category - this may be worth asking [Company] about as well.
    - Annualized Midpoint of Salaries experienced significant distribution skew as a result of outliers. The vast majority of observations earn less than 1 Million (83,846 observations), with the remaining values ranging from 1.01 Million to 49.85 Million. This column is not in USD denomination, so this issue may be resolved or exacerbated when we convert to a standard common denomination like the USD.

- **Null values**
    - Nationality (Label)/Responsibilities and Achievements: These are technically listed as “numeric columns” but are populated only with NULL values so will obviously be excluded completely.
    - The education data in general (there are 6 columns on this material) is filled with almost all NULL values. It will be hard to incorporate this data outright but if we can get more insight into why so much of it is NULL, we may be able to use it in some form (for example: employee chose to provide education data Y/N).
    - Certain employees have null values in their Job Group fields, which might make it difficult to analyze churn rate based on job groups. We will have to figure out what groups employees with similar characteristics belonged to, and then find out replacements for these null values.
- **Sanity checks**
    - The Frequency feature does not have an entry in the data dictionary and therefore we do not know how to interpret. For example, what is the difference between an observation value of Hourly 2028 and an observation value of Hourly 2080?
    - Count of direct reports also experienced outliers, including some of the counts of direct reports values that are negative. After a discussion we decided that it is not sensible and these observations may need to be removed, or handled another way. There are also a lot of 0 values within “Count of Direct Reports” among various job categories (including operators) where we would expect only the CEO to have a value of 0 here.

Data Cleaning Introduction

In order to move forward with building a model to provide [Company] with insights on how to reduce churn, our next step was to clean the data. As it can be daunting to figure out where to begin, we decided to rely on guidance from data experts at Tableau. Tableau's article, "Guide To Data Cleaning: Definition, Benefits, Components, And How To Clean Your Data", lays succinct steps to follow in order to clean a messy dataset. We have modified these steps to meet our needs as outlined at a high level below.

Step 1: Remove duplicate or irrelevant observations
Early on in our data exploration, we had given each variable a "category" to help us easily identify repetitive columns for data cleaning purposes. Specifically, we focused much of our effort here on the repetitive location and job categories. We also chose to remove some columns that we believed were not relevant to our analysis.

Step 2: Fix structural errors
As the vast majority of variables are categorical, we decided to transform many of these variables via ordinal assignment and one-hot coding. We also had many variables that had too many unique values to work with, so in those instances we created inclusive bins for specific categories.

Step 3: Filter unwanted outliers
There were a few instances where there were extreme outliers. In these instances, we plan to take a mean of each variable (and categorized), and we will then change the value of this variable. In other instances we will remove the row all together.

Step 4: Handle missing data
There were a few instances where the vast majority of a column contained NULL values. In these cases, we chose to get rid of the entire column as they would not be useful in the analysis

Step 5: Validate and QA
There were some variables that we were undecided about whether to keep or not, so to air on the side of caution, we kept more data than we believed we needed, specifically in instances of repetitive data. We plan to run different versions of our models that include different variables to see which will give us the best fit.

In the following sections, we will provide specifics about each of the steps outlined above.

Data Cleaning

Dropped Columns

As discussed in the preceding paragraphs, the first step of our data cleaning was to drop columns. At the time of writing this report, we have decided as a team to drop at least 27 columns. These columns we dropped generally fall into one of two categories. First, they are duplicative or empty (predominantly NA observations). Second, they are irrelevant to the analysis at hand or made irrelevant as a result of some decision making and/or feature engineering (which we will discuss in a subsequent section).

First, we dropped duplicative or empty columns. Since we are in the infancy of this project, we decided that our threshold for removing columns based on their null-value counts will be 100%. In other words, we will only remove columns that contain more than 75% null (for the time being). Otherwise, the reasons for removing these columns are rather straightforward and therefore are presented in tabular format, below.

Feature	**Type**	**Reason**	**Details**
Location_(Location_Name)	Categorical	Duplicative	Is a duplicate of *Location_Name*
Nationality_(Label)	Categorical	Empty	0 non-null values (100% null)
Responsibilities_and_Achievements	Categorical	Empty	0 non-null values (100% null)
Rating_Label	Categorical	Empty	12614 non-null (85% null)

Second, we targeted irrelevant columns. We decided these columns are irrelevant for reasons that are fairly particular to each column. We aggregated columns into reasons for removal where applicable, and subsequently listed columns individually with their reason(s) for removal as listed below.

<u>Education</u>
The details of our feature engineering are discussed in the next section. Therefore, suffice it to say for now that we removed the following education-related columns because we engineered a new column that captured what we believed was important from the following. *Degree_(Picklist_Label)*, *Did_you_graduate?_(Picklist_Label)*, *Major_(Picklist_Label)*, *Education_Record_is_blank?*, *Highest_Degree*, and *Is_Highest_ Degree*.

<u>Similarity - Job Information</u>
Some columns with job position information were similar to (although not complete duplicates of) other columns. For example, *job_category_(Picklist_Label)* was very similar to *Job_Category_(Job_Class)*. In this case we dropped the former because it contained more NAs than the latter. Another related column we dropped was *Job_Family_(Job_Class)*, because this column was a more granular version of the previously mentioned *Job_Category_(Job_Class)*.

Keeping the *Job_Family_(Job_Class)* column could have led to two specific problems. First, this could have introduced a problem of multicollinearity since both columns are related in that one is merely more specific than the other. But describing the same information. Second, we noticed that 120 of the 176 unique values in *Job_Family_(Job_Class)* had less than 100 observations each. This low count of observations per unique value means these values would be uninformative and thus this level of granularity is most likely not helpful.

Another example of one column being a nested and more granular version of another is the *Comp_Grade_Profile*. We found this column was a more granular version of the *Compensation_Grade* column (essentially combining compensation grade with location), which we kept. We chose to remove *Comp_Grade_Profile* for the same reasons we removed *Job_Family_(Job_Class)* from the previous paragraph. That is, multicollinearity and the increased granularity resulted in substantial numbers of unique values without numerous observations. Specifically, 85.1% of the 699 total unique values had less than 100 observations each.

Other examples are the *Length_of_Service_-_Buckets*, *Length_of_Service*, and *Continuous_Service_Date*. The first of these columns is based on the second and the second is based on the third. This is a prime opportunity for multicollinearity to disrupt model effectiveness. Therefore, we elected to remove the first column and the third column, since the second column contains all of the information of the other two and would not require further pre-processing (other than scaling).

<u>Homogeneity within Columns</u>
Some columns did not contain meaningful information because they were predominantly homogeneous. For example, *Time_Type* included 85,088 full-time employees and 360 part-time employees. That is, 99.6% of records are full-time employees. Without sufficient variation in the data we suggest such columns will not be informative during modeling. Accordingly, the *Is_Fulltime_Employee?* column is not informative either. As a result, we dropped both columns.

The second example of homogeneity is the *Employee_Type* column, which we also dropped. 97.0% of employees were Regular (as opposed to Intern or Fixed Term). And, as we will discuss below, Interns were removed from the dataset before dropping this column, meaning it only became more homogenous and uninformative.

<u>Cost and ROI Information</u>
There are two columns we dropped before running models which we intend on using at a later point. These columns are *Currency* and *Annualized_Mid_Point*. These columns will be useful when we are running financial implication models including cost of churn and ROI of implementing our recommendations. However, they are not particularly useful for predicting churn itself. Specifically, Annualized Midpoint is not useful for predicting churn because even once all currency is converted into USD, it is still unfair to compare salary information among such a variety of locations and costs of living. The Compa Ratio column is meant to take this discrepancy into account and allow for fair comparisons among the whole data set; however we will discuss the issues within that column later.

We removed the *Frequency* column after our meeting with [Company] on February 3rd. During this Zoom-based meeting we asked them to explain the *Frequency* column. Their answer was that this column is used to represent how often their employees are paid, and it varies by country. They then went on to explain that this column was primarily for their use, and not to be used in our analysis.

<u>Similarity - Age Information</u>
We explored cases where age information was stored in multiple columns in the same or different formats. For example, *Generation* (generation an employee is binned into) and *Age_Buckets* (binning of employee ages into 10-year groupings) provide information similar enough to the *Age* column (ages of employees, so in fact those columns are based on the *Age* column) that we were yet again concerned about multicollinearity. Further, the generation column provided a similar issue of homogeneity with about 67% of the column representing Millenials and almost the entire rest of the 33% representing GenX. In this case, we have removed *Generation*, but not *Age* or *Age_Buckets* (yet) and will remove one of those two after running models and determining which one provides more useful information.

<u>Engineering a Target Variable</u>
As mentioned, the details of our feature engineering are discussed in a subsequent section. That being said, in order to produce a target variable we could use to train a model we honed in on the aspects of *Event_Reason_Name* and *Termination_Reason(externalName)* we believe are useful for solving the business problem at hand. After extracting the information we needed to produce a binary churn/ no-

churn target variable we dropped these two columns (*Termination_Reason(externalName)* and *Event_Reason_Name*) as well as *Employee_Status* since we felt this information was captured as much as necessary in our target variable..

To recap the steps above, we started with 46 columns in our dataset, engineered 2 new columns (and transformed others), and removed a total of 28 columns. After moving *GID_Anonymized* into the Index position, this leaves us with a new dataset of 19 columns. Below, we discuss the rows we dropped outright and the steps we took to engineer our features.

Dropped Rows

GID_Anonymized

The purpose of the GID_Anonymized column is to provide a unique ID that maps each row to a specific employee. After we had finished our feature engineering and dropped the columns mentioned above, we ran the value_counts() method on the GID_Anonymized column. We noticed that there were many duplicate values, which actually turned out to be fully duplicated rows. We used the drop_duplicates() method to drop duplicated rows and then were left with 3 duplicated GIDs. After further exploration, we could see that these rows were identical except NA had a value for ethnicity and one was NA. We manually dropped the last three rows of duplicated GIDs containing NULL ethnicities and then set GID_Anonymized to be our index name.

Interns

We decided to drop interns for two reasons. First, there was only a small number of records (~1,000) who represented interns. Second, our partners at [Company] informed us that Interns are not a party of interest.

Between the Interns and the duplicate rows identified by finding duplicate values in GID_Anonymized, we removed 4,644 records from the dataset.

NULL Values in Job Group/Job Category/Job Level/Compensation Grade

As of right now, our data set contains about 4,848 rows which are populated with NULL values regarding the employee's role at [Company]. We are aware that [Company] is looking into why these rows have NULL values however, in the case that they are not able to get us this information - these rows will be dropped as well.

Engineering/Binning Columns

Once we had dropped the appropriate rose and columns, we continued with cleaning up the data we had in the following ways.

Data Manipulation

Education:

Given the extensive number of NULL values across the education columns and the fact that education information we have is based on what employees chose to voluntarily disclose, we created a new column that maps 1 if the employee chose to share their education information and 0 otherwise. We ran the

Pearson correlation coefficient between this new column and our target variable (churn or not) and got a value of -0.313 which is not insignificant. We dropped the rest of the education columns and will continue with only this column. We would like to note, however, that this is simply putting a bandage on a bigger wound. Ideally, we would prefer to have been able to use better detailed education data in our analysis but could not run anything meaningful with the small amount of data provided.

Data Binning/Bucketing

Marital Statuses:

Originally the marital status column had 10 different values that could be taken. Given the similar nature of some of these options, we decided to create a new column binning them together. We feel this new column captures all the relevant data in a more concise way and takes care of the potential issue of including labels that appear less than 1000 times.

Our new column has 4 possible marital statuses and was created in the following way:

- “Single” if the original column was ‘Single’
- "Married/Living Together" if the original column took any of 'Married', 'Domestic Partner', 'Domestic Partner (Unregistered Marriage)', 'Living Together', or 'Widowed With Surviving Pension'
- "Divorced/Widowed" if the original column took any of 'Divorced', 'Legally Separated', or 'Widowed'
- “Unknown” if the original column was ‘Unknown’ or NULL

Locations:

Originally our location column contained 137 different locations. More importantly, 66 of these locations appeared less than 10 times and 101 of them appeared less than 50 times. First, we noticed that up to 71 of these locations began with “remote” and these all had very few employees attached to them. We decided to bin all of these into one location titled “Remote” and then gathered all of the other locations with less than 50 employees into a location called “Other”. We mapped the rest of the locations to their original value. Our final data set has 35 unique locations.

Ethnicities:

Similar to marital statuses and locations, we noticed the same problem in ethnicities (too many different options with too little data in some). We realized that binning races together can be potentially color blind. Nevertheless, we felt that we were able to do this in a fair way due to the amazing diversity of our team. Our new column looks as follows:

- "Native American" if the original column was "American Indian or Alaskan Native, not Hispanic or Latino" or "Native Hawaiian or Other Pacific Islander, not Hispanic or Latino"
- "White" if the original column took any of "White, not Hispanic or Latino", "Caucasian", or "Eurasian"
- "Southeast Asian" if the original column took any of "Malay", "Sarawakian", "Vietnamese", or "Filipino"
- "South Asian" if the original column took any of "Indian", "Sikh", or "Pakistani"
- "Other/Unknown" if the original column was NULL or took any of "I do not wish to disclose", "Two or More Races, not Hispanic or Latino", "others"
- The rest of the races listed were mapped to their original values

Dealing With Outliers

Count Of Direct Reports:
"Count of Direct Reports" indicates how many people directly report to each employee. For example, an entry level operator would have 0 direct reports while the managers would have more than zero. We noticed that there were 86 employees in our data with negative values in this column. Clearly, there is no situation where this value should ever be less than zero. We looked at the compensation grade of each of these employees and found they were all managers of some sort. The first solution we thought of was to simply map the negative values to the same positive value but thought this may be too presumptuous. Instead, we took the mean "Count of Direct Reports" grouped by "Compensation Grade" and mapped any negative values to the corresponding mean value (based on the "Compensation Grade" of the employee in question).

Compa Ratio:
The other field with obvious outliers is the Compa Ratio field. As of right now, we have not decided how to handle this column. It is clear that salary is probably the most important predictor when it comes to employee churn but the way the column has been engineered as of now renders it essentially useless. The issue is this: Compa Ratio is calculated based on the Midpoint Salary of each employee's location, job type, and job level. Because of this, Compa Ratio is constantly changing as either the midpoint changes or the employees actual salary changes. However, once an employee is terminated, their Compa Ratio is not frozen - meaning as the midpoint of the job they held changes (likely increases), their Compa Ratio decreases. Essentially, we do not have any salary information for the employee at the point they were terminated and can't reasonably build analysis on our previous findings that a lower Compa Ratio indicated a likeliness to Churn. We know that [Company] is looking into this issue so pending their response, we will decide how to proceed with this feature.

Target Variable

The goal of our research is to predict employee churn. This means, we want to predict which employees will voluntarily choose to leave [Company]. Though it seems fairly straightforward, we wanted to make sure we were considerate in the way we chose to define an employee that "churned". The first obvious definition of churn was under the column titled "Event Reason Name" that sometimes took the value "Voluntary Termination". However, upon further inspection, we found that often employees labeled as "Involuntary Termination" took the value "Job Abandonment" in the "Termination Reason" column. Job abandonment refers to when an employee essentially stops showing up - so quits without any sort of notice. To us, this also qualifies as employee voluntary churn, so we decided to include it in our target. Our target variable is binary and takes a value of 1 for either of the reasons mentioned above or a 0 in any other case (which could include active, laid off, fired, etc.).

We have considered another type of target variable which would allow us to run more of a multi-class classification problem as opposed to a binary one. In this case, we would probably use the following labels: "Active", "Voluntary Churn", "Involuntary Termination: Not Employees Fault", "Involuntary Termination: Employees Fault". We are aware that [Company] is only interested in in employees who voluntarily leave but wonder if it may be weird to make that prediction after lumping together active/laid off/ and fired employees as the only other option (as in, is it weird to predict someone leaving - the negative - while also considering employees who were fired as the positive class). If time permits, we will probably attempt to run models using this type of target as well.

Data Subsets For Comparison

Considering the amount of data we have and all the probable factors that could affect the churn rate, our team has decided to take into consideration multiple subsets of data for making predictions. Ultimately, we propose to run several statistical and predictive models on these datasets to solve the business problem using the best model.

Job Type Classification: Since we have varying stratification of Job categories/types, we decided to focus on these classifications on separate degrees.

- For our first data subset, we plan to focus on Compensation Grade as one of the main predictors as this facilitates the most granular combination of Job types and levels. This classification has 65 unique records.
- In the next data subset, we want to use Job Category and Job Level as separate explanatory variables to have a broader categorization as compared to Compensation grade segregation. This segregation gives us a broader umbrella of 13 job categories clubbed with 8 job levels. It should provide the same information as compensation grade alone but will allow for predictions considering these variables separately.
- In the final dataset for this model, we intend to incorporate employee's Job Groups and their corresponding Job Levels separately to have a comprehensive evaluation based on their profession. With this classification, we will be working with 6 job groups and 8 levels.

Age Classification: We are also interested in finding out if the age of an employee has any effect on their leaving the company.

- We propose to have an evaluation based on the employee's age as a continuous variable and one based on what age bracket these employees belong to.
- We have 7 age brackets that we plan to work with. These are as follows:
    - 0-19 years
    - 20-29 years
    - 30-39 years
    - 40-49 years
    - 50-59 years
    - 60-69 years
    - 70 + years

Location classification: Since employees work from different locations with varying pay, we are hoping to find interesting metrics when we use these fields to deduce their effects on the churn rate.

- First, we plan to include 'Country' as one of the predicting instances to have a broader spectrum of the analysis. We have employees working from 35 different countries in our data.
- Second, we will use 'Location' to have a more specific evaluation which consists of employees from the 139 different work locations (as the column is described above).

Preliminary Data Conclusions

After completing the cleaning, we have a few different subsets of features we will run. With the high levels of redundancy among columns, it was hard to make some decisions about which columns to actually use. Instead of making those decisions, we decided to run models on different subsets of features and see which proved the most useful. Our final data set looks as follows:

What our data looks like now: We have 80,804 rows and 18 columns. We set the index names of each row to be the "GID_anonymized" column we were provided with so that the output of our models would be identifiable to specific employees. Note that this dataset includes everything for now as we have not yet made decisions regarding the NULL values in our Job Group/Job Level/Job Category columns and have not yet decided how to deal with the Compa Ratio information as described above.

Features that will be included in all models:

- Years in Service
- Gender
- Region
- Compa Ratio
- Count of Direct Reports
- Race/Ethnicity
- Education (whether or not employee chose to share their education information)
- Marital Status

Features that will be subsetted to see which fit best:

- Job Category & Job Level, Job Group & Job Level, Compensation Grade
- Age, Age Buckets
- Country, Location

In order to proceed with our model fitting, we will also have to do some pre-processing steps that we did not include in detail in the preceding report. This will include one hot encoding of our non-numeric columns and normalizing our numeric columns.

Model Performance Report

Data Preprocessing

In previous reports we described our data, its general shape and structure, and the methods we employed to clean the data. Following data cleaning, while preparing to train models, we found our data was not yet ready for model training. Essentially, we found three tasks were necessary before training and evaluating models. We called this phase Data Preprocessing and the tasks included were: Normalizing, Factorizing, and One-Hot Encoding.

First, we normalized our continuous variables (numeric variables that can take on a nearly-limitless number of values) by mapping each instance to $(x - \mu)/\sigma$. This step is necessary to prevent overweighting of continuous variables which are naturally larger than others. For example, *Age*, we would logically expect to be larger than certain other variables (e.g. *Years_in_Service_(Continuous_Service_Date)*). These discrepancies can distort model training weights if not normalized or scaled appropriately. The variables we normalized here were *Age*, *Compa_Ratio*, *Count_of_Direct_Reports*, and *Years_in_Service_(Continuous_Service_Date)*.

Second, we factorized the categorical variables where order is inherently informative. For example, assume we have the following categorical variables: *Postal_Code* and *College_Year*. *Postal_Code* is categorical, but order has no meaning. A code of 80210 does not inherently mean anything more or less than 80401. *College_Year*, on the other hand, contains ordinal meaning. Someone who is a 'Junior' is indeed closer to graduating than a 'Freshman'. Now, in our case, we determined the following variables were ordinal, and therefore had to be factorized: *Job_Level*, *Job_Group,* and *Age_-_Buckets*. To transform these variables into an order which the model can interpret and utilize (aka factorized), we converted their respective values from string (e.g. Age Bucket "20-29") into their implied ordinal position (0, 1, 2, etc.). *Job_Group* had a slightly less obvious order; we factorized it in the following way: 'Operators' : 0, 'Support' : 1, 'Operations Supervisors' : 2, 'Professional' : 3, 'Management' : 4, 'Executive' : 5.

Third, and finally, we transformed categorical variables without ordinal meaning into 1 (where the value is present) and 0 (where the value is not present), known as One-Hot Encoding. This meant any possible value for any of the following variables was transformed into a variable of its own (and if present a 1 was inserted for that record), the initial variable was then dropped. This is because, just as was the case with the previous step (factorizing), mathematical models cannot interpret values like "Operator" or "Professional". But, the model can interpret and mathematically employ values such as 1 and 0. The variables that we one-hot encoded are: *Job_Category_(Job_Class)*, *Gender*, *Region*, *Country*, *Management_Level_(Picklist_Label)*, *Compensation_Grade*, *Marital_Status*, *Location*, *Ethnicity*.

Finally, as mentioned in our data cleaning report, we reserved a few subsets of data to run on each model because so many of our variables captured the same information in varying degrees of specificity. We created 12 different data sets with the different rotating variables for age, job type, and location so as not to make upfront assumptions about which feature would fit the best while also avoiding multicollinearity. Below is the breakdown of each of the rotating variables included in the different variations of the models (in addition to the rest of our cleaned data):

1. Bucketed Age, Compensation Grade, Location
2. Age Continuous, Compensation Grade, Location
3. Bucketed Age, Compensation Grade, Country
4. Age Continuous, Compensation Grade, Country

5. Age Continuous, Job Category, Job Level, Country
6. Bucketed Age, Job Category, Job Level, Country
7. Bucketed Age, Job Category, Job Level, Location
8. Age Continuous, Job Category, Job Level, Location
9. Age Continuous, Job Group, Job Level, Country
10. Age Continuous, Job Group, Job Level, Location
11. Bucketed Age, Job Group, Job Level, Country
12. Bucketed Age, Job Group, Job Level, Location

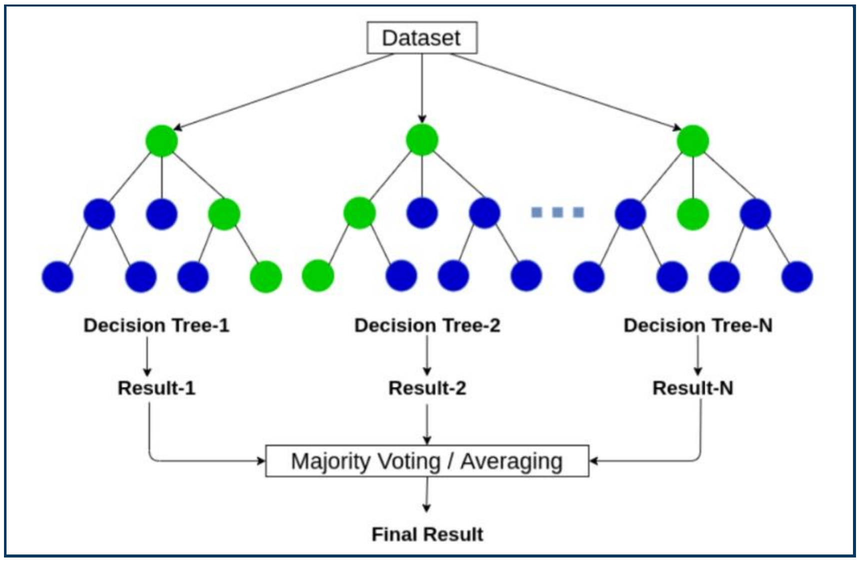
Data Modeling

Harpreet Kaur Setia - Random Forest

Introduction

Our team's main objective is to assist [Company] in predicting and identifying employees that might voluntarily leave the company based on their behavioral, demographical, and technical characteristics. I implemented the Random Forest algorithm for this requirement, a flexible, easy-to-use supervised machine learning algorithm that combines the outputs from numerous decision trees to a single outcome.

Random Forest is an ensemble learning model which outputs the class predicted by the most number of trees for classification tasks and returns the mean prediction from individual trees for regression tasks. Each node on these trees operates with a random subset of data features to calculate the output. The random forest then combines the results from these individual trees to formulate the final result, as shown in the figure below.



Random Forest: Majority Voting or Averaging

Random Forest uses the ensemble technique of Bagging (Bootstrap Aggregation) to choose a random subset of features for each node. Because of this feature selection, Random forest outperforms Decision tree results as it does not rely on a single feature selection. Random Forest also reduces overfitting on training data and higher accuracy on test data.

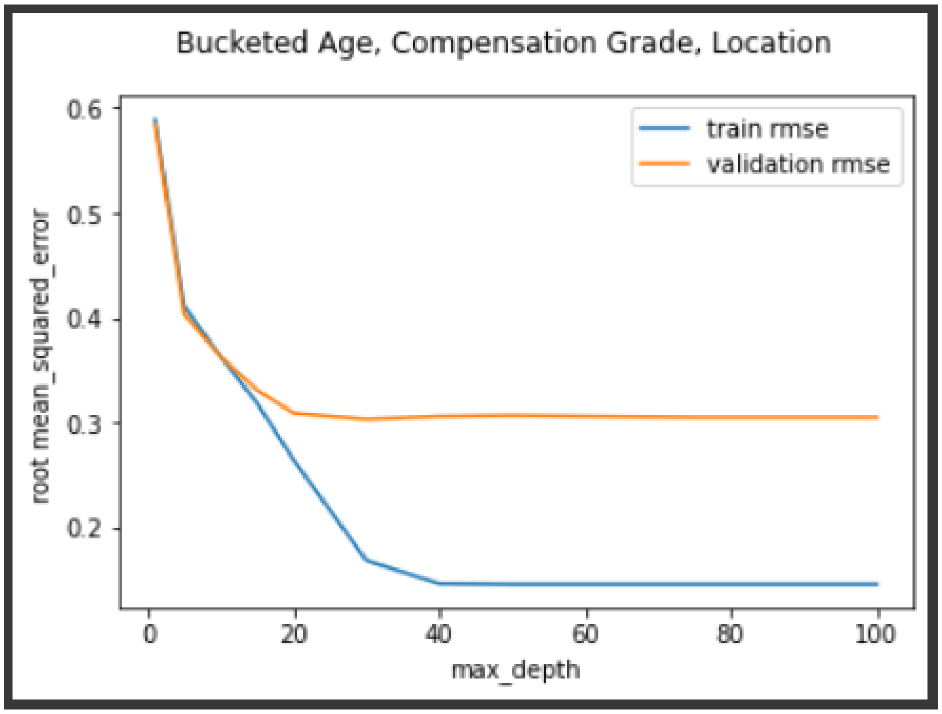
We can increase a Random Forest model's performance and prediction power by fine-tuning a few hyperparameters such as n_estimators (no. of trees), max_features (max no. of features considered for splitting a node), random_state (to tune the randomness of samples), and n_jobs (to define no. of processors assigned for a task).

I decided to implement Random Forests for this business problem, as they generate accurate predictions across a wide range of data with minor hyper-tuned configurations.

Model Performance: Development and Training

As mentioned above in the Data Preprocessing section, our team decided to implement our models on different subsets of data to capture the essential features and their impact on our target, whether an employee voluntarily churns or otherwise. I used 12 different subsets and divided each into train, validation, and test partitions to run my models by the ratio 60-20-20, respectively.

For each subset, I ran a RandomForestClassifier model for a series of max_depths and n_features. Once I had the train and validation mean squared errors using the classifier to predict the Status target, I plotted those values against the list of max_depth values that I initially used to train the model. This portrayal showed me how well my model asses the data based on different max_depth values. To calculate the best metric for the n_feature hyperparameter, I tested various categories such as the log of the number of columns, the square root of the number of columns, and the actual number of columns in each dataset.



Model Optimization

The next step was to use the best-tuned max_depth (ranging from 20-45 from individual subsets of data) and n_feature (square root value of the number of features in each subgroup) to run final models on the test data to calculate statistics on the accuracy, precision, and recall.

Below, you can see the overall test accuracy for each model on the 12 data subsets:

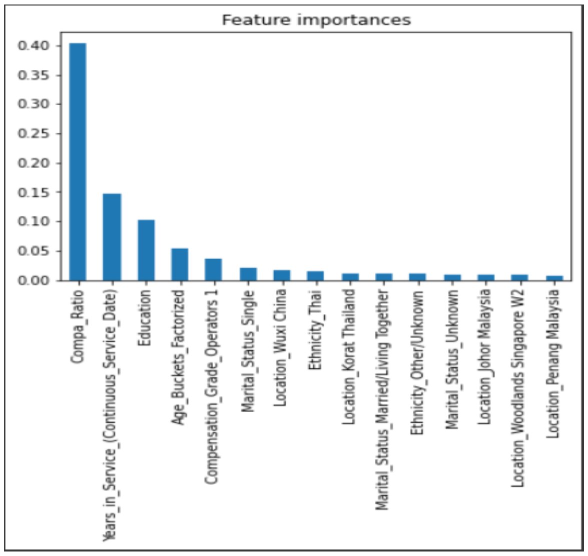
Model	Data Subset: Focussed Features	Max_Depth	N_Features = sqrt(len(columns))	N_Estimators	Random_State	Test Accuracy
1	Bucketed Age, Compensation Grade, Location	30	11	400	1234	0.903
2	Age Continuous, Compensation Grade, Location	30	11	300	1234	0.904
3	Bucketed Age, Compensation Grade, Country	23	11	200	1234	0.902
4	Age Continuous, Compensation Grade, Country	25	11	250	1234	0.903
5	Age Continuous, Job Category, Job Level, Country	22	8	200	1234	0.904
6	Bucketed Age, Job Category, Job Level, Country	20	8	250	1234	0.903
7	Bucketed Age, Job Category, Job Level, Location	40	8	250	1234	0.904
8	Age Continuous, Job Category, Job Level, Location	20	8	300	1234	0.906
9	Age Continuous, Job Group, Job Level, Country	30	9	300	1234	0.901
10	Age Continuous, Job Group, Job Level, Location	45	12	300	1234	0.9
11	Bucketed Age, Job Group, Job Level, Country	30	9	300	1234	0.902
12	Bucketed Age, Job Group, Job Level, Location	30	12	300	1234	0.899

As shown, all the models have performed somewhat similarly on the data, with the average test accuracy around 90.2%. The subset consisting of features Age_Buvekts, Job_Group, Job_Level, and location attained the lowest accuracy at 89.9%. Meanwhile, the highest accuracy at 90.6% was for the model that ran on Age Continuous, Job_Category, Job_Level, and Location.

Model: Operators vs. All Other Employees

In addition to those mentioned above 12 different models, our team was also interested in analyzing the records for Operators as they constitute almost 65% of the entire employee categories. I ran a new Random Forest algorithm on all the features of two new subgroups: Operators vs. all other employees.
I followed the same steps to find the best-tuned hyperparameters for these two models and analyzed their accuracies. The model I ran, the Operators dataset, gave me a higher accuracy (93%) than all other employee's datasets (83%).

Model: without Compa-Ratio



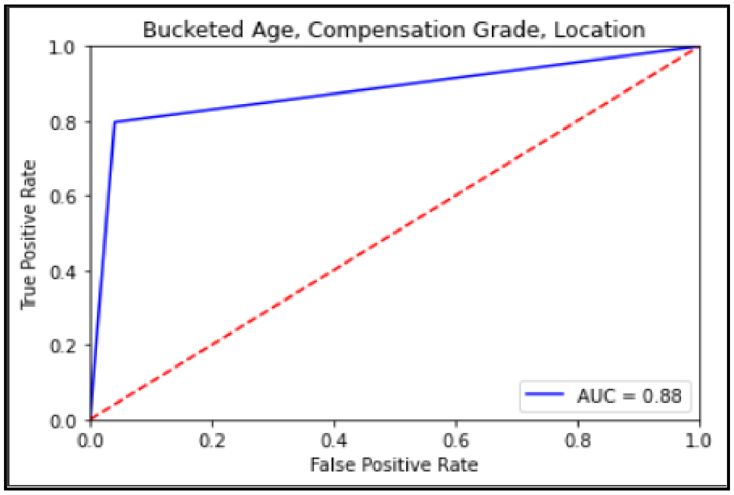
In addition to model accuracies, I also wanted to analyze what features were most relevant to my models. For this, I examined the feature importance for the given dataset.

As exhibited in the plot on the right, it is evident that my model is highly dependent on the Compa Ratio variable. Thus, I re-ran my model on the dataset without compa ratio to remove this high dependency.

With the new bag of features, I observed that the test accuracy reduced to 83%, with an AUC measure of 81%.

These metrics clearly show that the target value 'Status' is learned highly based on compa_ratio. Now, this would have been a typical scenario, but since compa_ratios are getting modified due to inflation and other factors, the question arises whether we can consider keeping compa_ratio as one of our regressors. We will be making this point to [Company] during our recommendations.

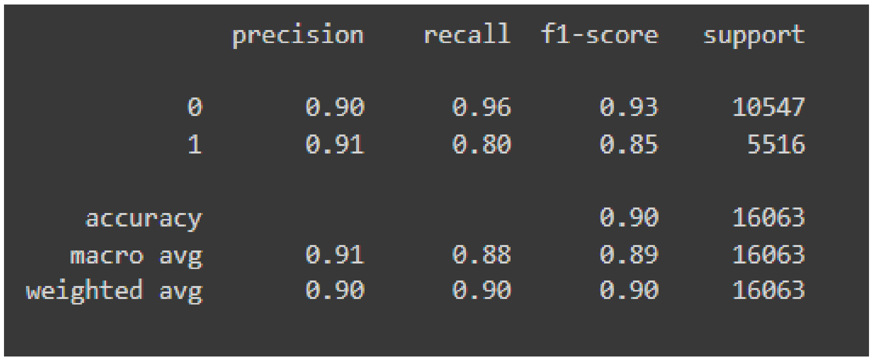
Model Quality (Results)



As mentioned above, I attained an average test accuracy of 90.2% for my models. This score indicates that my models could identify relationships between model predictors and the target value quite well. In addition, looking at the feature importance plot, we can see that the higher the importance weight value, the higher the chances of that employee voluntarily leaving the company. For example, such a high value for compa_ratio indicates that its weight highly affects an employee's decision-making to leave the company.

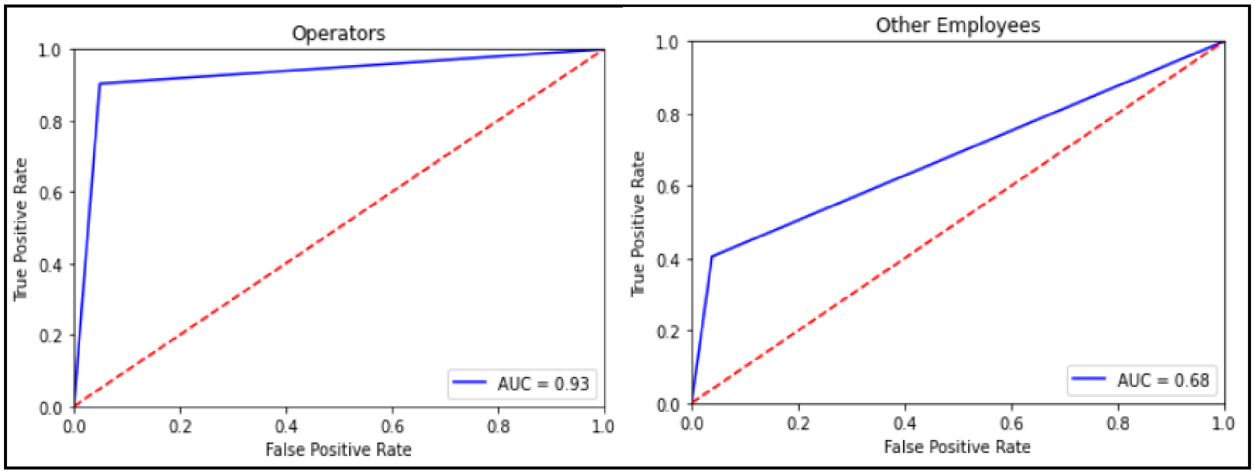
In addition to the above metrics, I also analyzed the Area Under the Curve score plots and classification reports for all my models to learn how well my models can predict actual positive churned employees.

For my first model focusing on Bucketed Age, Compensation Grade, Location, my model attained 88% of the area under the curve score, i.e., my model was able to correctly distinguish between positive and negative classes (voluntary termination vs. other reasons) 88% of the time.



```
              precision    recall  f1-score   support

           0       0.90      0.96      0.93     10547
           1       0.91      0.80      0.85      5516

    accuracy                           0.90     16063
   macro avg       0.91      0.88      0.89     16063
weighted avg       0.90      0.90      0.90     16063
```

The classification report for this model showcases a precision of 91%, indicating that the model correctly predicted 91% times if an employee would churn. Out of these predictions, 80% of those employees churned. These high metrics suggest that the implemented model can generalize well on new data and predict solid results. Similar metrics were evident from all other 11 dataset models as well. For my models on Operators. Other Employees, I attained a very high accuracy of 93% on the test data, with an AUC score of 93%. Since we had a considerable amount of records for Operators, my model learned better and gave better scores.

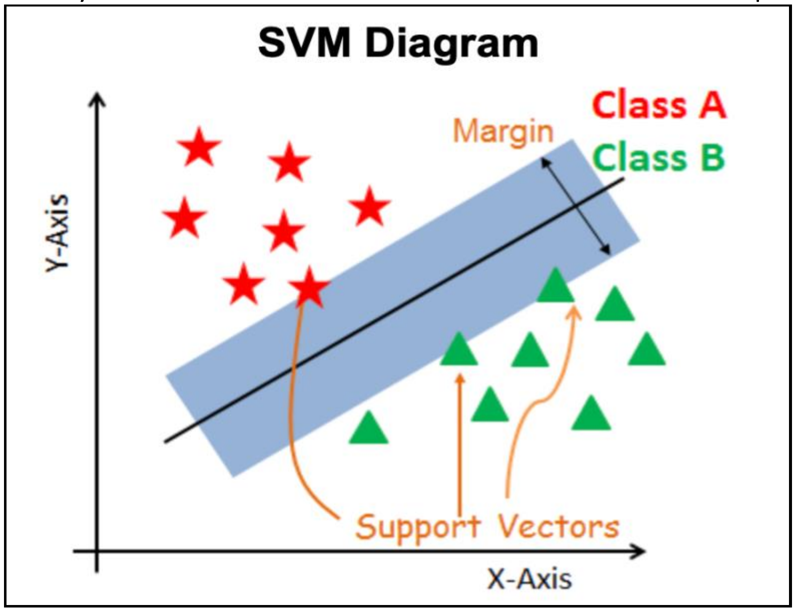


But the model on other employees gave contrasting results. Since the data size was reduced by almost a half, the model could only distinguish between classes 68% of the time. This AUC score indicates that my model needs more data for employees working in all other job categories to attain better insights.

Sara Tanabe - Support Vector Machine

Introduction

When considering appropriate models to utilize to help [Company] identify individuals that are likely to voluntarily terminate their employment with the company, we knew we needed to use a classification model. Support Vector Machine, or SVM, is a classification model that separates data into different classes using a best-fit hyperplane. We felt SVM might be a great contender for this project. The goal of an SVM model is to fit the data with a hyperplane that maximizes the margin, or space between the two classes. In the diagram created by Avinash Navlani for his DataCamp tutorial "Support Vector Machines with Scikit-learn" you can see the basic idea of what a SVM model is attempting to accomplish.



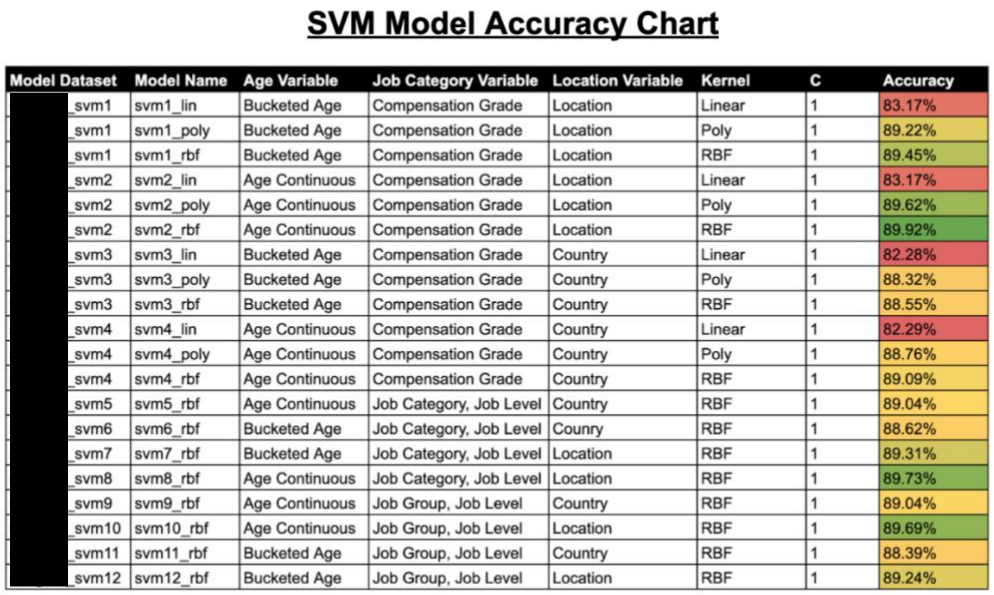
SVM's can utilize a few different types of hyperplanes for class segmentation. While the most basic form of a hyperplane is linear (as shown in the diagram), not all data is accurately separable in this way. SVM models have the ability to separate data using polynomial separation, which can separate data using curved or non-linear planes, or radial basis function separation, which can utilize infinite dimensional space to map data clases. This is why Support Vector Machine learning can be extremely useful, and in some cases, superior to other models.

Model Performance: Development and Training

Building an SVM model using python is straightforward and efficient using the scikit learn SVM package. In order to train the data, I divided up the data into a training and a testing set with an 80/20 split, respectively. For the first four models, I chose to run each model on all three kernels (linear, polynomial, and RBF) so that I could get a good baseline of which kernel would create the highest accuracy for the [Company] dataset. Through this, it became apparent that the RBF kernel would be the best fit for our model creation, and moving forward, I only created the models using the RBF kernel. Below you can find a summary of the different models and each accuracy associated with the model.

As you can see, the RBF model utilizing the dataset [Company]_svm2 (which included rotating variables

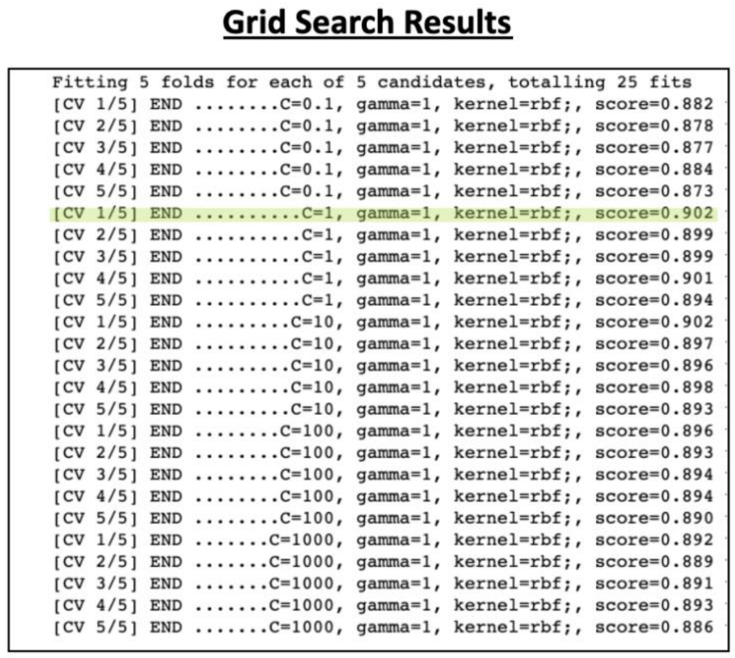
Age Continuous, Compensation Grade, and Location) yielded the highest accuracy of 89.92% on the baseline model.



SVM Model Accuracy Chart

Model Dataset	Model Name	Age Variable	Job Category Variable	Location Variable	Kernel	C	Accuracy
[redacted]_svm1	svm1_lin	Bucketed Age	Compensation Grade	Location	Linear	1	83.17%
[redacted]_svm1	svm1_poly	Bucketed Age	Compensation Grade	Location	Poly	1	89.22%
[redacted]_svm1	svm1_rbf	Bucketed Age	Compensation Grade	Location	RBF	1	89.45%
[redacted]_svm2	svm2_lin	Age Continuous	Compensation Grade	Location	Linear	1	83.17%
[redacted]_svm2	svm2_poly	Age Continuous	Compensation Grade	Location	Poly	1	89.62%
[redacted]_svm2	svm2_rbf	Age Continuous	Compensation Grade	Location	RBF	1	89.92%
[redacted]_svm3	svm3_lin	Bucketed Age	Compensation Grade	Country	Linear	1	82.28%
[redacted]_svm3	svm3_poly	Bucketed Age	Compensation Grade	Country	Poly	1	88.32%
[redacted]_svm3	svm3_rbf	Bucketed Age	Compensation Grade	Country	RBF	1	88.55%
[redacted]_svm4	svm4_lin	Age Continuous	Compensation Grade	Country	Linear	1	82.29%
[redacted]_svm4	svm4_poly	Age Continuous	Compensation Grade	Country	Poly	1	88.76%
[redacted]_svm4	svm4_rbf	Age Continuous	Compensation Grade	Country	RBF	1	89.09%
[redacted]_svm5	svm5_rbf	Age Continuous	Job Category, Job Level	Country	RBF	1	89.04%
[redacted]_svm6	svm6_rbf	Bucketed Age	Job Category, Job Level	Counry	RBF	1	88.62%
[redacted]_svm7	svm7_rbf	Bucketed Age	Job Category, Job Level	Location	RBF	1	89.31%
[redacted]_svm8	svm8_rbf	Age Continuous	Job Category, Job Level	Location	RBF	1	89.73%
[redacted]_svm9	svm9_rbf	Age Continuous	Job Group, Job Level	Country	RBF	1	89.04%
[redacted]_svm10	svm10_rbf	Age Continuous	Job Group, Job Level	Location	RBF	1	89.69%
[redacted]_svm11	svm11_rbf	Bucketed Age	Job Group, Job Level	Country	RBF	1	88.39%
[redacted]_svm12	svm12_rbf	Bucketed Age	Job Group, Job Level	Location	RBF	1	89.24%

Model Optimization
Despite already having a high accuracy of around 89%, I decided to move forward with tuning the svm2_rbf model to see if we could change the gamma or the C values to increase accuracy even further. To do this efficiently, I used the GridSearch scikit learn package to cycle through testing different C and gamma values with the model. Based on the findings from the Grid Search results, I decided to build the final SVM model with a kernel of RBF, a gamma of 1, and a C of 1.



Grid Search Results

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
[CV 1/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.882
[CV 2/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.878
[CV 3/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.877
[CV 4/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.884
[CV 5/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.873
[CV 1/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.902
[CV 2/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.899
[CV 3/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.899
[CV 4/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.901
[CV 5/5] END ..........C=1, gamma=1, kernel=rbf;, score=0.894
[CV 1/5] END .........C=10, gamma=1, kernel=rbf;, score=0.902
[CV 2/5] END .........C=10, gamma=1, kernel=rbf;, score=0.897
[CV 3/5] END .........C=10, gamma=1, kernel=rbf;, score=0.896
[CV 4/5] END .........C=10, gamma=1, kernel=rbf;, score=0.898
[CV 5/5] END .........C=10, gamma=1, kernel=rbf;, score=0.893
[CV 1/5] END ........C=100, gamma=1, kernel=rbf;, score=0.896
[CV 2/5] END ........C=100, gamma=1, kernel=rbf;, score=0.893
[CV 3/5] END ........C=100, gamma=1, kernel=rbf;, score=0.894
[CV 4/5] END ........C=100, gamma=1, kernel=rbf;, score=0.894
[CV 5/5] END ........C=100, gamma=1, kernel=rbf;, score=0.890
[CV 1/5] END .......C=1000, gamma=1, kernel=rbf;, score=0.892
[CV 2/5] END .......C=1000, gamma=1, kernel=rbf;, score=0.889
[CV 3/5] END .......C=1000, gamma=1, kernel=rbf;, score=0.891
[CV 4/5] END .......C=1000, gamma=1, kernel=rbf;, score=0.893
[CV 5/5] END .......C=1000, gamma=1, kernel=rbf;, score=0.886
```

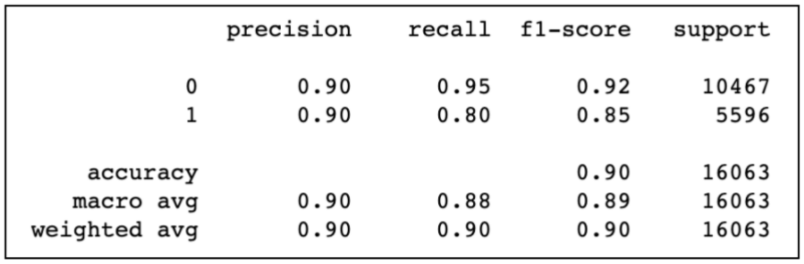
Model Quality - Results

The optimized version of the model resulted with a 90.24% accuracy rate. In the screenshot below, you can see the classification report, which displays the statistics about precision and recall. With a precision on the “churn” outcome of .9 and a recall on the “churn” outcome of .80, this is a reasonable model to utilize for this purpose. To further support this conclusion, I have also provided a visual of the confusion matrix, which displays the number of false positives and false negatives that the model predicted incorrectly. We will take the information in the confusion matrix into consideration for our economic analysis as it will be important to determine if what would be more costly, False Positives (predicting someone will churn when they actually will not), or False Negatives (not predicting someone will churn when they actually do).

Job Specific Models

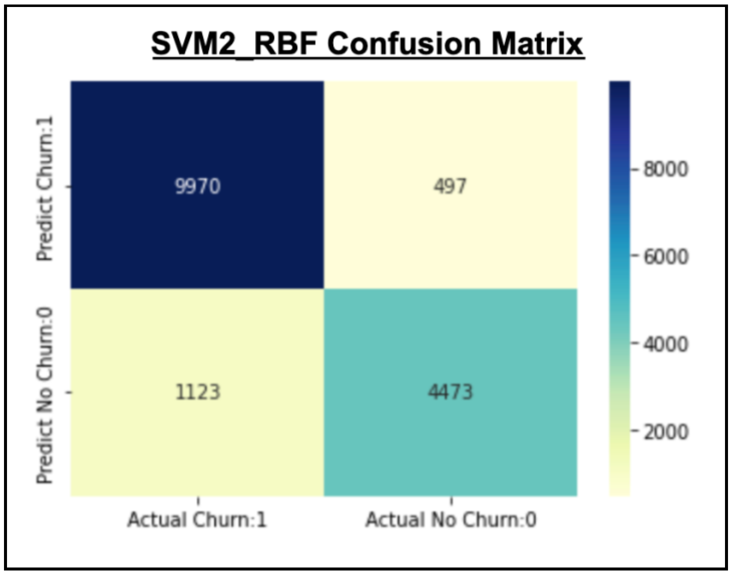
Operators make up almost 65% of the employee data in the [Company] dataset. With such a vast amount of data attributed to one job category, we felt that it might be important to separate our models to distinguish between operators and all other employees.

SVM2_RBF Model Classification Report

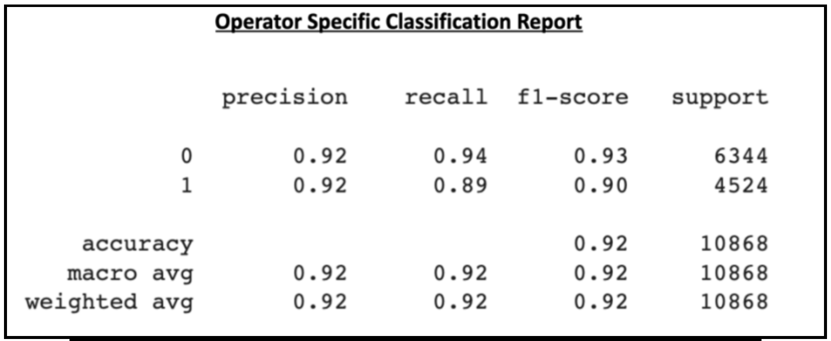


```
              precision    recall  f1-score   support

           0       0.90      0.95      0.92     10467
           1       0.90      0.80      0.85      5596

    accuracy                           0.90     16063
   macro avg       0.90      0.88      0.89     16063
weighted avg       0.90      0.90      0.90     16063
```

Precision - The ratio of correct positive predictions to the total predicted positives.
Recall - The ratio of correct positive predictions to the total positive examples.
-https://www.kdnuggets.com/2020/01/guide-precision-recall-confusion-matrix.html



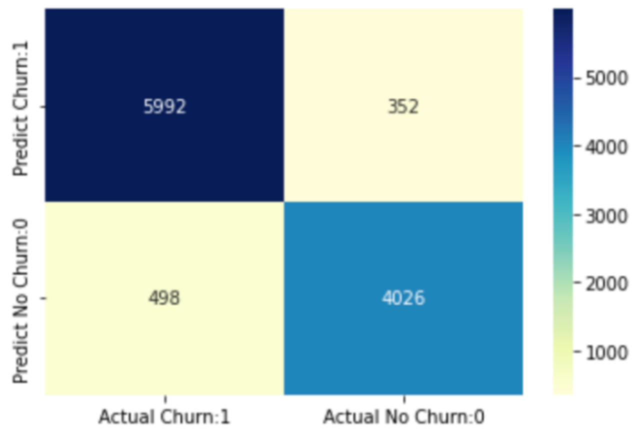
After dividing the data into two different datasets, one for operators and one for the other job categories, I was able to run the same model (SVM with RBF kernel, gamma=1, and C=1) across the operator specific data. This resulted in higher accuracy, precision, and recall scores as seen in the Classification report below. I have also provided the confusion matrix for these operator specific churn predictions.



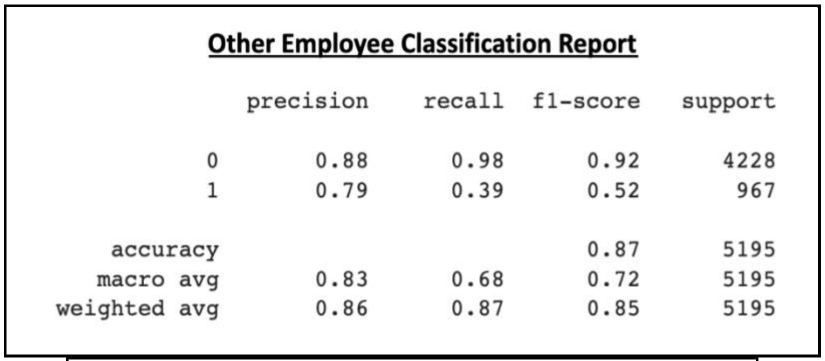
Operator Specific Classification Report

```
              precision    recall  f1-score   support

           0       0.92      0.94      0.93      6344
           1       0.92      0.89      0.90      4524

    accuracy                           0.92     10868
   macro avg       0.92      0.92      0.92     10868
weighted avg       0.92      0.92      0.92     10868
```

Operator Specific Confusion Matrix

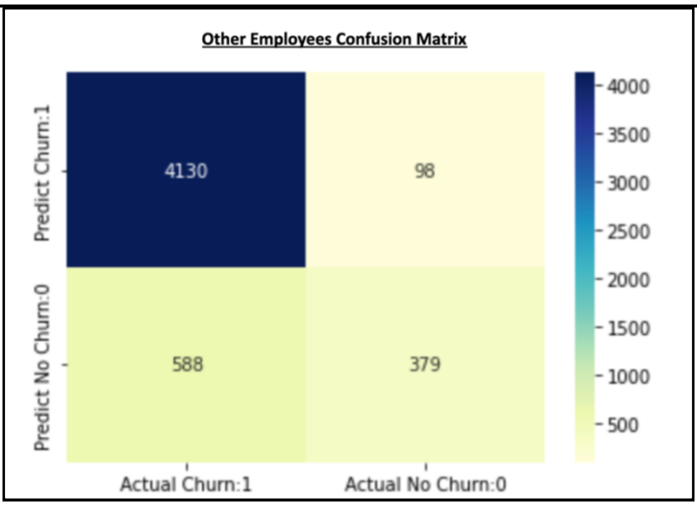


I then repeated the same steps for the other employees. While ideally, I would be able to create models for each individual job category, there was not enough data for management level and up employees to be able to train a model. You can see in the below classification report and confusion matrix the results for this were not as great, bringing accuracy down to around 87%. Most notably, the recall for employees who are predicted to churn dropped significantly to around 39%.



Other Employee Classification Report

```
              precision    recall  f1-score   support

           0       0.88      0.98      0.92      4228
           1       0.79      0.39      0.52       967

    accuracy                           0.87      5195
   macro avg       0.83      0.68      0.72      5195
weighted avg       0.86      0.87      0.85      5195
```



Model Limitations

While this model produced great results, it is important to evaluate the limitations and challenges that could potentially affect the overall usefulness and reliability of the results. Three of the main limitations are highlighted below:

Model Run Time

The biggest flaw with this model is that it took over 30 minutes to run the code for fitting the SVM classifier to the training data on the entire dataset. This is likely due to the vast number of dummy variables that I needed to create in order to include the many categorical variables in this type of model. Based on the run time, this model is unrealistic and likely not feasible to utilize in the long term. It was, however, a little bit faster to run on the job category specific models (~10 min); however, these models did not yield as consistent of results.

Lack of Visibility Towards Variable Importance

It is possible that we could decrease the processing time by decreasing the number of variables in the model. However, one of the limitations with utilizing an RBF kernel for this model is that unlike the linear kernel, it is impossible to assign a weight value to each variable, making it hard to find the ranked importance of the different variables. This can be an important step in creating a model as it can help you understand if too much importance is being placed on one particular variable that could skew results, or if there are variables that are not impacting the model at all that can be eliminated to make the model more efficient. Without the ability to understand the importance of each variable, it is difficult to provide actionable insight about what variables might impact whether a person might voluntarily terminate employment or not.

Data Issues

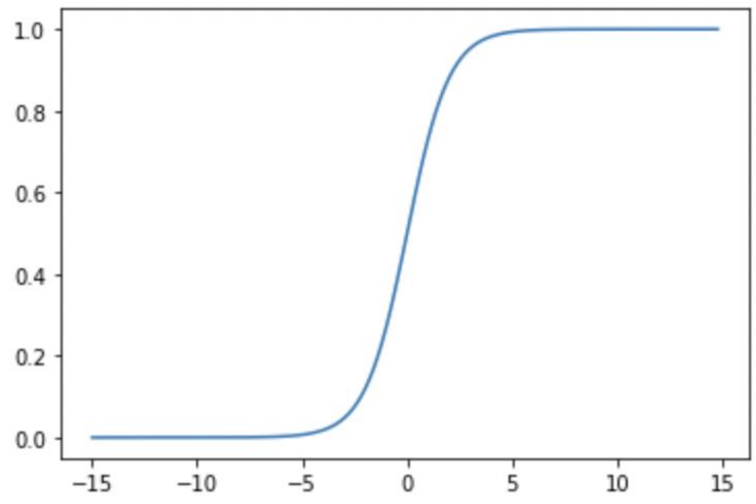
One of the main issues that we grappled with as a group was the idea that compa_ratio is updated over time. This means that as the median salary for employees in a job group increases with inflation, employees who had left some time ago drop down in compa_ratio in comparison to where they were when they terminated employment. This likely skews the data and makes it seem as though employees who left had a lower salary in comparison to their counterparts than they actually did. As compa_ratio is likely an important factor in deciding whether an employee leaves or not, I expect this is impacting this model's results greatly.

Additionally, the data lacks some important statistics that we feel could improve the model. For example, the data does not include any information about when the employee's last promotion was. From a logical perspective, career growth is likely an important factor that most employees consider when deciding whether or not to leave a job. It would be interesting to include this variable in our model to see how it impacts results.

Talia Zalesne - Logistic Regression

Introduction

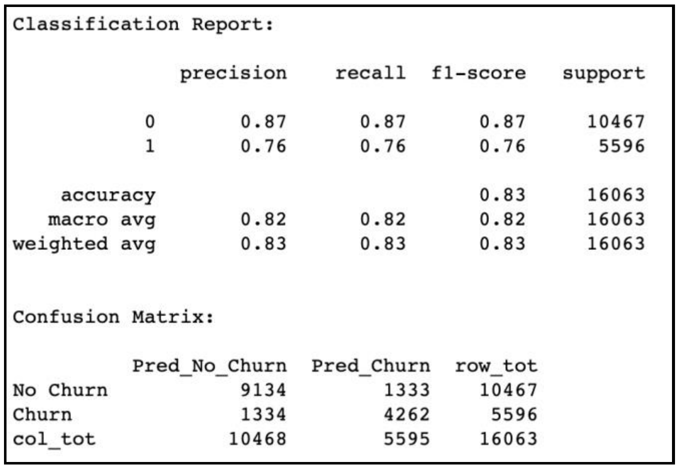
Logistic Regression is a type of binary classification model that learns weights then uses log odds to output the probability of an event occurring - it outputs a value between 0 and 1 where 0 indicates no possibility of occurrence and 1 indicates total certainty of occurrence. Based on some threshold (typically 0.5), the logistic regression model then makes predictions of whether or not the event in question will occur based on it's computed probability. Logistic regression utilizes the sigmoid function to map computed values to probabilities. This allows the outputted probability to only take values between 0 and 1 (which is reasonable in probability analysis) and also helps logistic regression be fairly resistant to the effect of outliers.



The overall goal of this project is to successfully predict whether an employee is going to churn. A false positive in this case would be incorrectly predicting someone will leave while a false negative would be incorrectly predicting someone will not leave. Though we do not yet have a full scope of the financial impact information from [Company], I infer that a false negative will have a much larger impact than a false positive. For that reason, I chose to focus on optimizing both accuracy and recall. In context, recall encompasses the statistic: "out of every employee at [Company] who voluntarily churns, how many did we predict would churn."

Model Performance: Training

I began by doing an 80/20 training/test split on my data and running a simple logistic regression model on each subset of data we had set aside (12 models in total). For this initial round of training, I used the "SAG" (Stochastic Average Gradient) solver because it tends to converge faster on larger datasets. Although the data we are working with is not too big, the number of features became decently large as we got dummies for more and more variables. By default, these base models also employed 'l2' regularization (ridge regression) with a C-value of 1.0. Regularization is a method used in regression to help prevent overfitting and the C-value controls the strength of the regularization - where a smaller C corresponds to stronger regularization and vice versa.

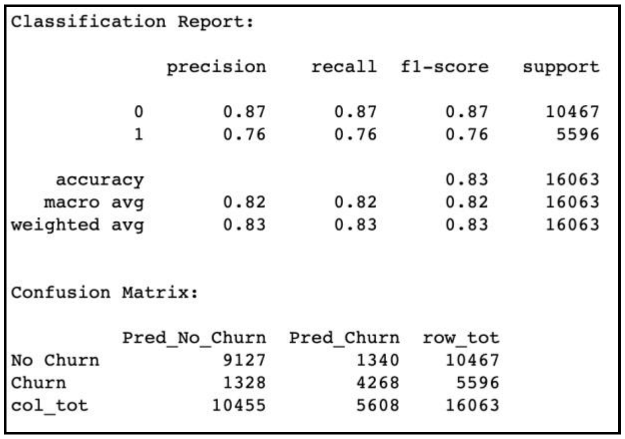


```
Classification Report:

              precision    recall  f1-score   support

           0       0.87      0.87      0.87     10467
           1       0.76      0.76      0.76      5596

    accuracy                           0.83     16063
   macro avg       0.82      0.82      0.82     16063
weighted avg       0.83      0.83      0.83     16063


Confusion Matrix:

         Pred_No_Churn  Pred_Churn  row_tot
No Churn          9134        1333    10467
Churn             1334        4262     5596
col_tot          10468        5595    16063
```

Overall, I found pretty comparable results among each model indicating that it did not make a huge computational difference whether we used age buckets vs. continuous age features, location vs. country features, or compensation grade vs. job group & job level vs. job category & job level features. After inspecting both the classification report and confusion matrices for each model, I decided that the model utilizing age as a continuous predictor, compensation grade, and location had slightly better performance than its counterparts. To the right, we can see that this model performed with an overall accuracy of 83% and a recall rate of 76% with only 1,334 false negatives.

My next step was to attempt to improve this base model. I began by constructing a grid containing all of the possible solvers, the l2 penalty (because this is the only penalty compatible with all of the solvers) and C values ranging from 100 to .001 (decreasing by a factor of 10) and using GridSearchCV to output which parameters fit the best. I did this twice, once with the goal to maximize recall and once to maximize accuracy.

Model to optimize recall:



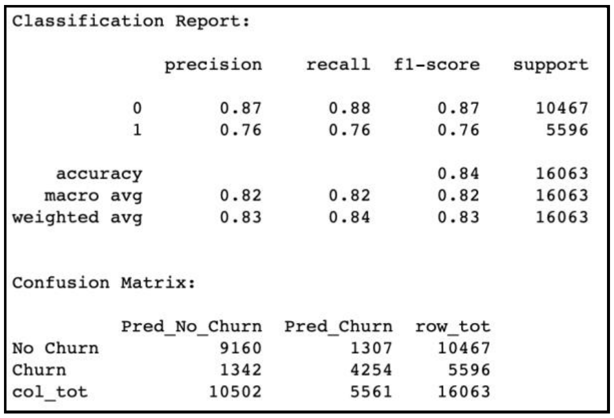
```
Classification Report:

              precision    recall  f1-score   support

           0       0.87      0.87      0.87     10467
           1       0.76      0.76      0.76      5596

    accuracy                           0.83     16063
   macro avg       0.82      0.82      0.82     16063
weighted avg       0.83      0.83      0.83     16063


Confusion Matrix:

         Pred_No_Churn  Pred_Churn  row_tot
No Churn          9127        1340    10467
Churn             1328        4268     5596
col_tot          10455        5608    16063
```

This model employed the 'lbfgs' (Limited-memory Broyden–Fletcher–Goldfarb –Shanno) solver with C = 100. This solver typically isn't super fast with larger datasets as it only stores the last few updates to save memory. The classification report and confusion matrix are shown. This model is very comparable to the original model, with a very tiny improvement in minimizing false negatives.

Model to optimize accuracy:

This model employed the ‘saga’ solver with C = .001. The classification report and confusion matrix are shown. The ‘saga’ solver is very similar to the ‘sag’ solver I originally used. In fact, it is just an extension of that original solver in that it also allows for other regularization methods to be used. This model was again pretty comparable to the original model, but it was able to increase accuracy to 84% at the cost of reducing recall to 72%.

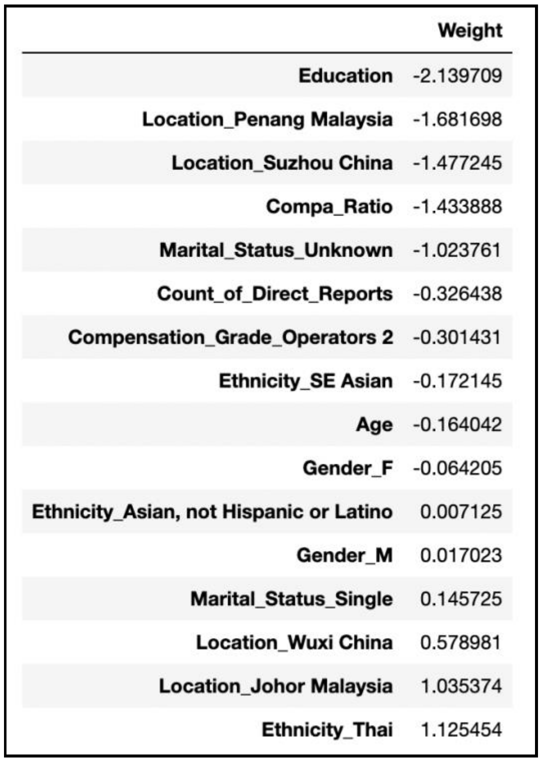
Since the ‘saga’ solver allows for other regularization methods, I decided to experiment with those parameters. Overall, I was able to determine that the best model employed the ‘saga’ solver, ‘l1’ regularization (lasso regression), with a C-value of 0.01. As shown to the right, this model allowed me to maintain my 84% accuracy and my 76% recall.



```
Classification Report:

              precision    recall  f1-score   support

           0       0.87      0.88      0.87     10467
           1       0.76      0.76      0.76      5596

    accuracy                           0.84     16063
   macro avg       0.82      0.82      0.82     16063
weighted avg       0.83      0.84      0.83     16063


Confusion Matrix:

          Pred_No_Churn  Pred_Churn  row_tot
No Churn           9160        1307    10467
Churn              1342        4254     5596
col_tot           10502        5561    16063
```

My last experimentation was to use Principal Component Analysis in conjunction with Logistic Regression (and the same parameters outlined above). PCA is a method of reducing the dimensionality of the data without losing any information. After testing a few different values for the number of components, I determined that the best was 8 and this was able to output a model with 85% accuracy. Though this was a definite accuracy improvement, the recall of this model took a hit so I ultimately chose to stick with the logistic regression without PCA as my final model.

Model Quality (Results)

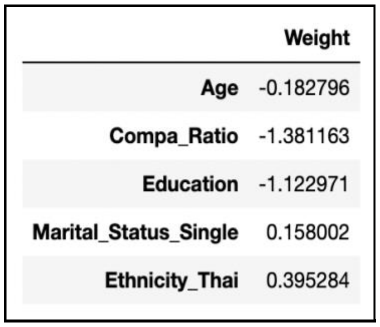


	Weight
Education	-2.139709
Location_Penang Malaysia	-1.681698
Location_Suzhou China	-1.477245
Compa_Ratio	-1.433888
Marital_Status_Unknown	-1.023761
Count_of_Direct_Reports	-0.326438
Compensation_Grade_Operators 2	-0.301431
Ethnicity_SE Asian	-0.172145
Age	-0.164042
Gender_F	-0.064205
Ethnicity_Asian, not Hispanic or Latino	0.007125
Gender_M	0.017023
Marital_Status_Single	0.145725
Location_Wuxi China	0.578981
Location_Johor Malaysia	1.035374
Ethnicity_Thai	1.125454

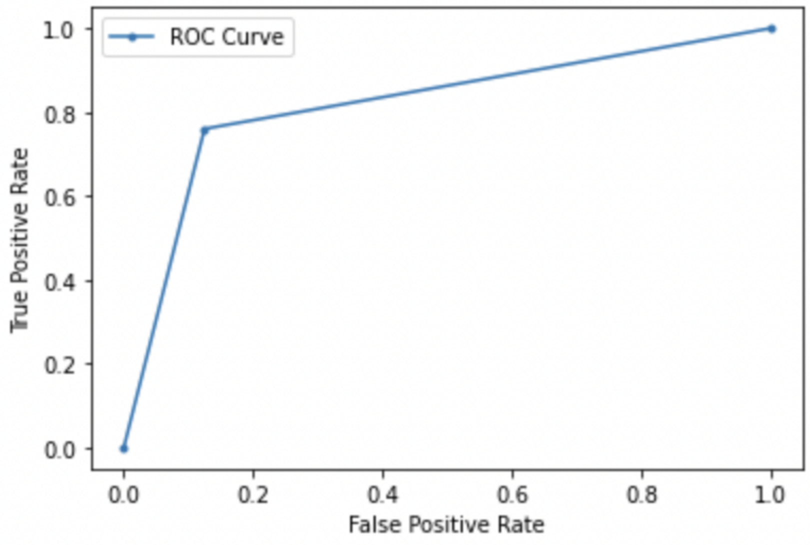
As stated above, my final model was able to attain 84% accuracy on the testing data. In addition, since I ended up using lasso regression for my regularization, not all of the features were utilized - lasso allows the less important features to actually attain zero weights whereas ridge regression lets the weights get smaller and smaller but never actually hit zero. A table of the nonzero weights along with their corresponding features is shown to the right. This can be interpreted as which features were most important to the analysis and how important they were. Any weights with a negative value indicate that a higher value for the corresponding feature makes an employee less likely to churn. For example, we can interpret from our model that a larger number of direct reports - people working directly under said employee - is correlated with a smaller likelihood of voluntarily leaving [Company]. Further, larger magnitudes indicate that the corresponding feature has a bigger effect on our target - however these cannot be interpreted linearly in a logistic regression. Alot of these results are as expected however I do find it concerning that the "unknown" marital status has such a high importance in this model. It is clear that there is some other factor leading this feature to feign importance.

As an aside, out of curiosity I ran the model one more time with a higher regularization weight (C = 0.001) to see which features still made the cut. The table to the right indicates that the most important features in this regression model are age (where older employees are less likely to churn), compa ratio (where employees making a higher salary as compared to the midpoint are less likely to churn), education (where employees who choose to divulge their education background are less likely to churn), marital status single (where single employees are more likely to churn), and ethnicity Thai (where Thai employees are more likely to churn). I would like to make a big disclaimer here that I do not think this is actually indicating that Thai employees are more likely to churn - this is more likely a case of multicollinearity in the model. The majority of our dataset is comprised of Thai employees and almost all of these hold the operator position. For those reasons, it makes a lot of sense to see this feature shine through as such an important predictor when, in reality, this is likely simply capturing that operators are more likely to churn and most of the data we have on that are Thai employees.

```
AUC Score: 0.8176586061383703
```



	Weight
Age	-0.182796
Compa_Ratio	-1.381163
Education	-1.122971
Marital_Status_Single	0.158002
Ethnicity_Thai	0.395284

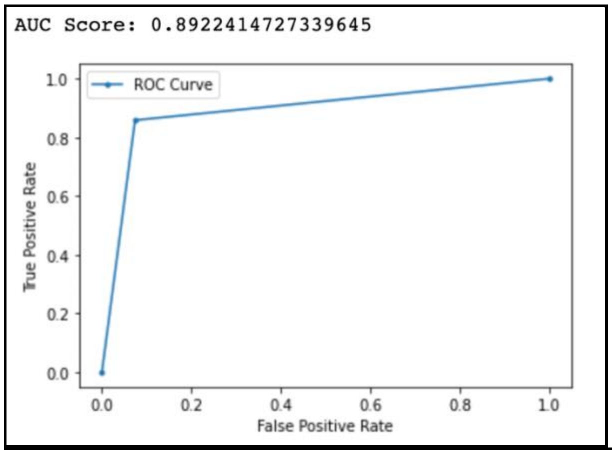


Finally, to further evaluate my model, I calculated the AUC score and the corresponding ROC curve. A higher AUC score indicates that the model is able to distinguish between the positive and negative classes very well. Visually, we are looking for the elbow on the ROC graph (which plots the true positive rate vs. the false positive rate) to be as close as possible to the upper left corner. I am pretty pleased to have achieved an AUC of 0.82 using logistic regression.

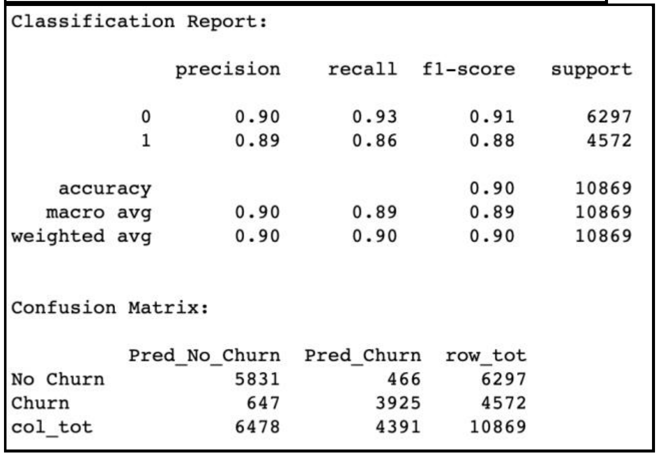
Model Split By Job Group

Based on the results of my general logistic regression and the knowledge that [Company] suffers a very different financial impact from the churn of operators vs. the churn of other job groups, I created two new data frames - one containing only the operators and one containing everyone else (I did consider running this analysis on the executive group alone but there are only 140 instances of executives in this dataset which is not enough to produce meaningful analysis). For both data frames, I started by using the features I had originally found to be the best (continuous age, compensation grade, location) and then immediately performed GridSearchCV to find the best parameters to run the logistic regression with. My best results are summarized as follows.

Operators:



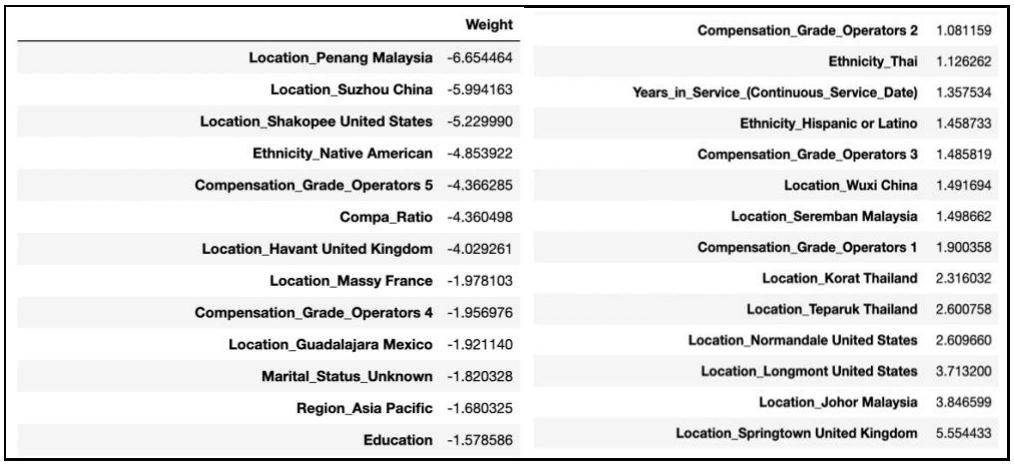
```
AUC Score: 0.8922414727339645
```



```
Classification Report:

              precision    recall  f1-score   support

           0       0.90      0.93      0.91      6297
           1       0.89      0.86      0.88      4572

    accuracy                           0.90     10869
   macro avg       0.90      0.89      0.89     10869
weighted avg       0.90      0.90      0.90     10869


Confusion Matrix:

          Pred_No_Churn  Pred_Churn  row_tot
No Churn           5831         466     6297
Churn               647        3925     4572
col_tot            6478        4391    10869
```

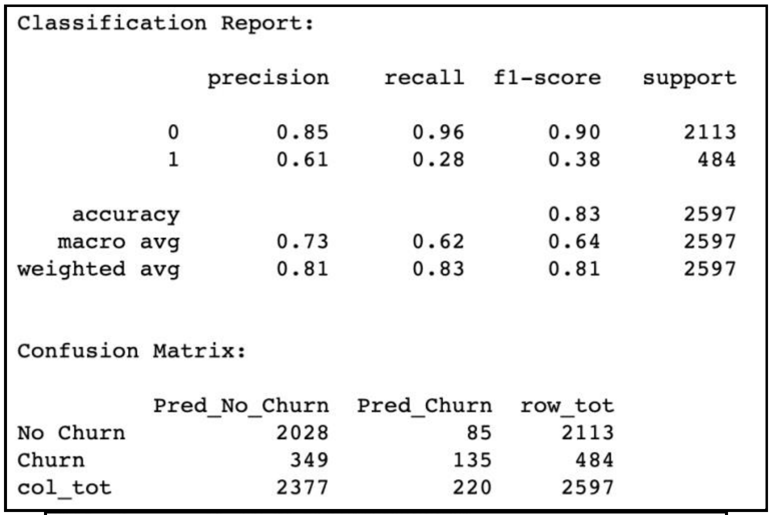
I was able to attain really good results on the operator job class alone. Not only did the accuracy, recall, and AUC score drastically increase but the relative importance of the features also increased - meaning there were alot more features with meaningful weights.

Since this model did not use lasso regularization, there weren't any zero weights but I was able to capture which features had the biggest positive and negative impact:

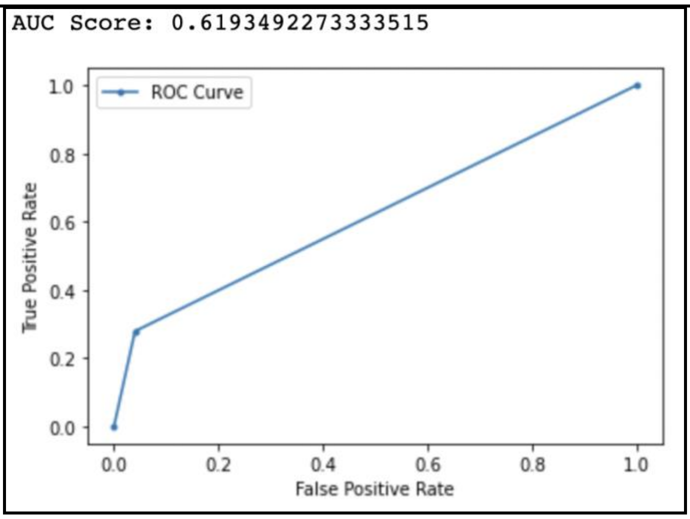


	Weight
Location_Penang Malaysia	-6.654464
Location_Suzhou China	-5.994163
Location_Shakopee United States	-5.229990
Ethnicity_Native American	-4.853922
Compensation_Grade_Operators 5	-4.366285
Compa_Ratio	-4.360498
Location_Havant United Kingdom	-4.029261
Location_Massy France	-1.978103
Compensation_Grade_Operators 4	-1.956976
Location_Guadalajara Mexico	-1.921140
Marital_Status_Unknown	-1.820328
Region_Asia Pacific	-1.680325
Education	-1.578586
Compensation_Grade_Operators 2	1.081159
Ethnicity_Thai	1.126262
Years_in_Service_(Continuous_Service_Date)	1.357534
Ethnicity_Hispanic or Latino	1.458733
Compensation_Grade_Operators 3	1.485819
Location_Wuxi China	1.491694
Location_Seremban Malaysia	1.498662
Compensation_Grade_Operators 1	1.900358
Location_Korat Thailand	2.316032
Location_Teparuk Thailand	2.600758
Location_Normandale United States	2.609660
Location_Longmont United States	3.713200
Location_Johor Malaysia	3.846599
Location_Springtown United Kingdom	5.554433

Other Job Groups:



```
Classification Report:

              precision    recall  f1-score   support

           0       0.85      0.96      0.90      2113
           1       0.61      0.28      0.38       484

    accuracy                           0.83      2597
   macro avg       0.73      0.62      0.64      2597
weighted avg       0.81      0.83      0.81      2597


Confusion Matrix:

         Pred_No_Churn  Pred_Churn  row_tot
No Churn          2028          85     2113
Churn              349         135      484
col_tot           2377         220     2597
```



```
AUC Score: 0.6193492273333515
```

In contrast to the operators class, I was not able to attain as good of results in this model. The biggest issue here was lack of data. After doing these subsets, I noticed that 42% of the operators' data set had churned while only 18.6% of the non-operators churned. These statistics are to be expected, however pairing the low percentage of churn with the considerably smaller dataset for the non-operators made it hard to produce a very meaningful model. Though the overall accuracy is not too bad, the precision and recall of the positive class suffered heavily. It is important to be aware in this sort of situation that predicting 0 employees would churn would produce a very meaningless model with a relatively high accuracy (almost 82%). With this in mind, I ran the grid search with the goal of maximizing recall, so I do believe that this is the best fit achievable given the circumstances. These results also lead me to believe that the lack of data among the non-operator job groups may be dragging down the accuracy of our general models. Here are some of the relatively more important features from this model:

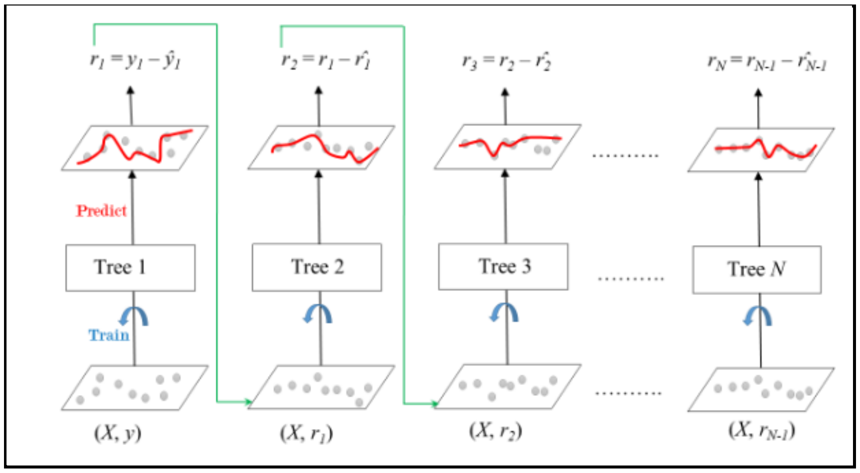
	Weight
Location_Suzhou China	-5.335403
Location_Massy France	-3.188215
Location_Penang Malaysia	-2.908454
Location_Gwanggyo South Korea	-2.128047
Education	-2.085337
Location_Havant United Kingdom	-2.008223
Compensation_Grade_Executive 4	-1.702603
Compensation_Grade_Engineering Professional 8	-1.684321
Compensation_Grade_Service Workers 3	-1.549632
Gender_D	-1.497977
Compensation_Grade_Management 8	-1.433537
Compensation_Grade_Support 2	-1.375010
Compensation_Grade_Management 7	-1.327314
Compensation_Grade_Engineering Support 3	-1.303979
Compensation_Grade_Sales Management 4	1.005115
Location_Korat Thailand	1.040862
Compensation_Grade_Sales Reps 1	1.048665
Compensation_Grade_Supervisors 2	1.092211
Compensation_Grade_Skilled Craft Workers 3	1.117924
Location_Taipei Taiwan	1.158929
Location_Shugart Singapore	1.226347
Location_Tokyo Japan	1.264327
Location_Cupertino United States	1.273808
Compensation_Grade_Supervisors 1	1.377453
Location_Woodlands Singapore W3	1.492376
Location_Woodlands Singapore W1	1.536868
Location_Woodlands Singapore W2	1.791610
Location_Penang Malaysia Suntech	1.869263
Location_Johor Malaysia	1.906991

Christopher Cannon - Gradient Boosting Classifier

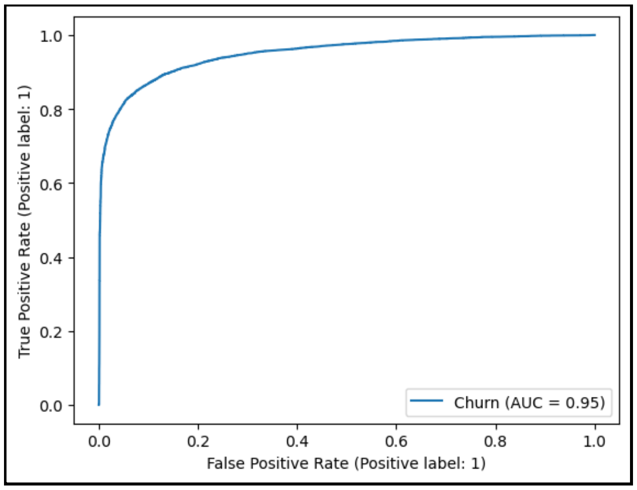
Introduction

The model I ran is a Boosted Decision Trees model using scikit learn's (sklearn) *GradientBoostingClassifier* module. Generally speaking, when gradient boosting, a model is built in which at each stage a tree is fit based on the gradient of the binomial loss function (in this case, where our problem is binary). Using subsequent trees one tree depends on the losses of its predecessor, hence the term "boosted", to differentiate between independent variables which lead to a positive dependent variable and those that do not.

In other words, and as described by the diagram provided by Geeks for Geeks, our model does the following. First, train a decision tree on the input data. Second, make predictions with the first tree. Third, return a value for the loss from the prediction against the data. Fourth, use that loss value to inform a new tree with the goal of minimizing the loss function. Repeat for the number of trees passed to the function.



I ran this model three times on three different versions of the dataset per our team's plan to learn whether certain variables are more informative as continuous or categorical. E.g. age is continuous but can also be organized into age groups or "buckets".

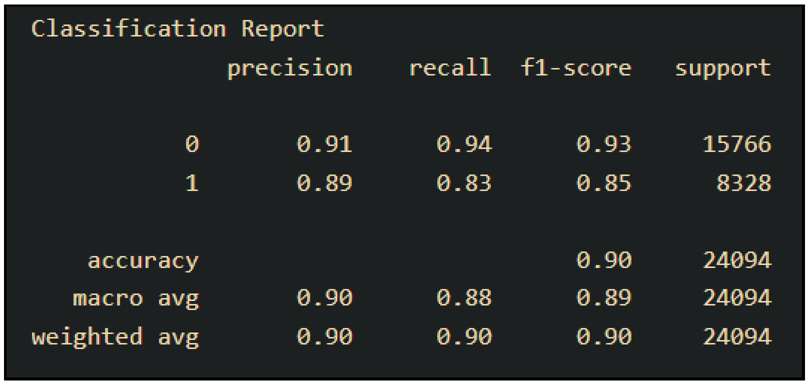


Model Performance: Training

This model performed exceptionally well in all three passes on the slightly different datasets. Here is a ROC Curve and Classification Report of the results from the first model. These results are similar to the results of the other two models.

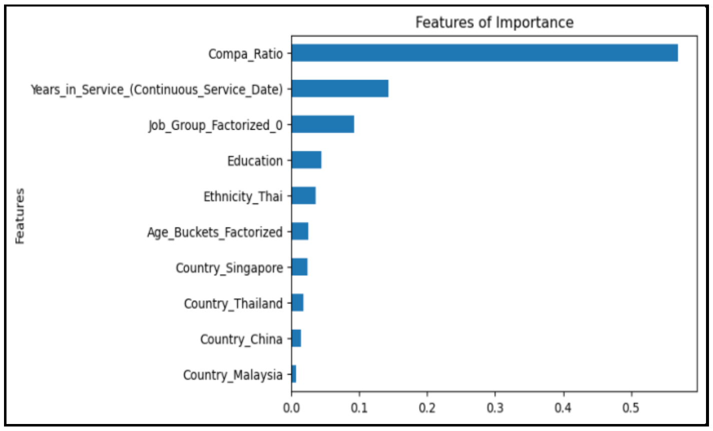
The ROC Curve presented provides an overview of the trade-off when generating predictions of those who will churn. At any point on the curve we find the probability of our churn prediction being correct (the y-axis) as well as the probability our churn prediction is not correct (the x-axis). In other words, if you want to correctly predict more of those who will churn we need to accept a lower confidence threshold which introduces more false churn predictions. We are looking for a line which gets as close to the top-left corner of the diagram as possible. Bottom line: this is an unusually good ROC Curve.

The Classification Report echoes the results of our ROC Curve. This report informs us that this model is, overall, 90% accurate and correctly predicting as churn 83% of those who will churn. Conversely, it also informs us that 89% of those we labeled as a churn actually do.



```
Classification Report
              precision    recall  f1-score   support


           0       0.91      0.94      0.93     15766
           1       0.89      0.83      0.85      8328

    accuracy                           0.90     24094
   macro avg       0.90      0.88      0.89     24094
weighted avg       0.90      0.90      0.90     24094
```

Overall, I was very pleased with these results. Too pleased, in fact. An out of the box accuracy of 90% indicated to me that the model may be learning to recognize churn risk from a flaw in the problem setup. Because I was skeptical, after running the models I also ran an analysis of the independent variables to learn which variables my model found most informative. I used the Mean Gini Index Reduction technique included in the *GradientBoostingClassifier* function to generate the Features of Importance plot (right).



With features on the y-axis and mean gini index reduction across the x-axis we can clearly see that the model is predominantly learning from the *Compa_Ratio* which employees have churned. And, accurately per the ROC Curve and Classification Report, predicting which employees have churned.

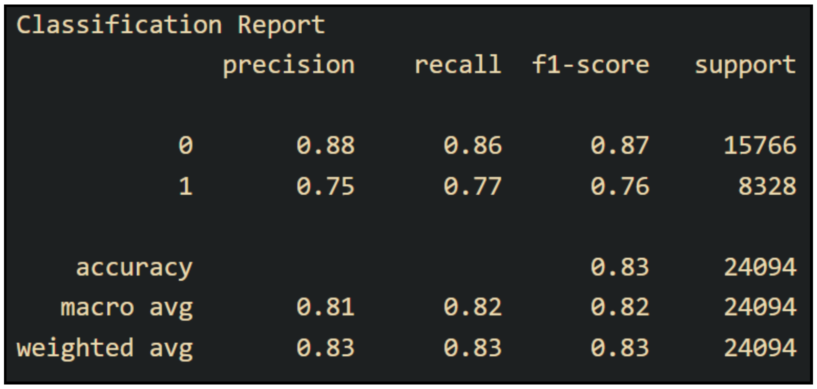
However, this is not accurate to the true business case because of *Compa_Ratio* degradation. According to [Company], once an employee leaves the company their compensation is locked in. But, *Compa_Ratio* will degrade over time as others in the industry continue to be compensated at higher and higher levels. Due to inflation, demand for employees, etc. Given *Compa_Ratio* degrades over time and we have churn data from the past 7 years, we would expect that current employees generally have a higher *Compa_Ratio* than employees who have left the company. And that is what our model has learned. But the business use case is to apply this model to current employees who *might* churn. In this case the *Compa_Ratio* is not degraded. Therefore, I decided the *Compa_Ratio* simply is not a variable we can use in its current state.

We have addressed this issue with [Company] in an attempt to apply some type of interest rate, to return *Compa_Ratio* to a level of normalcy. Not perfect, but better. Until then, due to this inappropriate leaning on *Compa_Ratio* I removed the variable and re-ran my models to obtain results I feel are truer to reality.

Model Quality (Results)

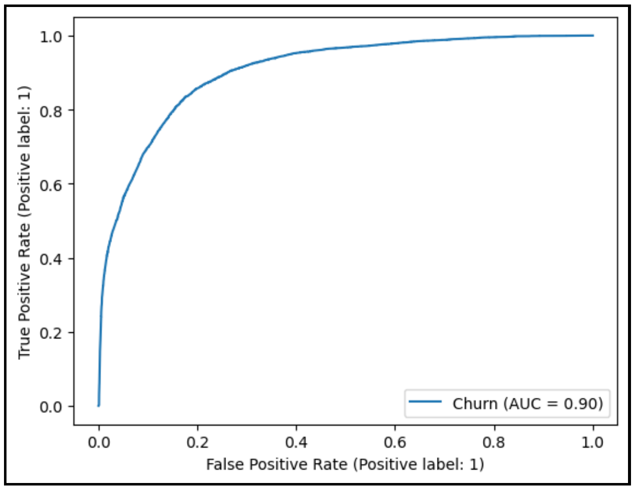
Luckily, even after removing *Compa_Ratio* my results were still better than originally expected. Provided is a table of Area Under Curve (AUC) values and Classification Accuracy for the three respective models, where Model 2 performed best (although barely).

As expected, when compared to the models which used *Compa_Ratio* there was a drop in performance across the board: AUC, Accuracy, Precision, Recall, etc. However, as mentioned, the results were still better than expected. Focusing on Model 2, I provide a ROC Curve, Classification Report, and Features of Importance for said model.



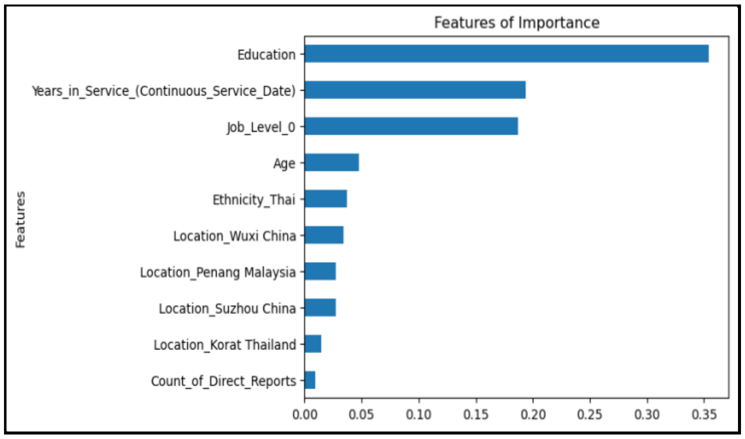
```
Classification Report
              precision    recall  f1-score   support

           0       0.88      0.86      0.87     15766
           1       0.75      0.77      0.76      8328

    accuracy                           0.83     24094
   macro avg       0.81      0.82      0.82     24094
weighted avg       0.83      0.83      0.83     24094
```

First, the classification report. This report provides two key takeaways. Namely, with this model we successfully predicted 77% of those who churned. Conversely, of those we said would churn, our model was correct 75% of the time. Meaning we can assign a high level of faith in our model both identifying those who would churn, and that when the model identifies someone as a churn risk they are likely (75% of the time) going to churn.



Second, the ROC Curve. This curve provides the information of the previous paragraph in an interactive, sliding scale format. In order to capture more and more of those who would churn (the y-axis) we have to be willing to accept more false positives (x-axis). However, the rate at which we must accept false positives changes depending on where you are in capturing churn risk. To capture only the churners we are most sure of we will likely not falsely identify someone as a churn risk. Conversely, if we want to capture as *many* churners as possible, we have to accept a high rate of false positives. Without having done a financial impact analysis I cannot provide a recommendation yet, but this will be a valuable tool when that time comes.

Third, and finally, features of importance. This plot provides the top-ten most important features (according to the model) in determining if someone is a churn risk. Again, this is based on a model which does not incorporate *Compa_Ratio*, as that single variable dominated the conversation previously. And while it may be insightful to know just how important *Compa_Ratio* is. In general, we want to learn more from the data and recommend more than simply "pay your employees more".

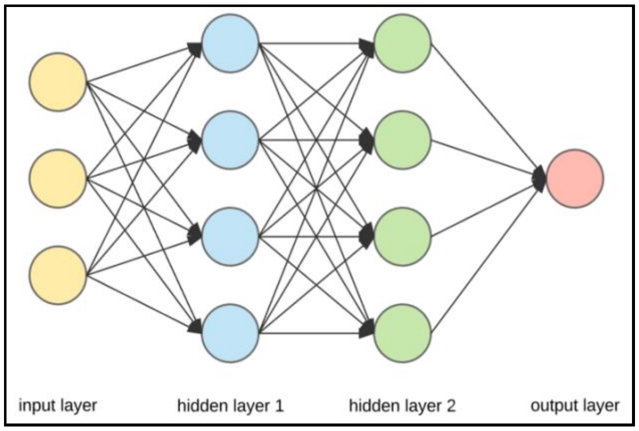


Unfortunately, we cannot interpret the results of this plot loosely. I cannot say that having education information leads to churn risk. I can say at this time that having education information, *Years_in_Service(Continuous_Service_Date)*, and being in *Job_Level_0* were all very important for our model to determine who was a churn risk. To be able to state with confidence whether having education information, for example, meant they were more or less likely to churn requires further analysis which we can conduct in the next phase of this project.

Poom Nichayapun : Neural Network

Introduction

Neural Network is considered one of the most booming algorithms prevalently used to solve today's data analytics problems. The algorithm reflects the behavior of the human brain in recognizing patterns of the dataset by having sophisticated layers of the input layer, hidden layers and the output layer to output the prediction result. Though capable of solving both regression problems and classification problems, the algorithm is primarily adopted to predict the churn rate of [Company] employee data.



Based on the business objective and available off-the-shelf neuron network algorithm on Scikit-Learn, I decided to choose the MLPClassifier (Multi-Layer Perceptron Classifier) to train and test the data because the target variable is of 2 classes. Hence, the Perceptron classifier should be sufficient for this binary classification. Though TensorFlow offers better configuration on deep learning algorithms, it is not necessary to fully utilize complex neural networks for a structured dataset as in this project. Practically, the training process would consume computational resources and would incur the additional costs from cloud services (e.g. GCP or AWS) for only low resource-consuming projects.

Model Performance: Training

To train the neural network model, I separated the training process into two distinct parts: Feature selection and Hyperparameters tuning. The train/test split ratios are 70/30 and 80/20.

Feature Selection:

Our baseline model starts with only numeric columns (*Year_in_service_(Continuous_Service_Date), Compa_Ratio, Count_of_Direct_Reports, Education, Age_Buckets_Factorized and Job_Group_Factorized*) as features to predict the target variable (*Status*) as a binary variable. The default hyperparameters are 100 hidden layers, Relu as activation function and 1000 epochs. The test accuracy begins with 79% without *compa_ratio* and skyrockets to 88% with *compa_ratio*. Initially, the results are acceptable. However, I extend the training schemes onto different combinations of features and subset of dataset to see how these impact the performance of the model.

General model:

In this model, I include some categorical columns on top of numerical columns (with compa_ratio) as mentioned in the previous subsection. After a myriad of feature combinations, the best option is to drop *Location*, *Ethnicity* and *Marital_Status* columns, while retaining all 80,311 rows of the cleaned dataset.

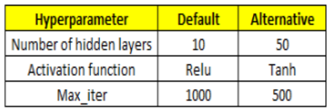
Job Category-specific model:

The general model is somehow too broad and generalized as it includes employees with different levels, regions and locations. Hence, I subset the dataset based on the unique values of *Job_Category* column. However, numbers of sample size for all types of *Job_Category* are too asymmetric. Hence, I only focus on the operator type as it represents the majority of observations in the dataset (54,430 rows or 67% of the entire dataset)and the sample size is large enough. The feature combination remains the same as the previous one.

Country-specific model:

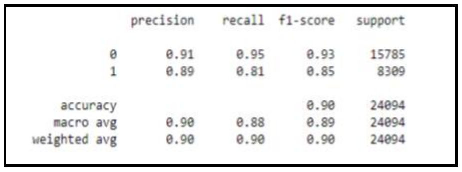
As cited in the previous section, the model would provide better insight if it is created based on the certain country or region since each country or region differs in cultural, economical and political landscape. Thus, the pattern of the dataset tied to the certain country may yield better performance. In this exploration, I implemented the model subsetted specifically for Thailand as it makes the largest sample size among other countries (31,332 rows or 39%) in the dataset. Furthermore, [Company] strategically outsources its majority of manufacturing process in Thailand, making Thailand an essential candidate for this country-specific model implementation.

<u>Hyperparameter tuning:</u>



Hyperparameter	Default	Alternative
Number of hidden layers	10	50
Activation function	Relu	Tanh
Max_iter	1000	500

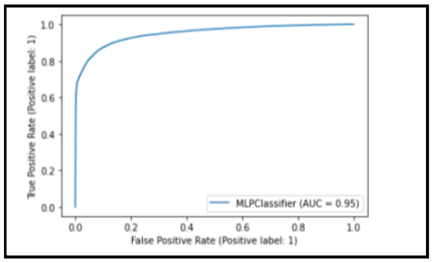
Despite features selected for the training, hyperparameters in this algorithm potentially impacts the performance of the analysis. To explore the feasibility of performance of MLP Classifier on training and testing the dataset , I selected 3 primary hyperparameters that are relevant to the training of our business objective: Number of hidden layers, activation function and maximum number of iterations. The detailed configuration of these 3 parameters are shown in this table. To train the models, we combine these configurations with the models with some selected features in the previous section and evaluate the performance.



```
              precision    recall  f1-score   support

           0       0.91      0.95      0.93     15785
           1       0.89      0.81      0.85      8309

    accuracy                           0.90     24094
   macro avg       0.90      0.88      0.89     24094
weighted avg       0.90      0.90      0.90     24094
```

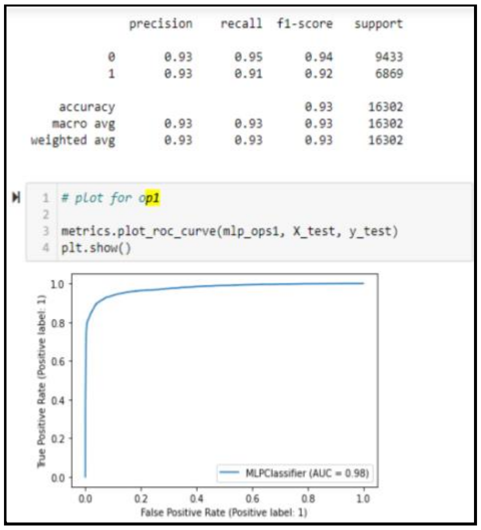
Model Quality (Results)

Generally speaking from the implementation of 3 primary types of models, hyperparameters slightly increase the classification metrics in exchange for a longer duration of training/testing time. For instance, when I increase the number of hidden layers from 10 to 50, the execution time is almost 5 fold compared to that of 10 hidden layers. Other hyperparameters such as activation function and max_iter barely improve the classification metrics. Hence, the results listed below would be based on the default settings as shown in the table of hyperparameter section.



General model:

As seen in the classification report and ROC curve below, the test accuracy is 90%, meaning that this model can predict whether the employee would churn and would not churn at 90% accuracy (10% could be wrong). The AUC in this ROC curve is 95%, meaning that there is a 95% chance that the model will be able to distinguish between positive class and negative class. In this case, positive class refers to those who actually churn and vice versa.



```
              precision    recall  f1-score   support

           0       0.93      0.95      0.94      9433
           1       0.93      0.91      0.92      6869

    accuracy                           0.93     16302
   macro avg       0.93      0.93      0.93     16302
weighted avg       0.93      0.93      0.93     16302
```

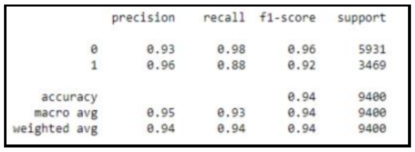
```
# plot for op1

metrics.plot_roc_curve(mlp_ops1, X_test, y_test)
plt.show()
```

True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
MLPClassifier (AUC = 0.98)

As a general model, this would reflect on the common pattern among job categories and countries. The next section would cover how the performance would be for the job-specific model.

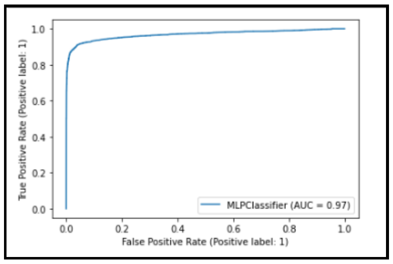
Job-specific model (Operators level):

With the narrow focus on operator level only, the test accuracy of the model increases from 90% to 93% as seen in the classification report. Other classification metrics (precision, recall and F1-score) also follow the same trend as the test accuracy. According to the ROC curve, the AUC surges to 98%, meaning that the chance of correctly distinguishing between the positive and negative class is 98%, which is satisfyingly accurate.



```
              precision    recall  f1-score   support

           0       0.93      0.98      0.96      5931
           1       0.96      0.88      0.92      3469

    accuracy                           0.94      9400
   macro avg       0.95      0.93      0.94      9400
weighted avg       0.94      0.94      0.94      9400
```

Nevertheless, note that this model would be accurate only to operator levels. Therefore, the result would be applicable only to operators if the main concern is the outflow of operators globally.



Country-specific model (Thailand):

When shifting the focus to the specific country such as Thailand, the country that makes up the majority of the dataset, the test accuracy of the model increases from 90% (the test accuracy of the general model) to 94% as seen in the classification report. Other classification metrics (precision, recall and F1-score) also follow the same trend as the test accuracy.

According to the ROC curve when compared to that of the general model, the AUC surges to 97% meaning that the chance of correctly distinguishing between the positive and negative class is 97%, which is decently accurate.

Nevertheless, note that this model would be accurate only to all employees in all job categories in Thailand. Therefore, the result would be applicable only to those working in Thailand if the main concern is the outflow of employees in Thailand.

Conclusions

Recommended Model

Model Comparison

Model Type	Accuracy	Precision	Recall	F1	Processing Time
Gradient Boosting	0.90	0.89	0.83	0.85	~45sec
Neural Network	0.90	0.89	0.81	0.85	~3-4m
Random Forest	0.90	0.90	0.80	0.85	~25sec
Support Vector Machine	0.90	0.90	0.80	0.85	~30m
Logistic Regression	0.84	0.76	0.76	0.76	~1-2m

To evaluate the performance of the binary classifier, we rely on classification metrics: test accuracy, precision, recall and F-1 score. Another metric that we take into consideration is processing time as practically we do not wish to sacrifice time and computational resources to achieve only slightly better prediction scores.

According to the result listed in the table above, the first 4 models perform relatively well in terms of classification metrics. However, when we consider the processing time, Random Forest is the fastest model and can be executed locally without the use of additional cloud computing services.

However, the accuracy of the model can be substantially improved if we subset the dataset for certain job categories or countries as seen in the result in Neural network and other models described previously.

Model Insights

We implemented five different machine learning algorithms to deduce the most promising insights from [Company]'s employee data. After accomplishing efficient data cleaning and preprocessing, we were able to infer churn predictions from the data almost 90% of the time across all models.

With Logistic Regression, we could adequately increase prediction accuracy and recall targeting reduced False Negatives. Next, even though the SVM model took roughly 30 minutes to gather insights from the training dataset, it could predict churn rate with a high precision of 90% on the test data indicating high generalizability on new data. The quickest of the lot was the Random Forest algorithm giving high overall performance. With fine-tuning of hyperparameters, we correctly predicted voluntary termination 90.2% of the time.

Neural Networks again showcased a high precision and a recall score of 89% and 81%, respectively, indicating that it predicted 89% of the time when an employee would churn, and out of that number, 81% of the time that employee did leave the company. And lastly, Gradient Boosting Decision Trees produced the best Recall score of 85% amongst all models.

All our models predicted highly accurate scores, which will help us gather valuable information to decide on important factors influencing the turnover and eventually reducing voluntary churn.

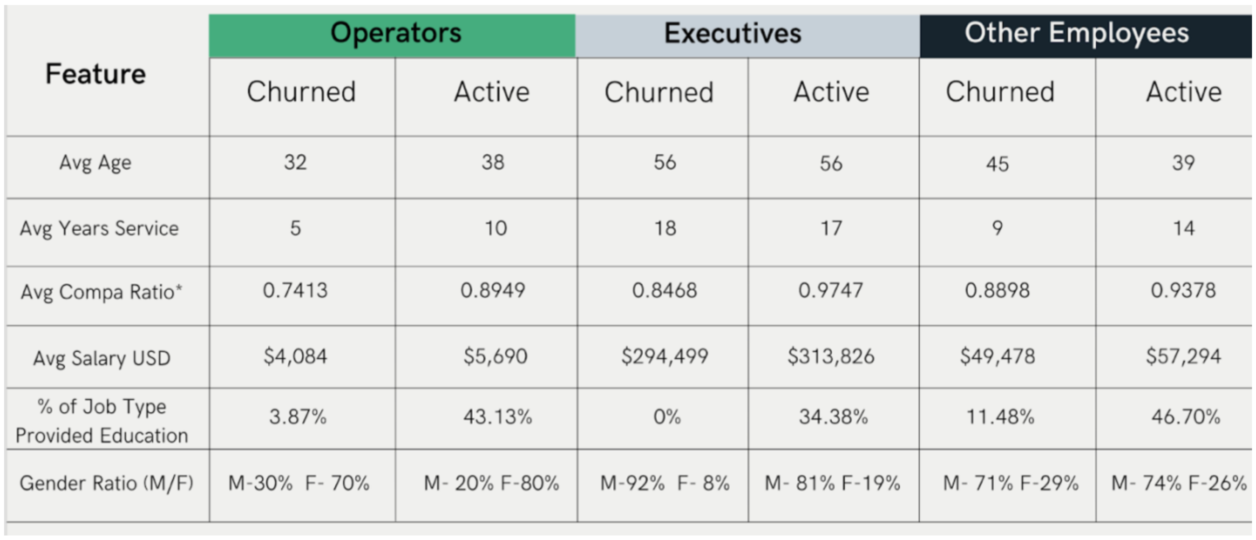
Down Select and Recommendation

While all of our model variations produced similar results, Gradient Boosting, Neural Network, and Random Forest seemed to be the three best options for us to consider for our final model. We chose to eliminate SVM based on the processing time and to eliminate Logistic Regression because it had a lower accuracy than the rest of the models. Overall, we are extremely satisfied with an accuracy of 90%, and we feel that any of the three final contenders would produce amazing results to help [Company] predict which employees are likely to churn. However, for our final model, we will be utilizing the Gradient Boosted Decision Tree as it not only gave the best results in accuracy, precision, recall, and F1, but it also has a quick processing time which makes it scalable for more data.

The next steps will be to develop actionable insights on how [Company] can improve overall churn rate. To do this, we will apply our model to active employees, predict which of them are likely to churn, develop recommendations to keep employees from leaving, and conduct an economic analysis to assign monetary value in implementing these suggested changes.

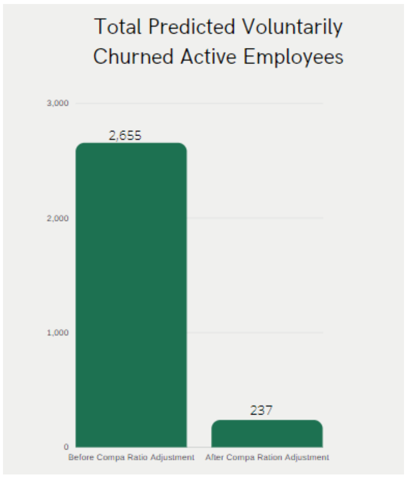
Business Impact

Using our model to predict reductions in employee churn, we estimate that of the 2,655 active employees identified as a churn risk. With the data that we had access to, we only felt comfortable making recommendations based on numerical values, rather than anything descriptive. As compa ratio was the most impactful and important feature in our final model, we felt it was best to evaluate how [Company] can adjust these numbers while still saving money they would lose from employees churning. In an ideal world, we would be able to create a model that predicts the most cost effective increase in compa ratio; however, due to lack of time, we decided to lean on the data analysis instead. In the chart below, you can see the differences between employees predicted to churn vs. employees predicted to be active.



Feature	Operators		Executives		Other Employees	
	Churned	Active	Churned	Active	Churned	Active
Avg Age	32	38	56	56	45	39
Avg Years Service	5	10	18	17	9	14
Avg Compa Ratio*	0.7413	0.8949	0.8468	0.9747	0.8898	0.9378
Avg Salary USD	$4,084	$5,690	$294,499	$313,826	$49,478	$57,294
% of Job Type Provided Education	3.87%	43.13%	0%	34.38%	11.48%	46.70%
Gender Ratio (M/F)	M-30% F- 70%	M- 20% F-80%	M-92% F- 8%	M- 81% F-19%	M- 71% F-29%	M- 74% F-26%

Based on the above statistics, we chose to use the difference between compa ratio of Churned and Active employees for a point of reference for the recommended increase in Compa Ratio. Additionally, we used the difference in years of service to recommend to [Company] at what point during the employee's life cycle to implement these changes. The recommended changes that we incorporated in our economic impact calculation are outlined as follows:



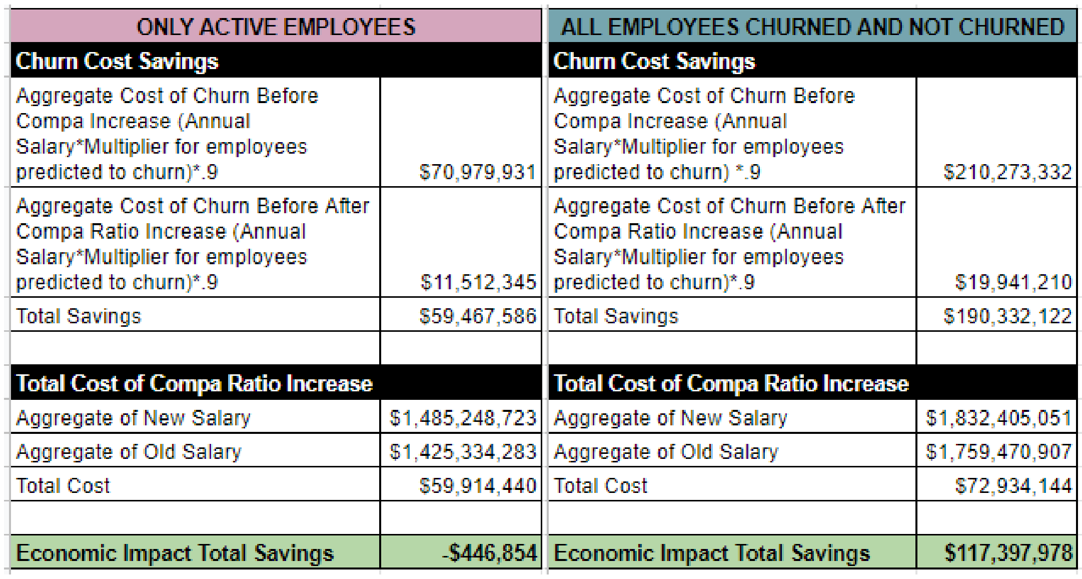
1) Increase Compa Ratio by .15 for Operators at around 5 years of service.

2) No changes to Executives' Compa Ratio.

3) Increase Compa Ratio by .05 for "Other Employees" at around 9 years of service.

If [Company] increases Compa Ratio according to our recommendations, all but 237 are no longer a churn risk.

The business impact of implementing our model depends on how [Company] implements the recommendations. We calculated economic impact using two different scenarios. First, we conducted this calculation across all employees in the dataset, regardless if they are former employees or currently active . This gave us the ability to estimate how much money [Company] could have saved in the long run. Second, we applied our economic impact calculation to the data subset of currently active [Company] employees. Summaries of our calculations and their respective impacts are provided in the tables below.



ONLY ACTIVE EMPLOYEES		ALL EMPLOYEES CHURNED AND NOT CHURNED	
Churn Cost Savings		**Churn Cost Savings**	
Aggregate Cost of Churn Before Compa Increase (Annual Salary*Multiplier for employees predicted to churn)*.9	$70,979,931	Aggregate Cost of Churn Before Compa Increase (Annual Salary*Multiplier for employees predicted to churn) *.9	$210,273,332
Aggregate Cost of Churn Before After Compa Ratio Increase (Annual Salary*Multiplier for employees predicted to churn)*.9	$11,512,345	Aggregate Cost of Churn Before After Compa Ratio Increase (Annual Salary*Multiplier for employees predicted to churn)*.9	$19,941,210
Total Savings	$59,467,586	Total Savings	$190,332,122
Total Cost of Compa Ratio Increase		**Total Cost of Compa Ratio Increase**	
Aggregate of New Salary	$1,485,248,723	Aggregate of New Salary	$1,832,405,051
Aggregate of Old Salary	$1,425,334,283	Aggregate of Old Salary	$1,759,470,907
Total Cost	$59,914,440	Total Cost	$72,934,144
Economic Impact Total Savings	**-$446,854**	Economic Impact Total Savings	**$117,397,978**

Model Limitations and Next Steps

In submitting these recommendations to [Company], we wanted to make sure to do our due diligence and touch on the potential limitations of our model. Mainly, our model placed a massive importance on both compa ratio and the education column we created (whether or not an employee chose to disclose their education background). If it is the case that a terminated employee's compa ratio is not frozen at the time they leave the company, we would expect to see the trend that terminated employees have a lower compa ratio (since it will continue to decrease), and our specific recommendations might not hold as much water. Similarly, if it is perhaps the case that [Company] has lost some of the education data on past employees, we would also expect to see the observed trend that terminated employees were less likely to have education data and we can no longer conclude this is due to a voluntary decision on the employee's part. Further, if [Company] were able to obtain more education data, we would have loved to see what impact a more specific education feature would have had.

Finally, we wanted to be able to make more specific recommendations about each region and each job type. In terms of region, we simply ran out of time to generate these models but we believe this would have also helped with some of the multicollinearity in our model - an example of this being the model placing a large importance on the "Thai" ethnicity when this relationship was more likely due to the fact that most of the employees in our data were Thai and most of those were operators. We did attempt to split our data by operators and other job types but ran into the problem of not having enough data on the non-operator job classes. We realize that losing an employee in a non-operator position will typically incur a higher cost to [Company] so it was disappointing to not be able to make better recommendations there. Further, we believe that the lack of data among non operators contributed to lower accuracy in our final model as a whole.

Finally, the biggest limitation our model had was its inability to predict *when* an employee would churn. We have begun creating a model to predict length of service which could be used in conjunction with our current model (as in, employees we predict will churn can be run through the length of service model to predict when this will happen). But due to a lack of time resources we were unable to fully tune and finalize this model.

In another universe, where we had more data and more time (or perhaps in [Company]'s future), there are a few next steps we would have loved to be able to take. First, some additional data we believe would have provided meaningful insights includes length of time since last promotion and/or raise, number of promotions and/or raises each employee has received during their time at [Company], engagement scores, and responses to surveys (not necessarily just numerical ratings but also raw word responses). With this data, a sentiment analysis model could be constructed which would also help inform who may churn and may also be able to provide less obvious insights into the reasons people are leaving.

Appendix A: Complete Categorical Data Description

Below is the complete table previously mentioned in the Categorical Data section of this report.

Feature	Count of Unique Values	Team Description	Count of Nulls	% of Values are Null
Employee Status	3	Job Type	0	0.0%
Employee Type	3	Job Type	0	0.0%
Job Category (Job Class)	13	Job Type	4883	5.7%
Job Group	6	Job Type	4883	5.7%
Job Family (Job Class)	176	Job Type	4883	5.7%
Is Full Time Employee?	1	Job Type	360	0.4%
Frequency	8	Compensation	317	0.4%
Generation	4	Employee Age	0	0.0%
Gender	3	Employee Demographic	3	0.0%
Marital Status (Label)	10	Employee Demographic	3047	3.6%
Region	3	Employee Location	0	0.0%
Time Type	2	Job Type	0	0.0%
Country	35	Employee Location	0	0.0%
Location Name	139	Employee Location	0	0.0%
Length of Service - Buckets	6	Length of Employment	0	0.0%
Age - Buckets	7	Employee Age	0	0.0%
Degree (Picklist Label)	24	Employee Education	56817	66.5%
Did you graduate? (Picklist Label)	3	Employee Education	56817	66.5%
Major (Picklist Label)	462	Employee Education	56817	66.5%
Education Record is blank?	2	Employee Education	0	0.0%
Highest Degree	24	Employee Education	56817	66.5%
Is Highest Degree?	2	Employee Education	0	0.0%
Race/Ethnicity	20	Employee Demographic	28316	33.1%
Management Level (Picklist Label)	7	Job Type	5	0.0%

Compensation Grade	66	Compensation	4883	5.7%
Comp Grade Profile	699	Compensation	4883	5.7%
Event Reason Name	5	Termination Reason	42758	50.0%
Rating Label	3	Employee Performance	72834	85.2%
Termination Reason (externalName)	35	Termination Reason	43414	50.8%
Job Category (Picklist Label)	13	Job Type	5231	6.1%
Location (Location Name)	139	Employee Location	0	0.0%
Organization (Label)	782	Employee Location	3	0.0%
Pay Grade	958	Compensation	0	0.0%
Currency	26	Compensation	317	0.4%
GID_anonymized	82150	Index	0	0.0%