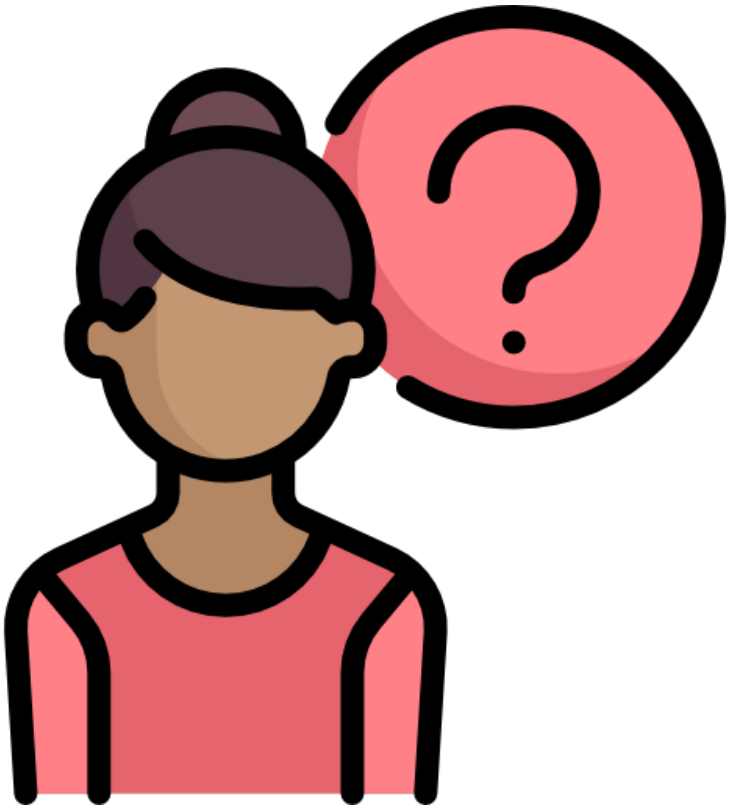
Predicting Employee Retention by Department



Icons made by Freepik from https://www.flaticon.com/

Group 8

Leah Jenkins

Joshua Sheriff

Marjan Jabalameli

Vaibhav Chaudhary

Table of Contents

Business Understanding……...**2**

Data Understanding……....**4**

Data Preparation……**10**

Modeling Phase……...**12**

Evaluation Phase……....**16**

Deployment Phase……...**17**

Works Cited……...**19**

Business Understanding

Background

For many HR departments, trying to determine which factors may lead to an employee leaving the business is of critical importance. High employee turnover rates not only impact the business' overall bottom line due to disruptions in project continuity, but it can also lead to further negativity amongst remaining staff, a lowered public perception of the company and make it much harder to attract and retain skilled talent in the future. Exploring and learning about what may be possible drivers that impact employee churn rates would permit HR to more effectively design and implement retention solutions, allow for early intervention for at-risk staff saving hiring and training costs and assist in promotion selection, etc.

Business Objective

The objective of our research project is to investigate if can we predict which drivers will lead an employee to decide to leave the organization across the different departments. By looking at a large sample of employee records (> 10,000 records), we will set out to devise an algorithm that can be used to predict whether a current employee is likely to leave the organization.

Success Criteria

For this project, success criterion includes:

- Delivering the project proposal and scope to the client by October 25th for approval
- Providing Python source coding for our proposed predictor algorithm
- Provide insight into which variables have the highest impact on an employee leaving
- Providing a project wrap-up report and presentation of our findings on November 3rd

Resources Inventory

- Project consultation team: Vaibhav, Marjan, Josh, Leah
- Business experts: Leah & Marjan
- Data scientists: Josh & Vaibhav
- Data: CSV file from HR department with detailed records of more than 10,000 employees (https://www.kaggle.com/pankeshpatel/hrcommasep)
- Software resources: Jupyter Notebook (using Python language)
- Python packages: Pandas, Numpy, Matplotlib, Sci-kit Learn, Seaborn

Requirements, Assumptions & Constraints

Our team consulted with the company that hired us to ensure that any data contained within the CSV file was anonymized (no PII data) to ensure from a legal perspective that none of the dataset that we are working with can be traced back to any current or past individual hired by the organization. The raw data table was permissioned for exclusive use by our project consultant team, and additional security measures were taken to ensure that data is stored in a secure, password-protected login that is not being shared with anyone else within our organization.

Schedule of completion:

- Project proposal outline due October 25th

- Data understanding and preparation phases October 20th - October 31st
- Algorithm testing and review phase October 29 - October 31st
- Project wrap-up, including Python notebook due November 2nd
- Presentation of findings November 3rd

For the purpose of our research, we assume that the data contained within the provided CSV file (see attachment) is up-to-date and does not contain any clerical input errors (such as mistyped records) which could incorrectly bias or influence the dataset used in our learning algorithm.

We do not foresee any major constraints regarding our ability to complete our research, such as timelines, availability of staff resources nor technological impacts due to the size of dataset we are working with for the scope of this project.

Risks & Contingencies

Potential risks/events that may impact the completion of this project include:

- Not receiving sign-off from the client on the proposal in a timely manner. Contingency to this is to simply continue with the research as scoped by our team and present the findings to the client, regardless of not receiving sign-off by deadline.
- Inability to correctly test proposed algorithm due to technological issues (such as issues with program coding).

Costs & Benefits

The cost associated with our team's time allocated towards this project is estimated to be $8,000 CAD. Based on external research, the average cost to hire a new employee is anywhere between $1,000 - $4,000[1], which based on our conversations with the HR team, in the past year they've had to hire more than 20 new employees to fill positions that were vacated by past employees. Having spent anywhere between $20,000 - $80,000 on new hire costs in 2019 alone, we believe that the $8,000 required to investigate how to predict churn will quickly repay itself in savings if it leads to as few as 8 entry-level employees (or 2 more senior employees) leave the company's employ.

Data Model Goals

The goal of our data model is to determine how few variables will most accurately predict whether an employee will leave the organization. We plan on determining this by devising 2 different versions of our algorithm to compare and contrast the prediction score output.

Model Success Criteria

Success for our model is to get to a prediction accuracy of 80% or higher based on our algorithm. We would also look to have a precision of at least 65% for our model.

[1] Workopolis,'How to calculate cost to hire (and why it's important)', March 27, 2018, https://hiring.workopolis.com/article/how-to-calculate-cost-to-hire-and-why-its-important/, Accessed October 20, 2019

Project Plan

Phase	Timeframe	Resources	Risks
Business Understanding	October 10 - October 14	Business experts Data scientists	Lack of support from senior leadership at company, delay in receiving data
Data Understanding	October 20th - October 31st	Data scientists Business experts	Technology issues, not enough data records to use for modeling
Modeling	October 29 - October 31st	Data scientists	Data problems, issues with modeling software
Evaluation	October 31 - November 2	Data scientists Business experts	Issues with software, issues with data model
Deployment	November 3	Data scientists Tech department	Inability to implement model,

For this project, we will be utilizing Python programming language and Jupyter notebook to help clean, manipulate and develop our model based on the dataset provided. Jupyter was selected as the software tool as it is easy to show step-by-step the model development and results back with the tech department for future deployment.

Data Understanding

Initial Collection Report

The anonymized employee data records were provided to us via a CSV file from the HR department. We did not encounter any issues with the transmission of the master data file. Below is a screenshot of the provided data file:

	A	B	C	D	E	F	G	H	I	J
1	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	Department	salary
2	0.38	0.53	2	157	3	0	1	0	sales	low
3	0.8	0.86	5	262	6	0	1	0	sales	medium
4	0.11	0.88	7	272	4	0	1	0	sales	medium
5	0.72	0.87	5	223	5	0	1	0	sales	low
6	0.37	0.52	2	159	3	0	1	0	sales	low
7	0.41	0.5	2	153	3	0	1	0	sales	low
8	0.1	0.77	6	247	4	0	1	0	sales	low
9	0.92	0.85	5	259	5	0	1	0	sales	low
10	0.89	1	5	224	5	0	1	0	sales	low
11	0.42	0.53	2	142	3	0	1	0	sales	low
12	0.45	0.54	2	135	3	0	1	0	sales	low
13	0.11	0.81	6	305	4	0	1	0	sales	low
14	0.84	0.92	4	234	5	0	1	0	sales	low
15	0.41	0.55	2	148	3	0	1	0	sales	low
16	0.36	0.56	2	137	3	0	1	0	sales	low
17	0.38	0.54	2	143	3	0	1	0	sales	low
18	0.45	0.47	2	160	3	0	1	0	sales	low
19	0.78	0.99	4	255	6	0	1	0	sales	low
20	0.45	0.51	2	160	3	1	1	1	sales	low
21	0.76	0.89	5	262	5	0	1	0	sales	low
22	0.11	0.83	6	282	4	0	1	0	sales	low
23	0.38	0.55	2	147	3	0	1	0	sales	low
24	0.09	0.95	6	304	4	0	1	0	sales	low
25	0.46	0.57	2	139	3	0	1	0	sales	low

As previously outlined, the data itself is stored in a secure, password-protected location on our internal server.

Data Description

As noted, the data is in CSV format. 10 features have been included in the file:

- **satisfaction_level**: This represents the employee's self-rated satisfaction level at their job. Rated on a rating scale from 0 (lowest) to 1 (highest).
- **last_evaluation**: This represents the rating received by an employee on their last job evaluation. Rated on a rating scale from 0 (lowest) to 1 (highest).
- **number_project**: The number of work projects an employee was involved in.
- **average_monthly_hours**: This represents the average number of hours in a month spent by an employee while at work.
- **time_spend_company**: This represents the number of years an employee has spent in the organization.
- **Work_accident**: This reflects whether the employee suffered a work-related accident during their employment tenure. 0 indicates no accident, while 1 indicates an accident occurred. No further details provided on what type of accident may have occurred.
- **left**: This reflects whether an employee remains in the organization or whether they have chosen to leave the company. 0 indicates the employee remains at the company, while 1 indicates the employee has left the company.
- **promotion_last_5years**: This represents the number of promotions an employee has received within the last 5 years of their tenure.
- **Department**: This represents which department the employee belong to within the organization.
- **salary**: This represents the salary paid to the employee. Individual salaries have already been clustered into 3 groups: low, medium or high.

The full CSV file contains 14,999 individual employee data records. After initial review, we did not find any missing or incomplete fields in the file. The data has passed our requirements in order to use for the purposes of our research.

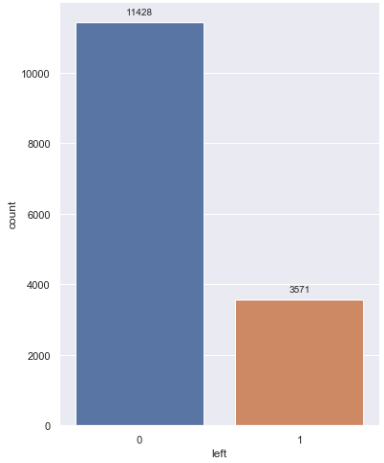
Data Exploration

Initially, we wanted to explore and determine overall how many employees in the larger dataset had actually elected to leave the company.

As shown in Figure 1, we can see that of the total 14,999 employees, nearly 1 in 4 employees have left the organization. Per a 2016 Compensation Force Study[2], the industry average employee turnover rate was 17.8%. In this organization, we can see that the churn rate is skewing higher than the average, which further highlights why the need to help predict whether employees may exit in future even more pressing.

[2] Nicole Roder, Zenefits,'Does Your Company Have a Healthy Employee Retention Rate?', April 4, 2019, https://www.zenefits.com/workest/your-company-healthy-employee-retention-rate/, Accessed October 21, 2019

Figure 1 - Overall depiction of employees who stayed vs left



From here, we next explored what the churn rate looked like by different departments, so see if we could identify particular departments where the turnover rate is higher or lower than the industry average.

Figure 2a - Employees staying vs leaving by department

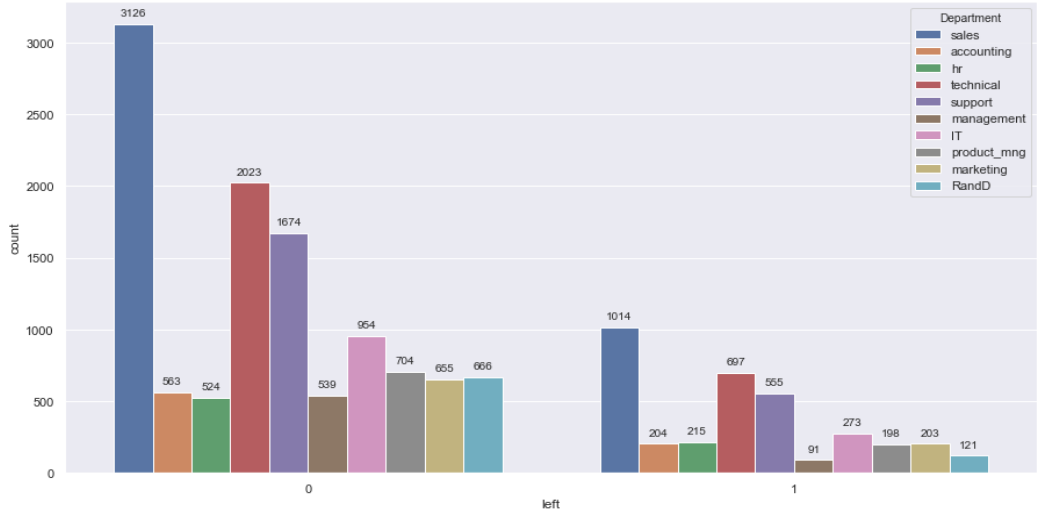
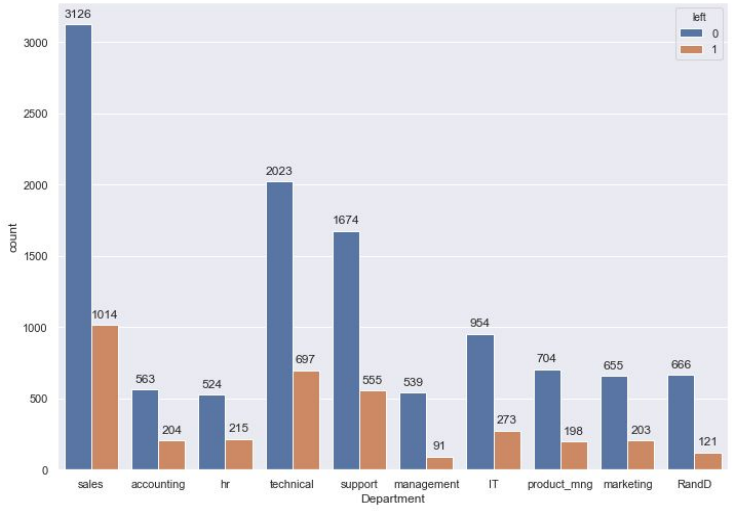


Figure 2b - Employees staying vs leaving by department



As shown in Figures 2a &2b, we can see that certain departments exhibit higher churns - in particular, the Technical department saw an estimated churn of 34%, while HR and Accounting rounded out at 29% and 27% respectively. Departments that appeared to be trending below the overall baseline rate of 24% were R&D (15%), IT and Product Management (22% each). By exploring the turnover by department, HR can gain further intelligence into departments that are at greater risk for employee loss. We would recommend that the organization take note of these rates and begin to conduct further internal review of these departments to gleam further information into what may be driving the departures.

Next, we wanted to explore all of the features provided in the datafile to see if we could start to identify any particular features that may indicate some correlation with each other. We used the Pairplot function to graph out all features, as shown in Figure 3.

In particular, we noticed an interesting clustering occurring between the average hours worked in the month and job satisfaction, as well as average hours worked in the month and last job evaluation. In Figure 4, there appears to be a clustering of departed employees who rated their job satisfaction as low and worked amongst the lowest monthly average hours. What's also interesting is that amongst those who worked the most average hours and also rated their job satisfaction higher along the scale (around 80%+), they still chose to leave the organization. This would seem to suggest that there's perhaps some correlation between the number of hours worked in the month and job satisfaction that may play a role in an employee's decision to leave or remain with the organization. In a similar clustering view in Figure 4, there's a similar trend amongst exited employees who worked fewer average hours but scored a less favourable job evaluation. Interestingly, amongst employees who worked some of the highest average hours and scored very favourable job evaluation we also see a heavy cluster of them exiting the company.

We also wanted to understand if there was any influence on tenure with the company and someone opting to leave. As shown in Figure 5, there was a very high exit rate amongst employees who had spent 5 years in the organization - in fact, more employees at this stage left than even considered remaining with the company. Perhaps this is linked to not having received any promotions within their 5 years, and may have influenced their choice to leave. The longer the overall tenure with the organization, the less likely an employee is to want to leave.

It may be worth further exploration from the HR department to look into the ages of their exiting employees, as it may be the case that a number of those exiting around the 3-5 year mark may be younger or in more entry-level job positions that are looking to make their next big jump up in their careers.

Figure 3 - Pairplot of features in datafile

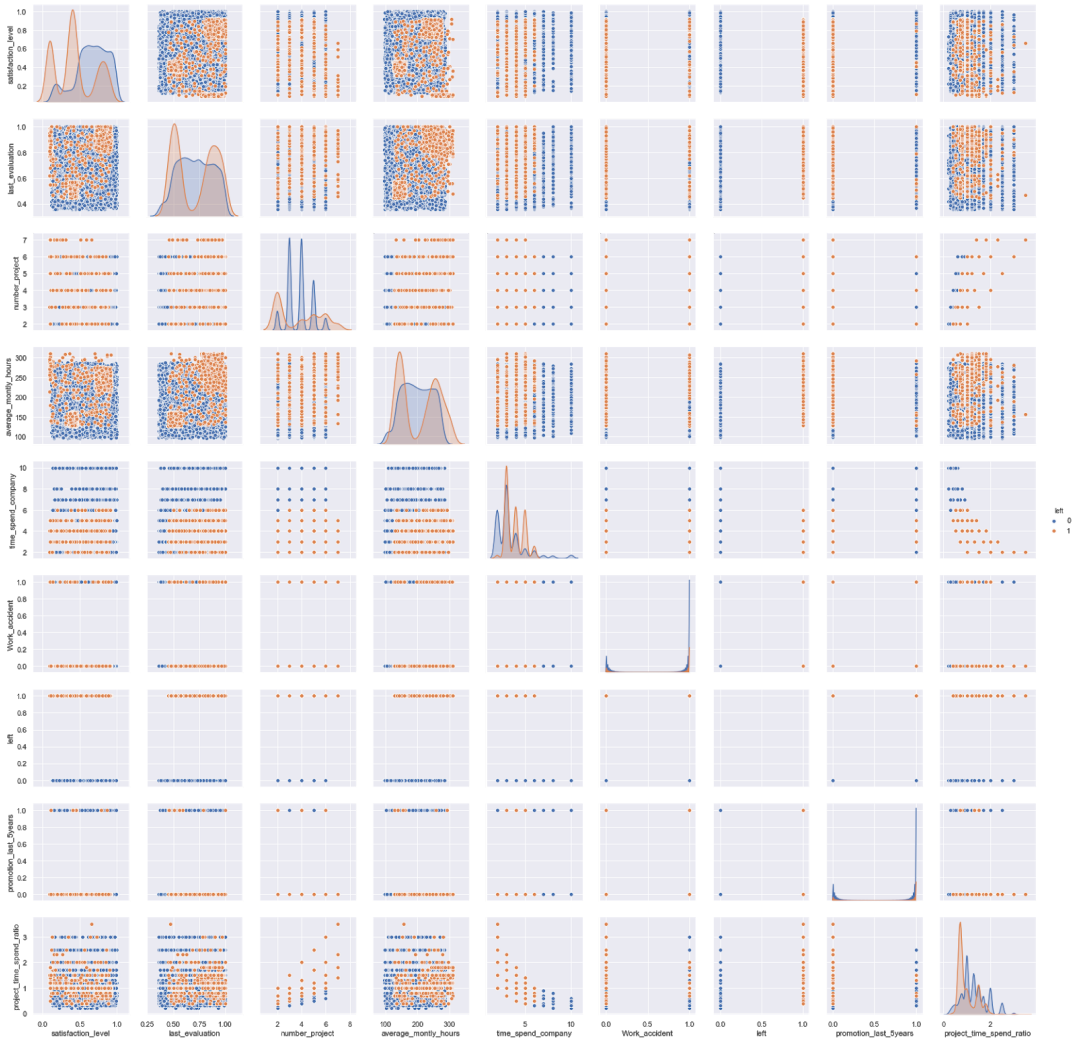


Figure 4 - Average Hours Worked in month vs Job Satisfaction and vs Last Evaluation

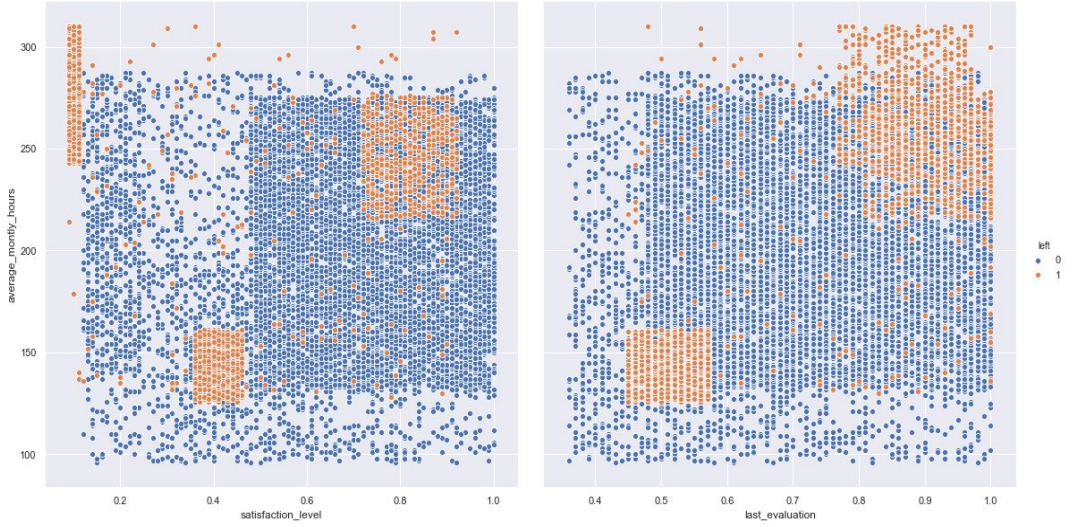
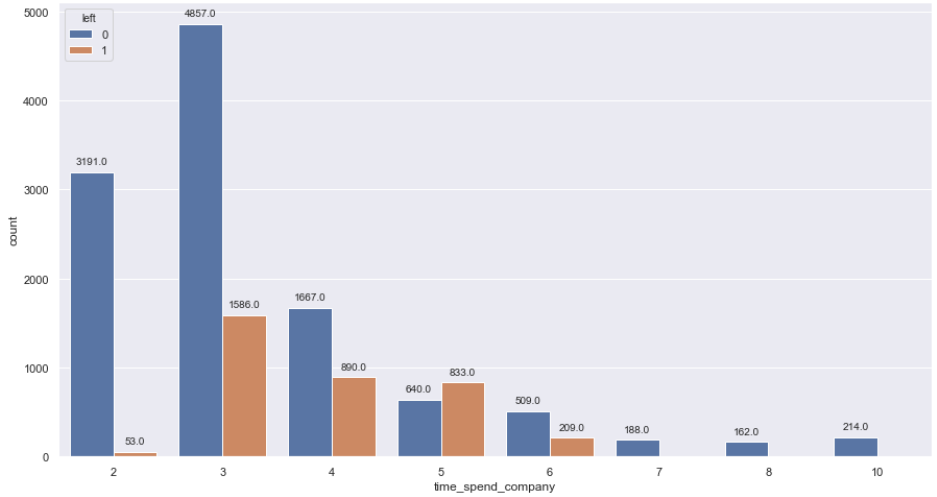
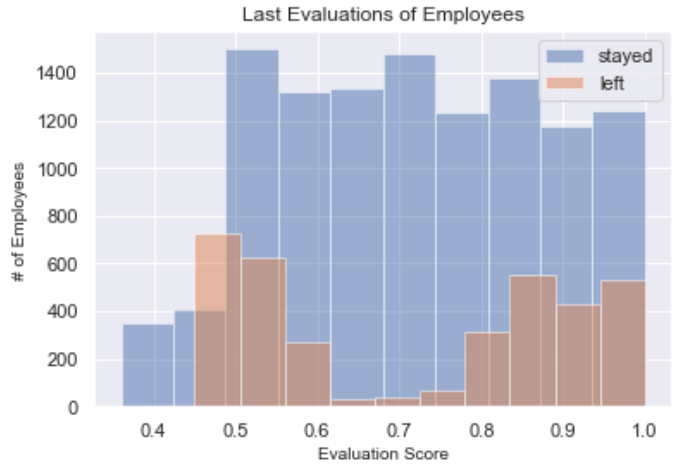


Figure 5 - Turnover by amount of time spent working at the company



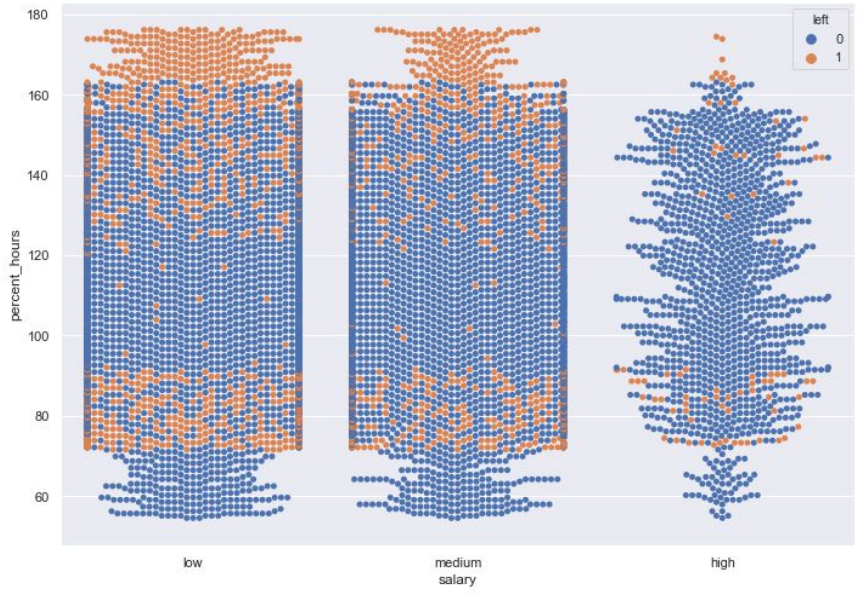
We also looked to compare the employees who stayed/left and their evaluation scores to understand the distribution, as illustrated in Figure 6. Amongst remaining employees, we saw that they consistently were scoring above 0.5 on their evaluations, but departing employees appeared to be clustered around scoring an average evaluation (approx. 0.5) or were on the higher end (0.85 and up). This would seem to suggest that among those with higher evaluations, perhaps they felt they had outperformed and wanted to seek out opportunities elsewhere.

Figure 6 - Number of employees who left and stayed grouped by last evaluation score



Lastly, we explored creating a percent hours worked metric (Average_monthly_hours/176*100) and used this to help us visualize what this looks like across salary ranges. As shown in Figure 7, it is clear that there is a higher likelihood of departure amongst those employees who are working above normal hours. Interestingly though, there also appears to be more likelihood to leave amongst those working some of the least amount of work hours. This is especially evident amongst the low and middle salary employees, whereas those earning the highest salaries see the least attrition regardless of the number of hours worked.

Figure 7 - Percent hours worked by salary range



Data Preparation

Data Selection Rationale

Based on our initial exploration of the data, we want to explore creating 2 different models - one that utilizes all the available features, and a second one with a smaller subset of features that we felt seem to indicate a stronger influence on the likelihood of an employee exiting the organization. The goal behind devising both models is to see which results in a higher and more accurate prediction score.

For the second model with only a few select features, we identified the below variables to include:

- **Department:** As we are seeking to devise a model that can predict employee departure by department, and after seeing that different departments had differing rates of turnover, this is a variable we want to ensure is included.
- **Satisfaction level:** As we uncovered, satisfaction appears to play a significant role in an employee's decision to remain or leave.
- **Average monthly hours**: As highlighted in Figure 4, the higher monthly hours worked, the more it seems to influence an individual's decision making.
- **Last evaluation**: As shown in Figure 4 and 6, how someone performed on their job evaluation appears to be an indicator of whether they leave or remain.

The output variable to be used in both versions of our algorithm will be whether the employee left (1) or remained (0).

Some features that we felt were not working including in the second model design include:

- **promotion in the last 5 years** - we noted that very few employees across the organization had received a promotion
- **number of work accidents** - very few accidents were noted, and without further clarity on the type of workplace accident we did not feel this was worth including in the model

- **salary** - based on the data, many employees are within the low or medium salary range, with very few in the high range. Due to this uneven distribution, we have opted to leave it out of this model
- **number of projects** - we noticed that anyone working on 7 projects (the highest amount per year) all left, but we didn't see enough of people leaving when working fewer than that to feel that this was a metric worth including in the smaller model

Data Cleansing

We checked for any issues/discrepancies and ensured uniformity in the data types of the various variables that might create problems for data processing. We also checked for any missing or incorrect values within the data and found that the data was clean and ready for processing further.

Describe data

```
In [12]:  1 df.describe()
```

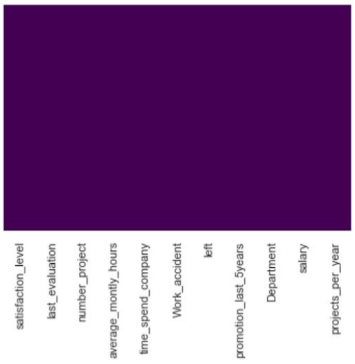
Out[12]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5year
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.00000
mean	0.612834	0.716102	3.803054	201.050337	3.498233	0.144610	0.238083	0.02126
std	0.248631	0.171169	1.232592	49.943099	1.460136	0.351719	0.425924	0.14428
min	0.090000	0.360000	2.000000	96.000000	2.000000	0.000000	0.000000	0.00000
25%	0.440000	0.560000	3.000000	156.000000	3.000000	0.000000	0.000000	0.00000
50%	0.640000	0.720000	4.000000	200.000000	3.000000	0.000000	0.000000	0.00000
75%	0.820000	0.870000	5.000000	245.000000	4.000000	0.000000	0.000000	0.00000
max	1.000000	1.000000	7.000000	310.000000	10.000000	1.000000	1.000000	1.00000

Check data for missing values (data quality)

```
In [14]:  1 missing_values = df.isnull()
          2 sns.heatmap(data = missing_values, yticklabels=False, cbar=False, cmap='viridis')
```

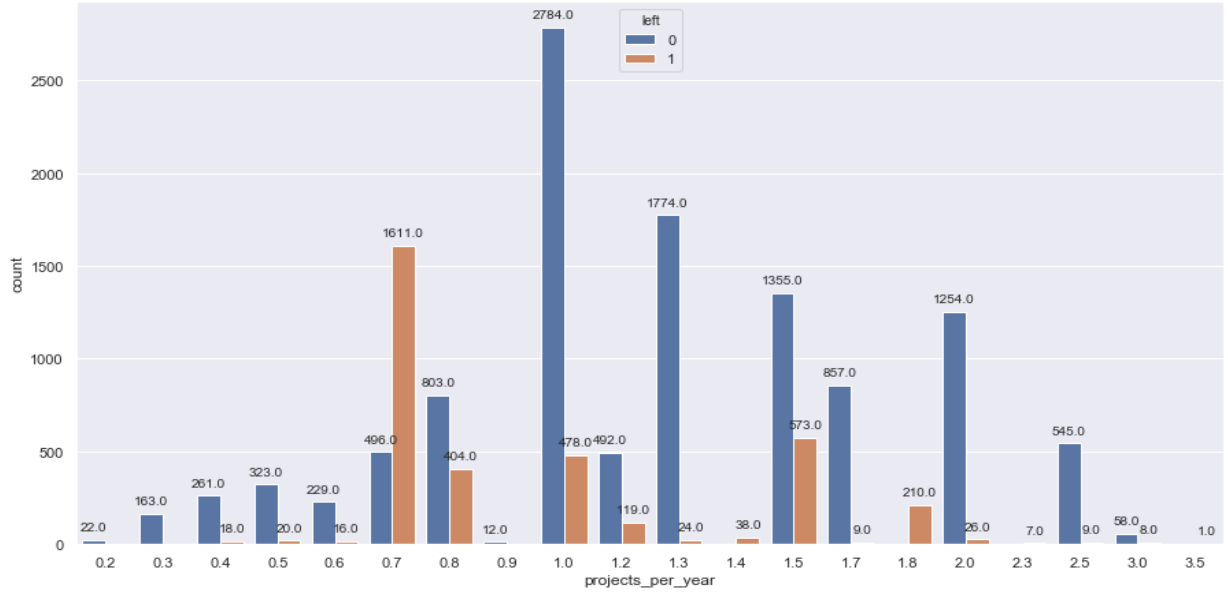
```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x1a22537278>
```



Construct Required Data

In our data, we decided to create our own derived variable to calculate a ratio based on the number of work projects completed based on the length of tenure in the organization. This ratio shows that there were a large number of employees that left where they worked on average 0.7 projects per year. This was undertaken by our data science team, as shown in Figure 8.

Figure 8 - Derived ratio of projects per year



Construct new dervived feature based on number of projects and time spent at company

```
In [41]:  1  df['projects_per_year'] = df.apply(lambda row: round(row.number_project / row.time_spend_company,1), axis=1)
```

This new derived ratio was added back into our CSV file for use in our algorithms.

Integrate Data

For our research purposes, we did not undertake any merging of data across tables and all aggregation of data was done solely in the data exploration phases of existing features.

We did not need to undergo any aggregations to our dataset for the purpose of our model.

Modeling Phase

Select Modeling Technique

Based on reviewing all of the features in our dataset, since we already have the binary output variable of whether someone stayed with the organization (0) or left (1), this would suggest that our model will be a form of supervised learning. As our goal is to create an algorithm to help us understand whether an employee will leave the organization by department, supervised learning will help devise a function (using our given sample data and output) to approximate the relationship between the input variables we identified and our output variable.

Our consultant team determined that a logistic regression is the best design for our research. A logistic regression is typically the suggested analysis to undertake when the dependent output variable (in our case whether the employee stayed or left the organization) is considered binary.

Based on our choice of logistic regression, there are some inherent assumptions to note[3]:

1. A logistic regression doesn't require a linear relationship to exist between our dependent variable (whether the employee left or not) and our independent variables.

[3] Statistics Solutions,'Assumptions of Logistic Regression', https://www.statisticssolutions.com/assumptions-of-logistic-regression/, Accessed October 24, 2019

2. Our dependent variable isn't being measured on an interval or ratio scale. In our case, our dependent variable is considered to be binary nominal.
3. Logistic regression requires that the independent variables not be too highly correlated with each other.

Generate Test Design

Prior to developing the model, we needed to transpose the department and salary variables from being categorical to numerical by utilizing get_dummies ().

Transpose Department column with prefix 'Dept'

In [21]:
```
dept = pd.get_dummies(df['Department'], prefix='Dept')
```

Transpose Salary column with prefix 'Salary'

In [22]:
```
salary = pd.get_dummies(df['salary'], prefix='Salary')
```

Combine Department and Salary dataframes into new dataframe for training

In [23]:
```
df2 = pd.concat([df, dept, salary], axis=1)
```

Drop insignificant features from the new combined dataframe

In [24]:
```
df2.drop(['Department', 'salary'], axis=1, inplace=True)
```

For our models, we looked to split out dataset into our "X" features (independent variables) and our "Y" output variable (left or not). Next, we split our data into a training and test set. This permitted our team to train our model using real data and help us evaluate our built models against the test data to help us isolate and identify any potential errors. Below is a screenshot from our notebook of this initial work, using all of the variables:

Training the Logistic Regression Model (all features)

In [26]:
```
# Split data into 'X' features and 'y' target label sets
x=df2[['satisfaction_level', 'last_evaluation', 'number_project', 'average_montly_hours', 'time_spend_company',
    'Work_accident', 'promotion_last_5years', 'projects_per_year', 'Dept_IT', 'Dept_RandD', 'Dept_accounting', 'Dept_hr'
    'Dept_support', 'Dept_technical', 'Salary_high', 'Salary_low', 'Salary_medium']]
y=df2['left']
```

In [27]:
```
# Import module to split dataset
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
# Split data set into training and test sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state = 42)
```

Build the Model

Next, our data scientists set out to build out the actual logistic regression model with both test and control data for our first model using all of the independent input variables:

In [28]:
```
# Import LogisticRegression model from sklearn
from sklearn.linear_model import LogisticRegression

# Apply training data to svc model
model_svc = LogisticRegression(solver='lbfgs', max_iter=2000)
model_svc.fit(x_train,y_train)
```

Out[28]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=2000,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

The team repeated the same work, this time only using the 4 select input variables identified earlier:

Training the Logistic Regression Model (4 features)

In [31]:
```
# Split data into 'X' features and 'y' target label sets
x2=df2[['satisfaction_level', 'last_evaluation', 'average_montly_hours', 'Dept_IT', 'Dept_RandD', 'Dept_accounting', 'De
    'Dept_support', 'Dept_technical']]
y2=df2['left']
```

In [32]:
```
# Import module to split dataset
from sklearn.model_selection import train_test_split
# Split data set into training and test sets
x2_train, x2_test, y2_train, y2_test = train_test_split(x2, y2, test_size=0.3, random_state=42)
```

In [33]:
```
# Import LogisticRegression model from sklearn
from sklearn.linear_model import LogisticRegression

# Apply training data to svc model
model_svc2 = LogisticRegression(solver='lbfgs', max_iter=1000)
model_svc2.fit(x2_train,y2_train)
```

Out[33]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=1000,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

Assess the Model

After developing both regression models, we checked the accuracy scores:

Algorithm #1 - All input variables

Print Model Classification Report and Accuracy Score (all features)

In [29]:
```
y_pred = model_svc.predict(x_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.87      0.93      0.90      3428
           1       0.71      0.57      0.63      1072

    accuracy                           0.84      4500
   macro avg       0.79      0.75      0.76      4500
weighted avg       0.83      0.84      0.84      4500
```

In [30]:
```
print(model_svc.score(x_test,y_test))
```

```
0.8415555555555555
```

Algorithm #2 - 4 input variables

Print Model Classification Report and Accuracy Score (4 features)

In [34]:
```
y2_pred = model_svc2.predict(x2_test)
print(classification_report(y2_test, y2_pred))
```

```
              precision    recall  f1-score   support

           0       0.80      0.93      0.86      3428
           1       0.51      0.25      0.33      1072

    accuracy                           0.76      4500
   macro avg       0.66      0.59      0.59      4500
weighted avg       0.73      0.76      0.73      4500
```

In [35]:
```
print(model_svc2.score(x2_test,y2_test))
```

```
0.7648888888888888
```

While both logistic regressions met our model criteria goal of an accuracy of 80% or greater, our data scientists decided to explore other supervised learning algorithms that might have an even higher accuracy prediction for our dataset. The data scientists identified that a linear Support Vector Machine (SVM/SVC) model might work in this case.

Revise Parameter Settings

The team then redid the exercise of developing the test and training data, again creating 2 algorithm versions. Below is a summary of the revised SVM/SVC algorithms our data team created:

Preparing the data for the SVM model by using get_dummies() function on salary and department

In [19]:
```
#Factorize department and salary
```

In [20]:
```
#df["DepartmentID"] = pd.factorize(df.Department)[0]
#df["SalaryID"] = pd.factorize(df.salary)[0]
```

Transpose Department column with prefix 'Dept'

In [21]:
```
dept = pd.get_dummies(df['Department'], prefix='Dept')
```

Transpose Salary column with prefix 'Salary'

In [22]:
```
salary = pd.get_dummies(df['salary'], prefix='Salary')
```

Combine Department and Salary dataframes into new dataframe for training

In [23]:
```
df2 = pd.concat([df, dept, salary], axis=1)
```

Drop insignificant features from the new combined dataframe

In [24]:
```
df2.drop(['Department', 'salary'], axis=1, inplace=True)
```

Training the model - Algorithm #1 all variables

Training the SVC Model (all features)

In [26]:
```
# Split data into 'X' features and 'y' target label sets
x=df2[['satisfaction_level', 'last_evaluation', 'number_project', 'average_montly_hours', 'time_spend_company',
    'Work_accident', 'promotion_last_5years', 'projects_per_year', 'Dept_IT', 'Dept_RandD', 'Dept_accounting', 'Dept_hr'
    'Dept_support', 'Dept_technical', 'Salary_high', 'Salary_low', 'Salary_medium']]
y=df2['left']
```

In [27]:
```
# Import module to split dataset
from sklearn.model_selection import train_test_split
# Split data set into training and test sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=101)
```

In [28]:
```
# Import svm model from sklearn
from sklearn import svm
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

# Apply training data to svc model
model_svc = svm.SVC(gamma='auto')
model_svc.fit(x_train,y_train)
```

Out[28]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Accuracy score - Algorithm #1 all variables

Print Model Classification Report and Accuracy Score (all features)

In [29]:
```
y_pred = model_svc.predict(x_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.98      0.96      0.97      3431
           1       0.88      0.93      0.90      1069

    accuracy                           0.95      4500
   macro avg       0.93      0.95      0.94      4500
weighted avg       0.95      0.95      0.95      4500
```

In [30]:
```
print(model_svc.score(x_test,y_test))
```

```
0.9528888888888889
```

Training the model - Algorithm #2 4 variables

Training the SVC Model (4 features)

In [31]:
```
# Split data into 'X' features and 'y' target label sets
x2=df2[['satisfaction_level', 'last_evaluation', 'average_montly_hours', 'Dept_IT', 'Dept_RandD', 'Dept_accounting', 'De
    'Dept_support', 'Dept_technical']]
y2=df2['left']
```

In [32]:
```
# Import module to split dataset
from sklearn.model_selection import train_test_split
# Split data set into training and test sets
x2_train, x2_test, y2_train, y2_test = train_test_split(x2, y2, test_size=0.3, random_state=101)
```

In [33]:
```
# Import svm model from sklearn
from sklearn import svm

# Apply training data to svc model
model_svc2 = svm.SVC(gamma='auto')
model_svc2.fit(x2_train,y2_train)
```

Out[33]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Accuracy score - Algorithm #2 4 variables

Print Model Classification Report and Accuracy Score (4 features)

In [34]:
```
y2_pred = model_svc2.predict(x2_test)
print(classification_report(y2_test, y2_pred))
```

```
              precision    recall  f1-score   support

           0       0.88      0.95      0.91      3431
           1       0.78      0.60      0.68      1069

    accuracy                           0.86      4500
   macro avg       0.83      0.77      0.80      4500
weighted avg       0.86      0.86      0.86      4500
```

In [35]:
```
print(model_svc2.score(x2_test,y2_test))
```

```
0.8646666666666667
```

Evaluation Phase

Evaluate Results

After reviewing both of the revised algorithms (built using SVM/SVC), we saw that the algorithm leveraging all input variables resulted in a prediction accuracy of 95% and a precision of 88% against those individuals who left. As previously noted, when our data team ran the same algorithm but only utilizing 4 variables, it resulted in lower prediction accuracy (86%), which would suggest that there are other variables outside of department, satisfaction level, average monthly hours and last evaluation that exert some degree of influence on someone's decision to depart the organization.

Referring back to our outlined model success criteria, we sought to get a prediction accuracy of 80% or greater, with a precision of at least 65%. Based on our results, we were able to surpass the criteria when we utilized the SVC model and used all available input variables.

Approve the Model

Based on the results, we would recommend that the organization look to implement our SVC algorithm utilizing all input variables, as this was the model that yielded the highest accuracy score.

Review Process

After reviewing the overall process of our research project, while we were able to devise a predictive algorithm to help solve for employee churn, we were not entirely successful in pinpointing the exact mix of input variables that would result in the highest accuracy.

We would recommend that further refinement and exploration of the input variables be carried out, as this would provide more clarity for the HR department to understand the variables that are the greatest indicator that an existing employee may soon opt to leave the organization.

We would also suggest to further train the model if there were even more employee records than what was provided to our consultant team, as it may help with the overall accuracy and precision.

Determine Next Steps

The next possible steps for the research project are as outlined below:

Action	Pros	Cons
Continue to refine the input variables and retest the model	This would permit our team to continue to uncover what unique combination of inputs has the most impact on the algorithm's accuracy output	This would require additional resources in terms of time and funding
Look to uncover if there are other models better suited to solving for prediction	Potential to devise an even more accurate algorithm	This would require additional resources in time and funding
Proceed forward with the existing algorithm	This would result in no further delays to the timeline or require additional costs	N/A

After weighing the different outcomes, we recommend proceeding forward with the algorithm deployment, as this would permit the organization to implement it in accordance with their specified timeline and incur no additional costs from our consultant team.

Deployment Phase

Deployment Plan

As outlined previously, the deployment is currently set for November 3rd, provided that the organization's leadership team approves after reviewing our report and presentation. Our data scientists will be working with the tech team to implement the algorithm into their HR department's software stack. At this point, it is estimated that the deployment will require 5 business days to implement.

Monitoring & Maintenance

Monitoring of the algorithm will be overseen by the organization's tech and IT departments. Our consultants will be available on an as-needed basis if there are any irregular issues that cannot be resolved by the organization's staff. Any outside assistance will be charged at an hourly rate by our team.

Final Presentation & Report

The presentation of our research findings is set for November 3rd. This report, which shall represent the final project report, will be delivered on November 3rd, along with our Jupyter notebooks containing the algorithm coding.

Review Project

Based on the outcome of our research, we recommend that further time be allocated towards refining the algorithms and determining the exact mix of input variables that get the highest accuracy towards predicting employee churn. It is the hope of our consultation team that the organization is able to better understand the key drivers of what is causing their higher than average employee turnover across the company and in select departments that are seeing closer to nearly a third of the employees leaving.

The outcome of our research should help HR and the leadership team be better poised to understand and counteract employee dissatisfaction that may be leading to the higher exodus, implement better strategies to reward high performing staff and devise new programs for recruitment of new talent.

Our team has greatly appreciated the patience and efforts of the organization while we worked to solve for this unique business challenge.

Works Cited

1. Workopolis,'How to calculate cost to hire (and why it's important)', March 27, 2018, https://hiring.workopolis.com/article/how-to-calculate-cost-to-hire-and-why-its-important/, Accessed October 20, 2019

2. Nicole Roder, Zenefits,'Does Your Company Have a Healthy Employee Retention Rate?', April 4, 2019, https://www.zenefits.com/workest/your-company-healthy-employee-retention-rate/, Accessed October 21, 2019

3. Statistics Solutions, 'Assumptions of Logistic Regression', https://www.statisticssolutions.com/assumptions-of-logistic-regression/, Accessed October 24, 2019