Attrition …
Why and When!?

Marty Fromuth

GW Bootcamp

Jun 2020– Jan 2021

Overview

The What

What is the topic?
What are we trying to learn?
What is the data?

The Why

Why did we choose this topic?
Why did we choose this data?

The How

How are we conducting data exploration?
How are we analyzing the data?
How are we storing the data?
How are we displaying our work?

The Results

The What: What is the topic?

- Primary Topic: Attrition, aka people leaving a company
  - Companies want to know when and why employees decide to leave a company, or ’attrit’
  - Employers maintain information on job performance and compensation
  - Companies also put out anonymous surveys on satisfaction
  - A company’s attrition ‘cycle’ can manifest in waves, a consistent trickle, or unpredictable cliffs

- Secondary Topic: Attrition in different types or categories of jobs
  - Most companies have smaller ‘communities’ based on job type or category
  - STEM, soft-skills and/or HR, and leadership positions may have different priorities
  - Like overall employees, attrition can manifest in different ways

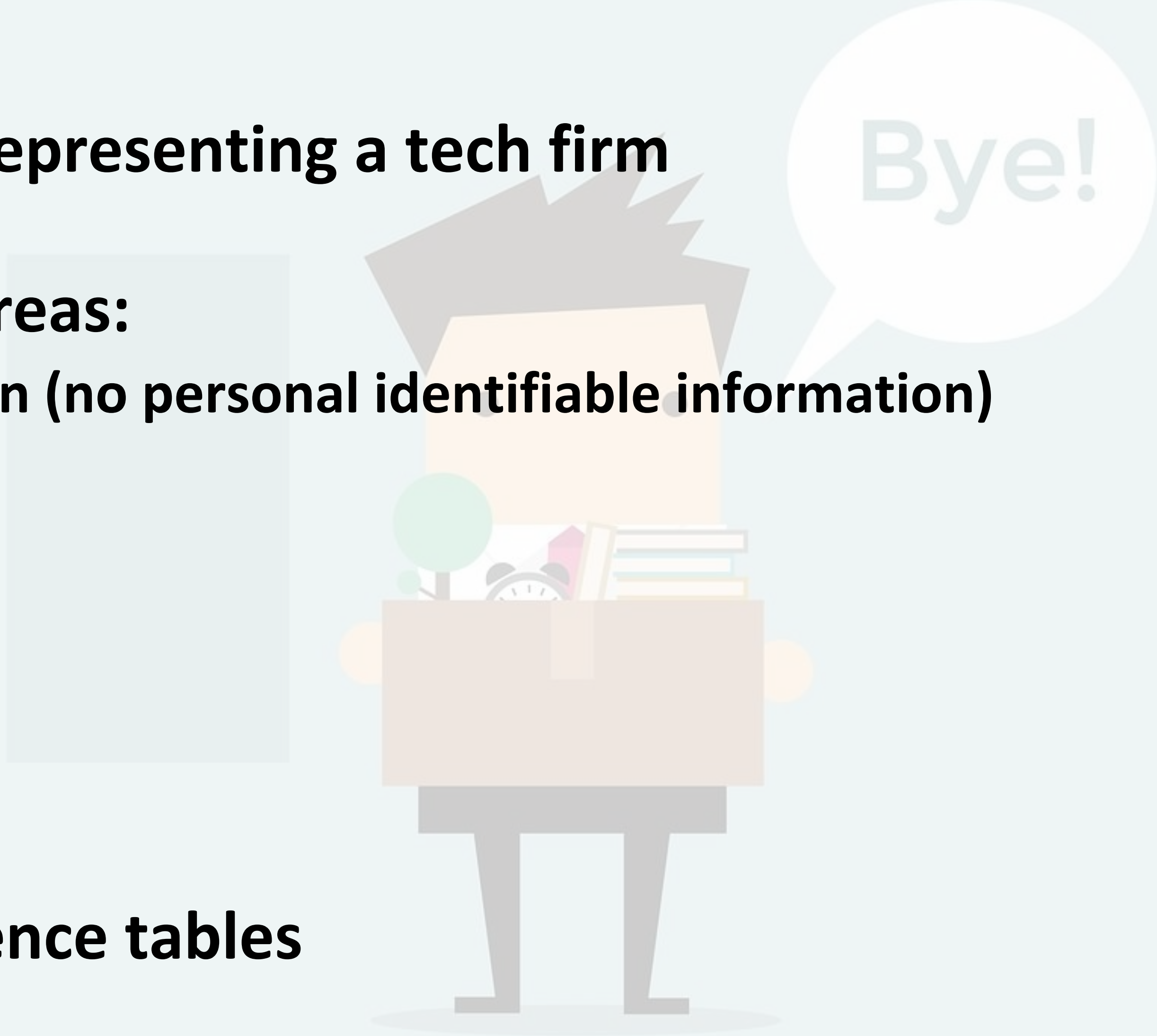
The What: What are we trying to learn?

- **Primary questions:**
  - **What are they key features that predict attrition?**
  - **When and how many employees do we predict will leave?**
  - **Do those answers change based on the job category?**

- **Additional insights:**
  - **What, if any, correlation is there between different features?**
  - **Are there any major differences in features between the job categories?**

The What: What is the data?

- Synthetic, anonymous HR data representing a tech firm
- Multiple fields/features in key areas:
  - Anonymized employee information (no personal identifiable information)
  - Job history
  - Education
  - Job role and department
  - Salary
  - Satisfaction scores
- 1 primary table with eight reference tables



The What: What is the data?

- 1470 rows / unique employees
- Key = ‘EmployeeNumber’
- Dropped = ‘EmployeeCount’, ‘StandardHours’, ‘Over18’, ‘MonthlyIncome’, ‘HourlyRate, ‘DailyRate’, ‘Department’, ‘TotalWorkingYears
- Created new column = ‘JobCategory’
  - Leadership
  - Non-Tech
  - Tech
- Left join main table w/ satisfaction ratings

The What: What is the data?

Combined:
attrition_combined_text (Postgres)
df_attrition_encoded (Python)

Non-Tech:
attrition_nontech_text (Postgres)
df_attrition_nontech_encoded (Python)

Tech:
attrition_tech_text (Postgres)
df_attrition_tech_encoded (Python)

Leadership:
attrition_ldrshp_text (Postgres)
df_attrition_ldrshp_encoded (Python)

- **Created two main dataframes**
  - **Encoded dataframe – used for machine learning model building and inference, and feature analysis**
  - **Visualization dataframe – used for the dashboard and to visualize data for the end user**
    - **Created tables in Postgres using SQL join & 'INTO'**
    - **Stored these in new Postgres database**
- **Created three sub-dataframes for each of the main dataframes**
  - **Leadership**
  - **Tech**
  - **Non-tech**
- **Build & stored 8 dataframes in total**

The Why: Why did we choose this topic?

- **Companies invest a lot in their employees …**

- **Companies want to build programs to keep employees**
  - **Which feature is most common among those who leave □ build a program for all employees related to that feature**
  - **Do those features change based on the job role □ build programs tailored for job roles or job categories**

- **Companies understand attrition is part of life but want to be able to minimize risk of shortage**
  - **When will they leave □ time recruiting efforts**
  - **How many can we expect to leave □ drive size of recruiting**
  - **Which roles will they leave from □ focus recruiting**

The Why: Why did we choose this data?

- Mixture of attrition results
  - People who stayed …
  - People who left …
- Includes multiple features in critical categories:
  - Personal background and information (not Personally Identifiable Info)
  - History and current employment information and performance
  - Survey results measuring an employees’ relative satisfaction at a given time
- Missing additional features that could be helpful
  - Time/date of survey
  - Date of resignation
  - Multiple survey results to create models for prediction over time

The How: How are we storing the data?

pgAdmin 4.24 with Postgres 12.4

- Original tables
  - IBMEmployeeAttrition
  - Rating and/or Satisfaction score explanation
- New tables created from ETL process in Python
  - Added ‘JobCategory’ field
  - Incorporated text fields from satisfaction/rating tables
  - Created 3x sub-tables for each JobCategory type

Python w/ Jupiter Notebook

- SQLAlchemy to upload original table from Postgres; used for ETL process
- Export new attrition dataframe to Postgres via SQLalchemy
- Maintained encoded dataframes for data exploration, model build & training, and inference assessments

The How: How are we conducting data exploration?

- Explore imbalance in the target
- Explore correlation between features
- Explore distribution of the features
- Explore distribution of features to one another

The How: How are we analyzing the data?

- **Want to predict if someone will attrit or not = classifier**
- **Highly imbalanced target dataset = sampling technique need**
- **Highly variable feature results = needs normalization**
- **Feature data is not evenly distributed = potential weak learners**
- **Large number of variables/features = Random Forest**
- **Rank the importance of features/variables = Random Forest**

IDEAL ALGORITHM: Balanced Random Forest Ensemble

TARGET: Attrition (‘Yes’, ‘No’)

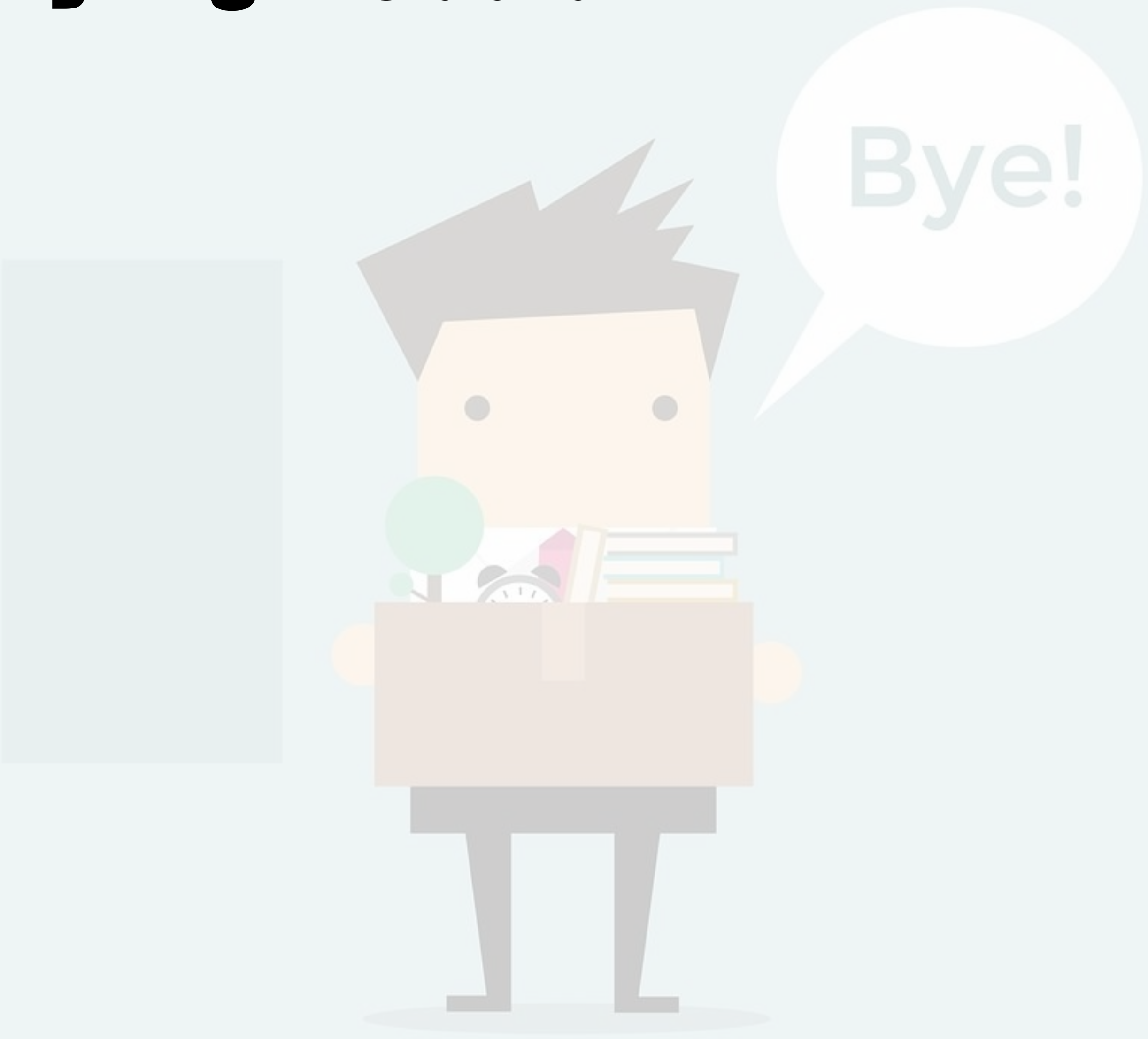
The How: How are we analyzing the data?

- Strengths of Balanced Random Forest Classifier
  - Based on bagging ensemble technique for machine learning; helps improve accuracy and robustness of ‘weak learners’
  - Good for categorical as well as well as continuous variables
  - Does not require feature scaling; robust to outliers
  - Incorporates techniques to balance an unbalanced target feature
  - Robust against overfitting; weak learners trained on different data
  - Good to rank importance of features in prediction of outcome

- Weaknesses of Balanced Random Forest Classifier
  - Complexity makes it challenging for explainability
  - Generally requires longer training time

The How: How are we analyzing the data?

- Train/Test Split:
    - 70 train // 30 test
    - Sklearn.model



The How: How are we displaying our work?

- Main tool: JavaScript + HTML + CSS

- Graph displays: Tableau

The How: How are we displaying our work?

Bye, Bye, Bye!

Attrition: Why and When?

*Selection Bar: **Overview** // Data Analysis // Machine Learning Model Assessments // Feature Analysis // Conclusions*

Dataframe:

Drop-down & update with different dataframes (combo, tech, non-tech, leadership)

Overview of the project: The Why

Overview of the project: The How

Overview of the project: The Overall Results

The How: How are we displaying our work?

Bye, Bye, Bye!

Attrition: Why and When?

*Selection Bar: Overview // **Data Analysis** // Machine Learning Model Assessments // Feature Analysis // Conclusions*

Dataframe:

Drop-down & update with different dataframes (combo, tech, non-tech, leadership)

Distribution Display:

Tableau visualization w/ drop-downs for each feature AND drop-down for JobCategory

Distribution Display:

Tableau visualization w/ Comparison of Features

Overall Assessments of Data Analysis:

Text

The How: How are we displaying our work?

Bye, Bye, Bye!

Attrition: Why and When?

*Selection Bar: Overview // Data Analysis // **Machine Learning Model Assessments** // Feature Analysis // Conclusions*

Dataframe:

Drop-down & update with different dataframes (combo, tech, non-tech, leadership)

Graphic Display:

Confusion Matrix, Accuracy, & Imbalance Report w/ drop down for the model used AND the sub-category

Overall Assessments of the ML Model:

Text

The How: How are we displaying our work?

Bye, Bye, Bye!

Attrition: Why and When?

*Selection Bar: Overview // Data Analysis // Machine Learning Model Assessments // **Feature Analysis** // Conclusions*

Dataframe:

Drop-down & update with different dataframes (combo, tech, non-tech, leadership)

Graphic Display:

Comparison of each of the features importance from the classifier

Overall Assessments of the Feature Analysis & Comparison:

Text

The How: How are we displaying our work?

Bye, Bye, Bye!

Attrition: Why and When?

Selection Bar: Overview // Data Analysis // Machine Learning Model Assessments // Feature Analysis // ***Conclusions***

Dataframe:

Drop-down & update with different dataframes (combo, tech, non-tech, leadership)

Overall Conclusions on Our Questions:

Text

Recommendations for Next Steps:

Text

The Results: Data Exploration - Overall Findings

- **Highly imbalanced dataset, especially within leadership dataframe**
  - **Target features imbalanced**
  - **Feature distribution is highly variable; leadership often older & paid more**

- **Data correlation is relatively even; most correlations expected**
  - **Positive:**
    - **Age → TotalWorkingYears / MonthlyIncome / JobLevel**
    - **MonthlyIncome → YearsAtCompany / TotalWorkingyears**
    - **PerformanceRating → PercentSalaryHike**
    - **JobLevel → YearsAtCompany / TotalWorkingYears / MonthlyIncome**
  - **Nevative: JobCategory → JobLevel / MonthlyIncome / TotalWorkingYears**
  - **Unexpected: MaritialStatus / StockOptionLevel (negative)**

Explore imbalance in the target

```
Combined:
No      1233
Yes      237
Name: Attrition, dtype: int64
Tech:
No      564
Yes     118
Name: Attrition, dtype: int64
Non-Tech:
No      359
Yes     102
Name: Attrition, dtype: int64
Leadership:
No      310
Yes      17
Name: Attrition, dtype: int64
```

The Results: Data

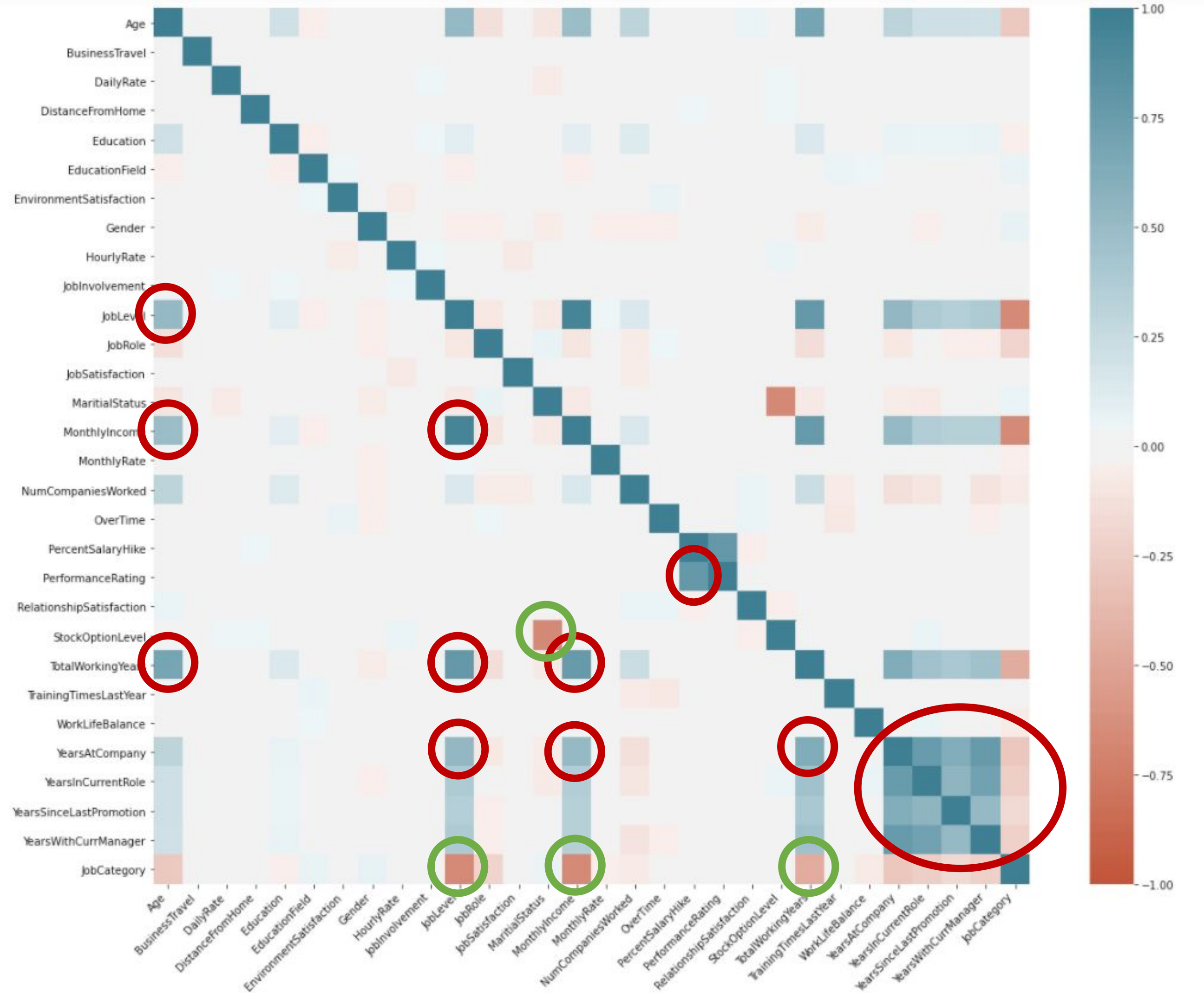
- **Target data for combined and three sub-categories are all imbalanced**
  - **Leadership most imbalanced**
  - **Non-tech least imbalanced**
- **Will require sampling technique to correct for imbalance in target data**

The Results: Data

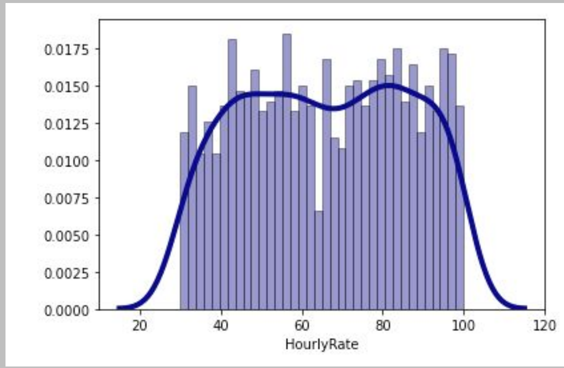
- **Multiple positive correlations:**
  - Age: MonthlyIncome / JobLevel / TotalWorkingYears
  - JobLevel: YearsAtCompany / TotalWorkingYears / MonthlyIncome
  - MonthlyIncome: YearsAtCompany / TotalWorkingYears
  - PercentSalaryHike: PerformanceRating
  - TotalWorkingYears: YearsAtCompany

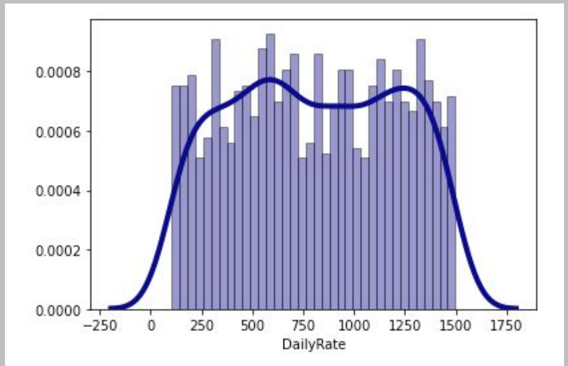
- **Multiple negative correlations:**
  - JobLevel: JobCategory
  - MaritialStatus: StockOptionLevel
  - MonthlyIncome: JobCategory
  - TotalWorkingYears: JobCategory

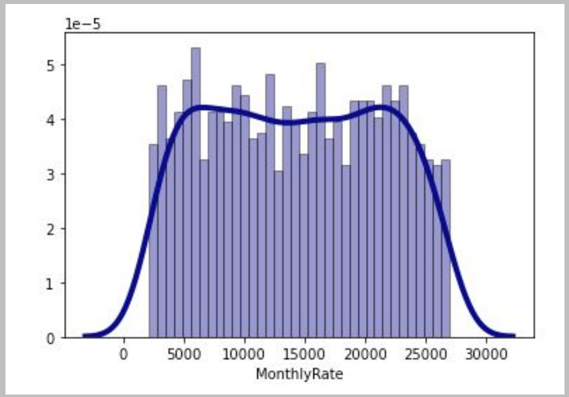
- **Features representing similar/same information:**
  - HourlyRate / DailyRate / MonthlyRate / MonthlyIncome
  - Age / TotalWorkingYears

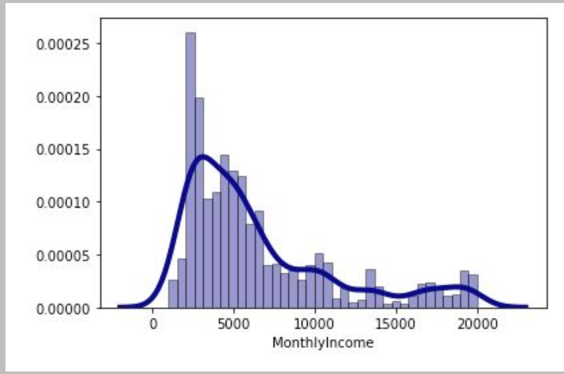


The Results: Data









- **Four features provide the same info; built new dataframe to assess similarity between data**

- **Assessed distribution to determine which feature would remain, included in model**
  - **MonthlyIncome – least normally distributed**
  - **HourlyRate, DailyRate – normally distributed**
  - **MonthlyRate – most normally distributed**

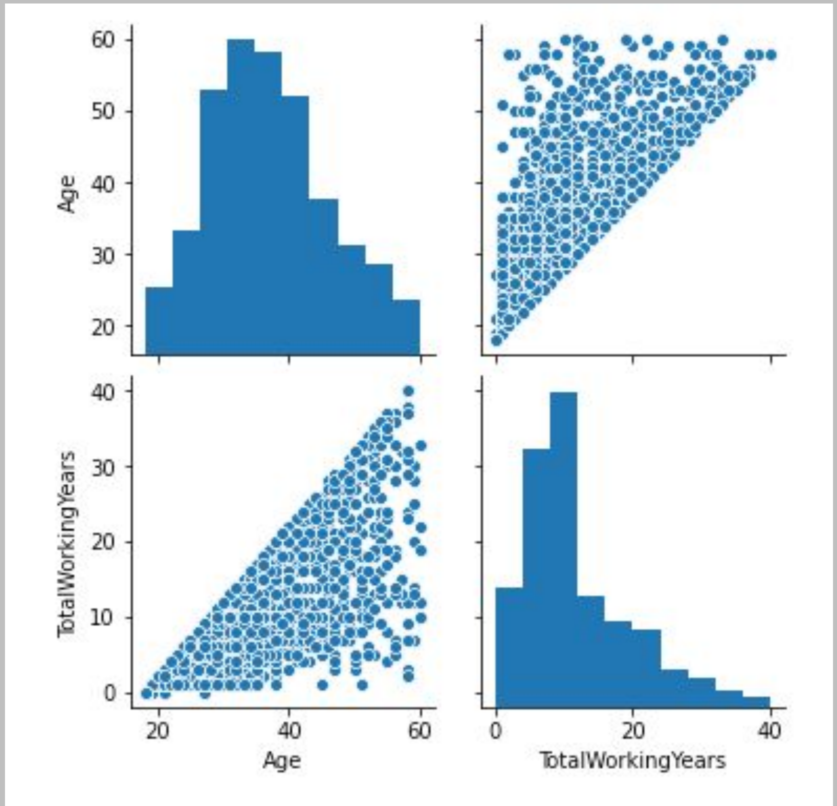
Dropping HourlyRate, DailyRate; MonthlyRate & MonthlyIncome used in model development

The Results: Data

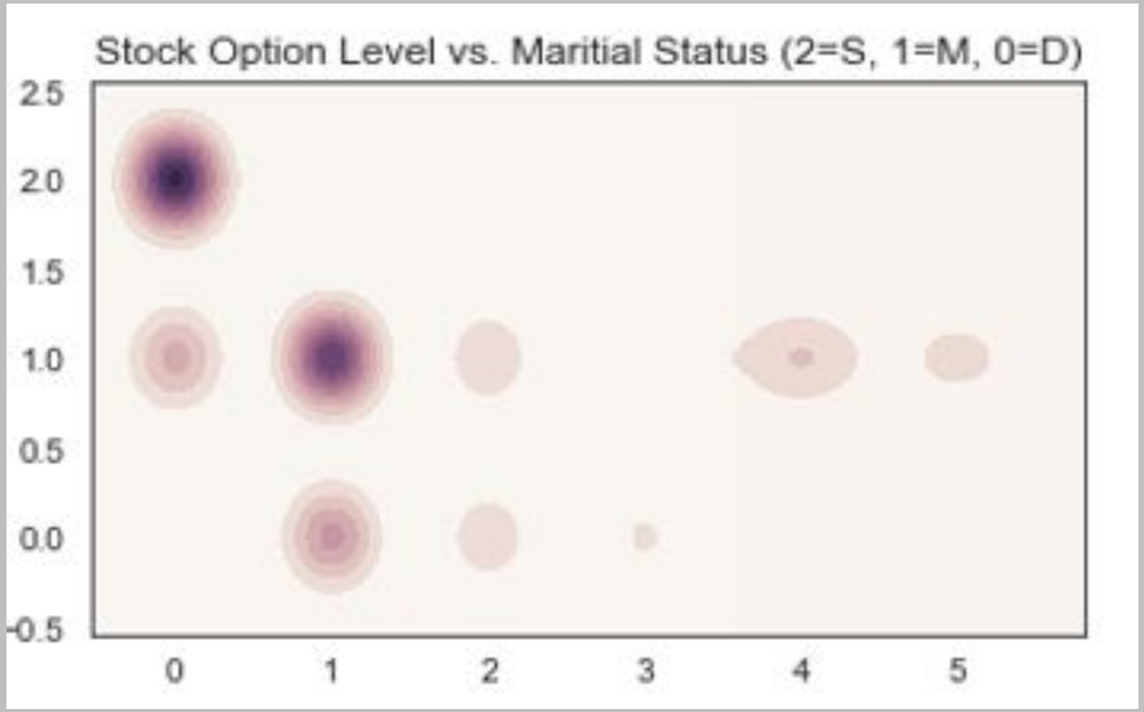
	Age	TotalWorkingYears	Difference
count	1470.000000	1470.000000	1470.000000
mean	36.923810	11.279592	25.644218
std	9.135373	7.780782	6.875481
min	18.000000	0.000000	18.000000
25%	30.000000	6.000000	20.000000
50%	36.000000	10.000000	24.000000
75%	43.000000	15.000000	30.000000
max	60.000000	40.000000	56.000000

- Age and TotalWorkingYears appear to highlight similar info, but not the same
  - Used describe and distribution analysis
  - Both skewed left, but still independent

Maintain both features for the model development and inference



The Results: Data

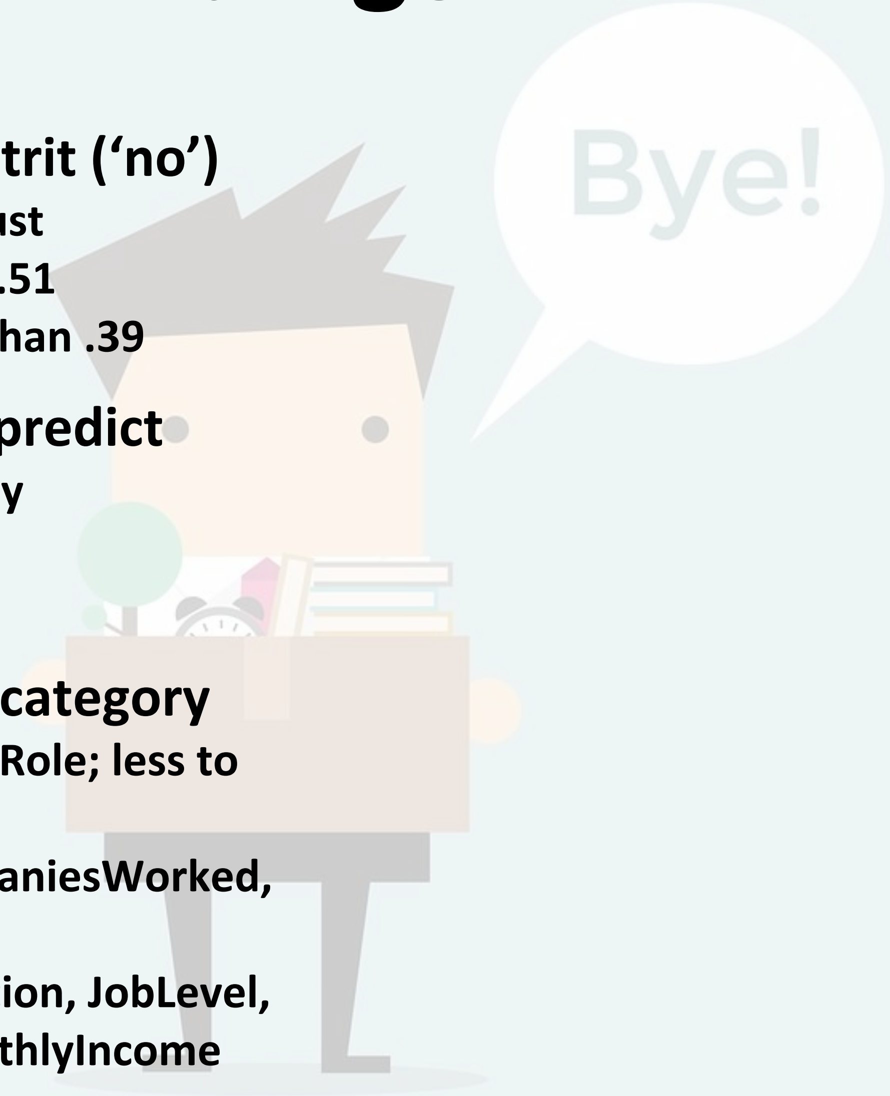


- **Single employees tend to have lower stock levels**
- **Married employees appear to have higher levels**
- **Divorced employees fall between the two above**

Younger employees are often single vs. married employees; some employees who are divorced get remarried later in life

The Results: Analysis - Overall Findings

- Ideal model use is to predict employees who do NOT attrit ('no')
  - Model prediction of employees who attrition ('yes') is not robust
  - F1 scores for categories and overall employees no higher than .51
  - Largest deficiency in 'yes' for attrition is precisions; no higher than .39
- Leadership imbalance too large; not able to accurately predict
  - Poor ability to predict employees who attrit/leave the company
  - F1 scores no higher than .06 for 'yes' to attrition target feature
  - Model appears to be overfitting for predicting 'no' for attrition
- Feature importance in attrition does vary based on job category
  - Tech more sensitive to age, TotalWorkingYears,YearsInCurrentRole; less to YearsWithCurrManager, JobRole
  - Non-tech more sensitive to StockOptionLevel, NumberofCompaniesWorked, DistanceFromHome; less to YearsWithCurrManager
  - Leadership more sensitive to YearsSinceLastPromotion, Education, JobLevel, RelationshipSatisfaction, PerformanceRating; less to Age, MonthlyIncome



The Results: Analysis - Combo

- Precision – Where all the ‘yes/no’s right?
  - ‘Yes’: Worse than a coin toss
  - ‘No’: Fairly accurate
- Recall – Did we get all the ‘yes/no’s?
  - ‘Yes’: Fairly accurate
  - ‘No’: Fairly accurate
- Overfitting –
  - ‘Yes’: Potentially overfit
  - ‘No’: Not likely overfit

Overall, no better than a coin toss at predicting employees who left, but far better at predicting those who stayed

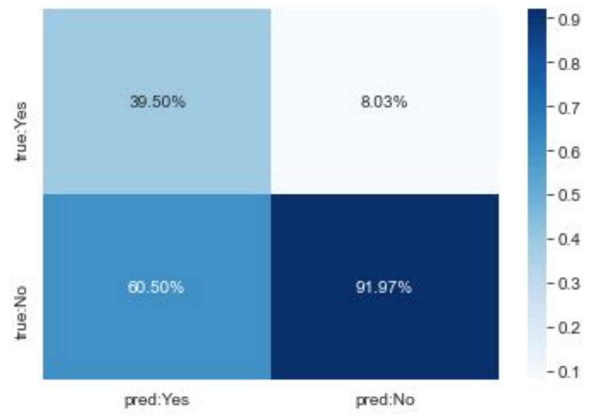
```
Combination-BRFC: accuracy - 0.7311449397530619
                   pre       rec       spe        f1       geo       iba       sup

         No       0.92      0.76      0.70      0.83      0.73      0.54       301
        Yes       0.39      0.70      0.76      0.51      0.73      0.53        67

avg / total       0.82      0.75      0.71      0.77      0.73      0.54       368


Combination-BRFC:
          pred:Yes  pred:No
true:Yes        47       20
true:No         72      229

<matplotlib.axes._subplots.AxesSubplot at 0x7fd8cf1755b0>
```



The Results: Analysis - Tech

- **Precision – Where all the ‘yes/no’s right?**
  - **‘Yes’: Worse than a coin toss**
  - **‘No’: Fairly accurate**
- **Recall – Did we get all the ‘yes/no’s?**
  - **‘Yes’: Fairly accurate**
  - **‘No’: Fairly accurate**
- **Overfitting –**
  - **‘Yes’: Potentially overfit**
  - **‘No’: Not likely overfit**

Overall, worse than a coin toss at predicting employees who left, but far better at predicting those who stayed

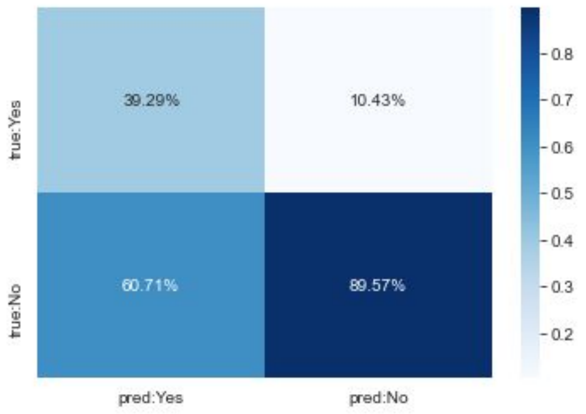
```
Tech-BRFC: accuracy - 0.69944182052383
                   pre       rec       spe        f1       geo       iba       sup

         No       0.90      0.75      0.65      0.82      0.70      0.49       137
        Yes       0.39      0.65      0.75      0.49      0.70      0.48        34

avg / total       0.80      0.73      0.67      0.75      0.70      0.49       171


Tech-BRFC:
          pred:Yes  pred:No
true:Yes        22       12
true:No         34      103

<matplotlib.axes._subplots.AxesSubplot at 0x7fd8cef3b880>
```



The Results: Analysis - Non-Tech

- Precision – Where all the ‘yes/no’s right?
  - ‘Yes’: Worse than a coin toss
  - ‘No’: Fairly accurate
- Recall – Did we get all the ‘yes/no’s?
  - ‘Yes’: Fairly accurate
  - ‘No’: Fairly accurate
- Overfitting –
  - ‘Yes’: Potentially overfit
  - ‘No’: Not likely overfit

Overall, worse than a coin toss at predicting employees who left, but far better at predicting those who stayed

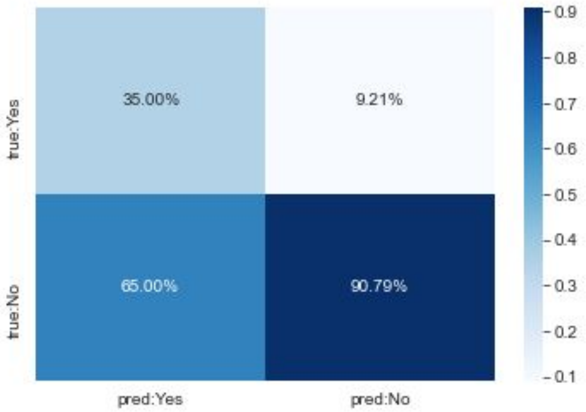
```
NonTech-BRFC: accuracy - 0.6964912280701754
                   pre       rec       spe        f1       geo       iba       sup

         No       0.91      0.73      0.67      0.81      0.70      0.49        95
        Yes       0.35      0.67      0.73      0.46      0.70      0.48        21

avg / total       0.81      0.72      0.68      0.74      0.70      0.49       116


NonTech-BRFC:
          pred:Yes  pred:No
true:Yes        14        7
true:No         26       69

<matplotlib.axes._subplots.AxesSubplot at 0x7fd8cf0f76d0>
```



The Results: Analysis - Non-Tech

- Precision – Where all the ‘yes/no’s right?
  - ‘Yes’: Worse than a coin toss
  - ‘No’: Fairly accurate
- Recall – Did we get all the ‘yes/no’s?
  - ‘Yes’: Fairly accurate
  - ‘No’: Worse than a coin toss
- Overfitting –
  - ‘Yes’: Potentially overfit
  - ‘No’: Not likely overfit

Overall, although accuracy is better, the F1 score is no better than a coin toss at predicting people who leave; ability to predict those who stay is less accurate than the BRFC model

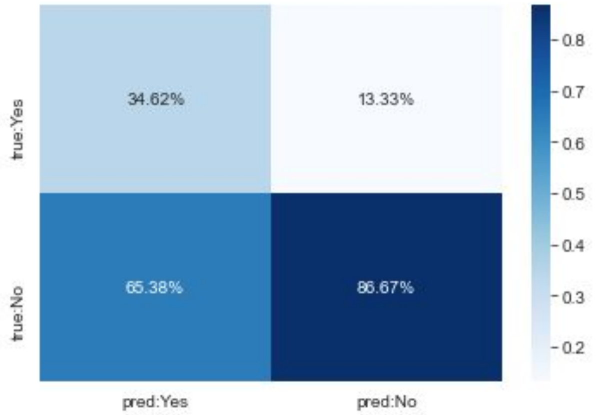
```
NonTech-SMOTEENN+RFC: accuracy - 0.75
                   pre       rec       spe        f1       geo       iba       sup

         No       0.87      0.82      0.43      0.84      0.59      0.37        95
        Yes       0.35      0.43      0.82      0.38      0.59      0.34        21

avg / total       0.77      0.75      0.50      0.76      0.59      0.36       116


NonTech-SMOTEENN+RFC:
          pred:Yes  pred:No
true:Yes         9       12
true:No         17       78

<matplotlib.axes._subplots.AxesSubplot at 0x7fd8cf042610>
```



The Results: Analysis - Ldrshp

- Precision – Where all the ‘yes/no’s right?
  - ‘Yes’: Completely unreliable
  - ‘No’: Fairly accurate
- Recall – Did we get all the ‘yes/no’s?
  - ‘Yes’: Only right ¼ of the time
  - ‘No’: Slightly better than a coin toss
- Overfitting –
  - ‘Yes’: Likely overfit
  - ‘No’: Possibly overfit

Overall, completely unable to predict employees who left, but far better at predicting those who stayed, aka did not attrit

```
Leadership-BRFC: accuracy - 0.4391025641025641
                    pre       rec       spe        f1       geo       iba       sup

         No        0.94      0.63      0.25      0.75      0.40      0.16        78
        Yes        0.03      0.25      0.63      0.06      0.40      0.15         4

avg / total        0.90      0.61      0.27      0.72      0.40      0.16        82


Leadership-BRFC:
          pred:Yes  pred:No
true:Yes         1        3
true:No         29       49

<matplotlib.axes._subplots.AxesSubplot at 0x7fd8cf03d8e0>
```

The Results: Analysis - Ldrshp

- Precision – Where all the ‘yes/no’s right?
  - ‘Yes’: Completely unreliable
  - ‘No’: Fairly accurate
- Recall – Did we get all the ‘yes/no’s?
  - ‘Yes’: Completely unreliable
  - ‘No’: Fairly reliable
- Overfitting –
  - ‘Yes’: Likely overfit
  - ‘No’: Unlikely overfit

Overall, far less able to predict employees who left as opposed to BRFC; although predicted employees who stayed better, it is still an unreliable model

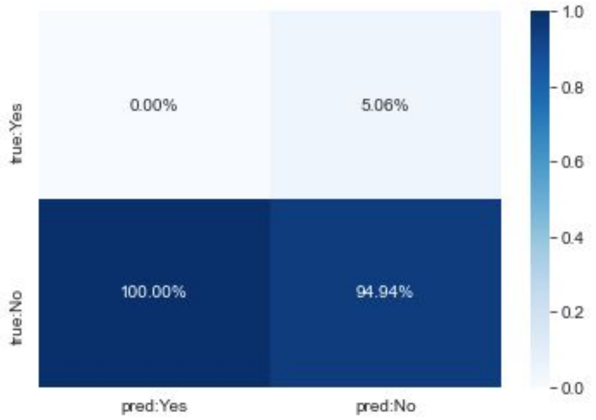
Leadership-SMOTEENN+RFC: accuracy - 0.9146341463414634

	pre	rec	spe	f1	geo	iba	sup
No	0.95	0.96	0.00	0.96	0.00	0.00	78
Yes	0.00	0.00	0.96	0.00	0.00	0.00	4
avg / total	0.90	0.91	0.05	0.91	0.00	0.00	82

Leadership-SMOTEENN+RFC:

	pred:Yes	pred:No
true:Yes	0	4
true:No	3	75

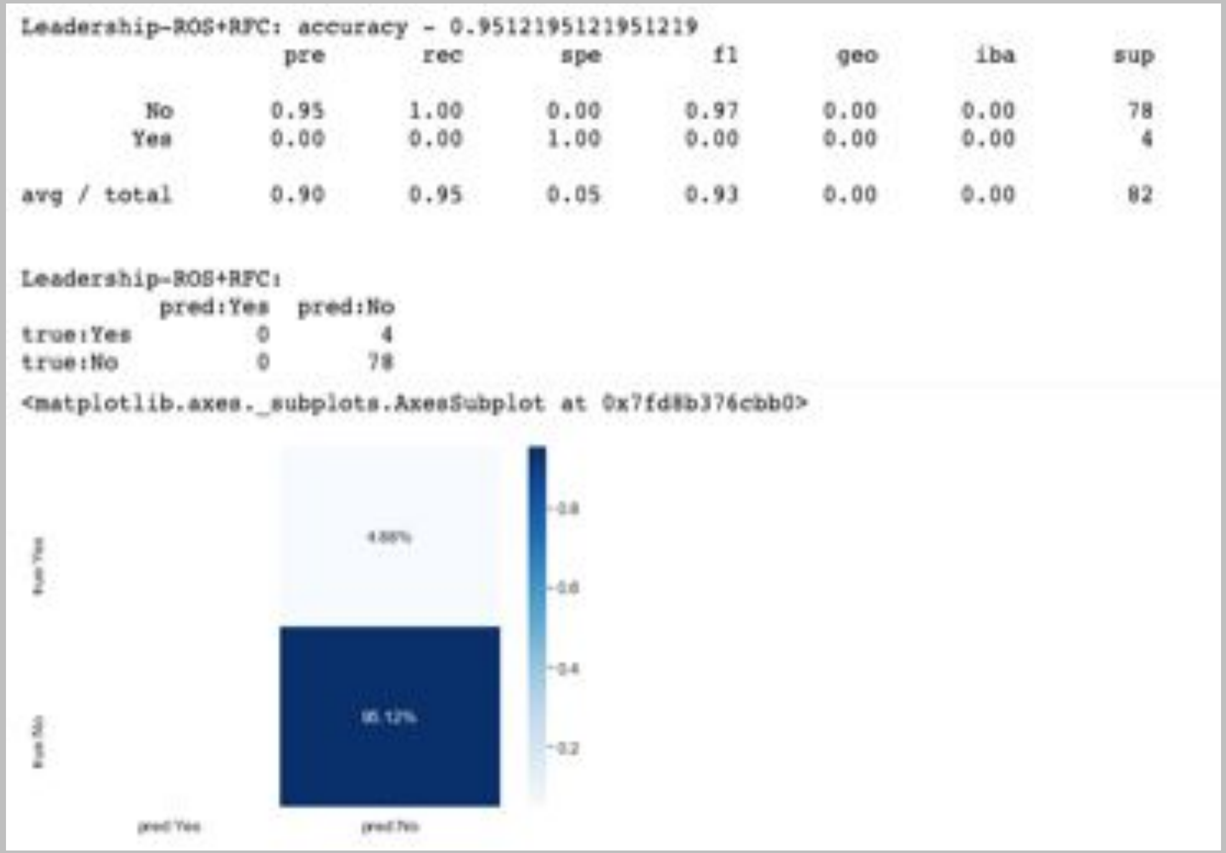
<matplotlib.axes._subplots.AxesSubplot at 0x7fd8ceffe970>



The Results: Analysis - Ldrshp

- Precision – Where all the 'yes/no's right?
  - 'Yes': Completely unreliable
  - 'No': Fairly accurate
- Recall – Did we get all the 'yes/no's?
  - 'Yes': Completely unreliable
  - 'No': Fairly reliable
- Overfitting –
  - 'Yes': Likely overfit
  - 'No': Unlikely overfit

Overall, far less able to predict employees who left as opposed to BRFC; although predicted employees who stayed better, it is still an unreliable model



The Results: Analysis - Feature Comparison