byrdwcrawford / HR_Analytics

Code Issues Pull requests Actions Projects Wiki Security Insights Setting

0 stars 0 forks 1 watching 1 Branch 0 Tags Activity

Public repository

1 Branch 0 Tags Go to file t Go to file Add file Code

byrdwcrawford almost done beb60b5 · 1 minute ago

Notebooks	updates	yesterday
PDFs	almost done	1 minute ago
images	almost done	1 minute ago
.gitignore	Initial commit	last week
HR_Model_Final.ipynb	almost done	1 minute ago
README.md	almost done	1 minute ago
hr_data.csv	Initializing	last week

README

HR_Analytics

Will Byrd

Flatiron School 2024

Introduction

In this notebook, we will take a look at an HR Analytics dataset and perform a binary classification to determine Attrition. Attrition is the departure of an employee from an organization fro any reason. In this dataset we do not have context on why the emlpoyee leaves the company-they may have been fired or they may have resigned.

Why is this notebook/repo important?

This dataset can be generalized quite well to other companies across America. First off, this company has a very traditional structure:

- Revenue generating department
  - Sales
- Non-Revenue generating department
- Research & Development
- Human Resources department

Secondly, the attrition rate for this company is roughly 16% which is very comparable to the national average which can be found here.

Lastly, this company has a lot of employees that are under 40 years old (based on Median and Mode). This gives every indication that this company data is generalizable to the rest of the country, based on data from the US Bureau of Labor Statistics.

The work force has changed drastically since 2019. Some companies are more flexible with work from home options and some companies expect employees to always be available. All of this is taking place on the heels of a global pandemic that interrupted lives. The impact that lockdown had on students, young professionals, and veteran employees will not be fully understood for years. Hopefully this repo will help stimulate more discussion on what needs to be done to increase employee retention and create a happier working environment for everyone.

Business Understanding

For companies and orgnizations-the employees are the most important asset. Therefore, it is vitally important to be able to predict behavior of these employees. Knowing which employees will stay with the company can:

- Reduce turnover costs
  - Hiring and firing employees is expensive
- Increase employee engagement and morale
  - It can be difficult to maintain strong company culture if turnover is high
- Assist with resource planning
  - Companies can add resources to employees they believe will contribute to longterm growth
  - Conversely, companies can take proactive measures to retain at risk employees
- Improve customer relationships
  - Customer percieve companies that retain talent more positively
- Enhance company reputation
  - Company brand is often times tied to the employees interacting with the customers

Goal

The main goal of our model is to maximize **accuracy**. Accuracy will be important here as being able to accurately predict which employees will stay and which will leave is important. It's also important to understand that since we don't know if employees are fired and which ones quit, simply targeting the positive Attrition class (Recall) will still not tell the entire story.

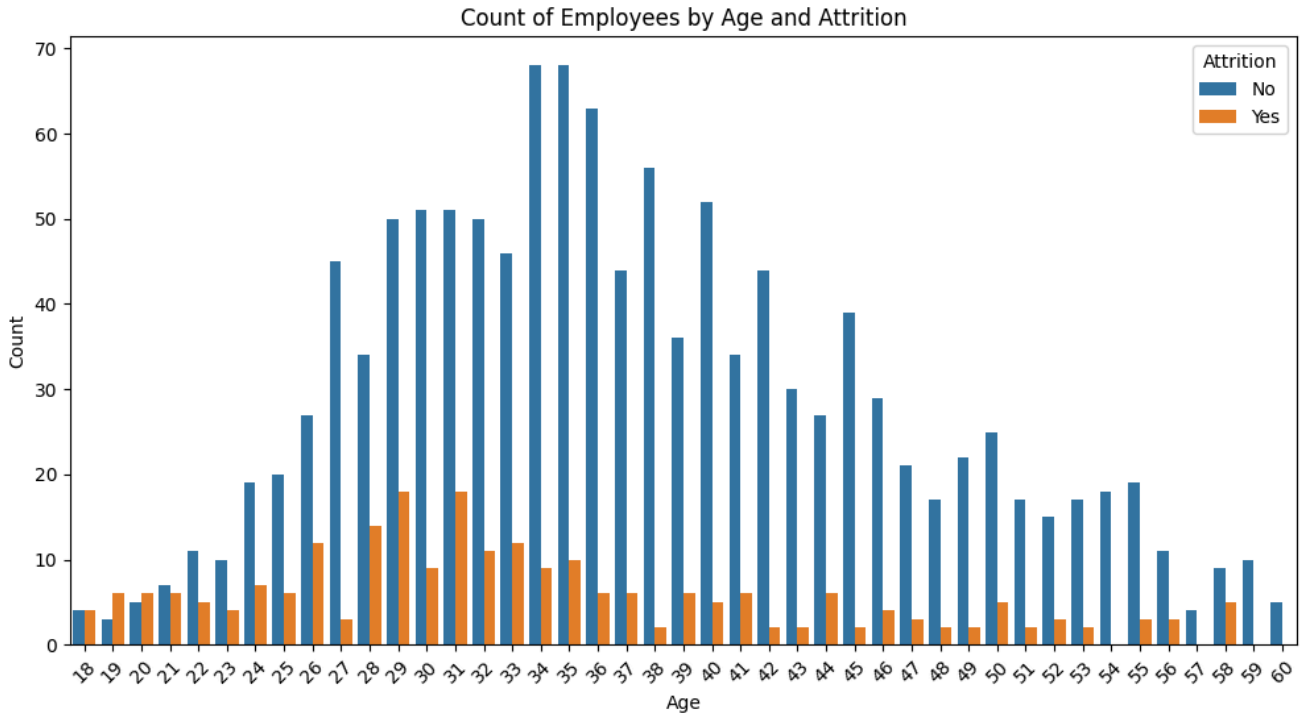
Here are the metrics we will be looking at for this business case:

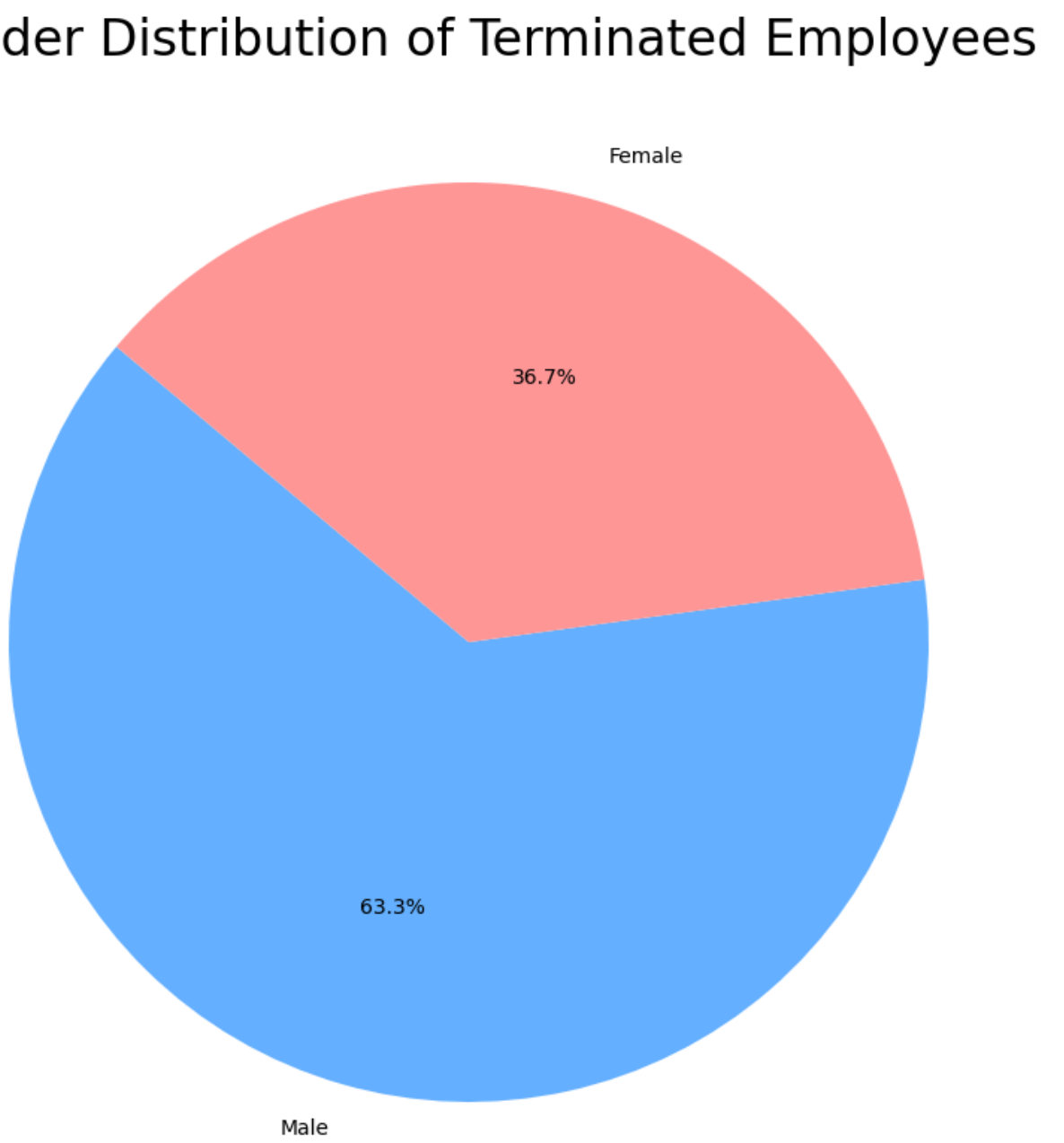
- Accuracy-Overall accuracy of our model
- Precision-Accuracy of positive predictions made by our model
    - High Precision indicates that an employee will leave, it is usually correct. **High Precision minimizes false positives.**
- Recall-The ability of our model to identify the actual positive observations
    - High Recall is going to be tough with this dataset specifically since we are battling a class imbalance issue. As we will see in our dataset, nearly 15% of our data is the positive **Attrition** class.
- F1-Score that factors in both Precision and Recall

EDA

HR Analytics data was collected here. This is a 228 kB, publicly available csv file with 1479 rows and 35 columns from Kaggle. Important features are going to be our target value '**Attrition**', and various features such as:

- Job Satisfaction
- Age
- Sex
- Job Title
- Salary
- Overtime


Count of Employees by Age and Attrition
Attrition
No
Yes
Count
Age
0
10
20
30
40
50
60
70
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60



Data Preperation

This dataset is already cleaned, but some processing still needs to occur. We will need to:

- Engineer Features
  - Combining, Transforming columns
  - Label Encoding, One-Hot Encoding categorical features
- Dropping Columns
- Balancing Classes
  - The class imblance issue will be a limitation of this dataset and a tradeoff between overall accuracy and Recall will happen

Modeling

We will end up building 20 models! Here is a quick synopsis of our methodology:

baseline model baseline model addressing class imbalance Baseline model addressing class imbalance with GridSearchCV performed to optimize hyperparameters model with unimportant features dropped model with unimportant features dropped addressing class imbalance model with unimportant features dropped addressing class imbalance with GridSearchCV performed to optimize hyperparameters

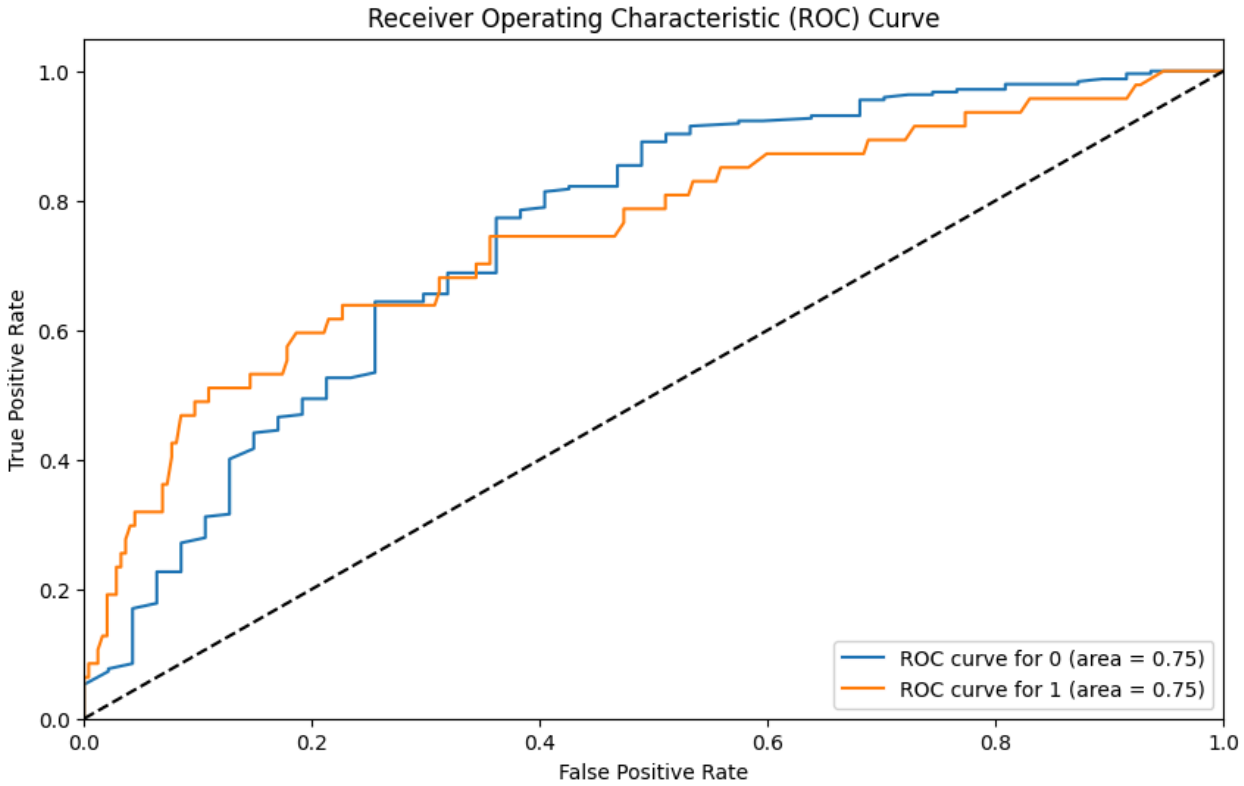
This methodology will be applied to all 3 types of models we are building:

Decision Tree Random Forest K-Nearest Neighbors

Then we will read in all results and pick the 4 best models. Those 4 best models will be used to create a Stacking Ensemble and then a Stacking Ensemble with GridSearchCV to finetune hyperparameters. Once our best model is created, we can discuss results.

Conclusion

- **Balancing our data raised the Recall and lowered the accuracy and precision** for our almost all of our models.
- When comparing respective base models (df, df_fi), retraining the model on data from **df_fi had a positive or nuetral impact on all scores except Precision.**
- Our **finetuned Voting model is our best model.** It has the highest F1-Score and ROC_AUC while also making the best trade-off for Accuracy to improve Recall and Precision.



Next Steps

Our **class imbalance issue is the biggest limitation to this analysis.** Our models and iterative approach yielded strong results. However, to truly optimize analysis here, we would need a more robust dataset. Gathering HR data can always be tricky, but if this was combined with US Census data and all features were standardized, this modeling approach could provide some great insights.

Other features could have improved the results or given more context. For example, if we had information on spousal income or state residence we could better understand the importance of these jobs to people. If someone lives in a more expensive state and contributes to the majority of their families income, they may be less inclined to leave a job in which they are unhappy, overworked, burned-out, etc.

Lastly, we don't know the reason these employees left their jobs. We still don't know if they were fired or let go. This would drastically change the context of the results as well.

Recommendations

The features with some of the biggest impact on our model are 'Overtime', 'Age', 'TotalWorkingYears', 'WorkLifeBalance', and 'MonthlyIncome'. Based on our EDA earlier, we know that the distribution of employee age is skewed to the right, meaning most employees are going be earlier in their career and some of them are going to be older executives. More holistically, the feature importances tell us that a shift in company culture is needed.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Jupyter Notebook** 100.0%