Madeleine Sharp

DSC550 – Term Project Write-Up

Introduction:

Employee attrition is a concern of employers, and likely has been a concern for as long as employers have existed to employ people. Amidst our current economic climate, spurred on by the COVID-19 pandemic, some might argue that employee attrition is of even greater concern than it was only a couple of years ago. This current economic climate has given rise to myriad scenarios, one of which has been dubbed "The Great Resignation." Because the current market is largely one in which employees have the upper-hand, and thus that power has shifted to the employees away from employers, employee attrition is likely to be a problem (not for employees, but for employers). Given this, not only is it useful for employers to have the ability to assess and predict employee attrition currently, but will be imperative for the future, also.

With respect to pitching to stakeholders for consideration of this issue, it would be imperative to highlight the points I have made above. In particular, focusing on the current economic climate and emphasizing important aspects of employee retention (less expensive and time-consuming to maintain current employees versus losing employees and hiring new ones in their places) would be of utmost importance. Stakeholders in this scenario are largely going to be employers, and likely individuals in the C-Suites or HR personnel/business partners. Cost and time savings are critical for companies and ensuring employee retention is pinnacle to that. Additionally, while this is not covered within the scope of my project, stakeholders that pay close attention to the wellbeing of employees (morale, compensation, resources, etc.) are likely to have less turnover anyway, so these conversations open-up a win-win potential for both parties.

For this specific project, my data was obtained from the dataset and source listed below. This dataset is an IBM employee attrition dataset (sans names to maintain privacy – the original dataset does not even have names included). Overall, this dataset includes a variety of data points about each employee (years at company, hours worked, marriage status, commute distance, etc.) as well as whether the employee experienced attrition or did not experience attrition. Unfortunately, the dataset does not give any indication as to the reason for an employee's attrition, and I think for any future considerations or analyses that would be an interesting additional variable to possess.

The overall goal of my project was to select a classification dataset that could be utilized to train a classification machine learning model.

Dataset Resource:

Kaggle. (2017). *IBM HR analytics employee attrition and performance*. Kaggle.

https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset

Summary of Project Milestones:

Exploratory Data Analysis (EDA) (Milestone 1):

In the first milestone of this project (Milestone 1), I conducted an EDA of my dataset and its variables encompassed within. This particular step largely involved assessing the variable snapshots, distributions, and relationships. To complete this step, the visuals I used included histograms, a correlation heatmap, and bar charts (vertical and horizontal). These visualizations are included below:

Figure 1: Above, set of EDA histograms for all variables in the employee attrition dataset.

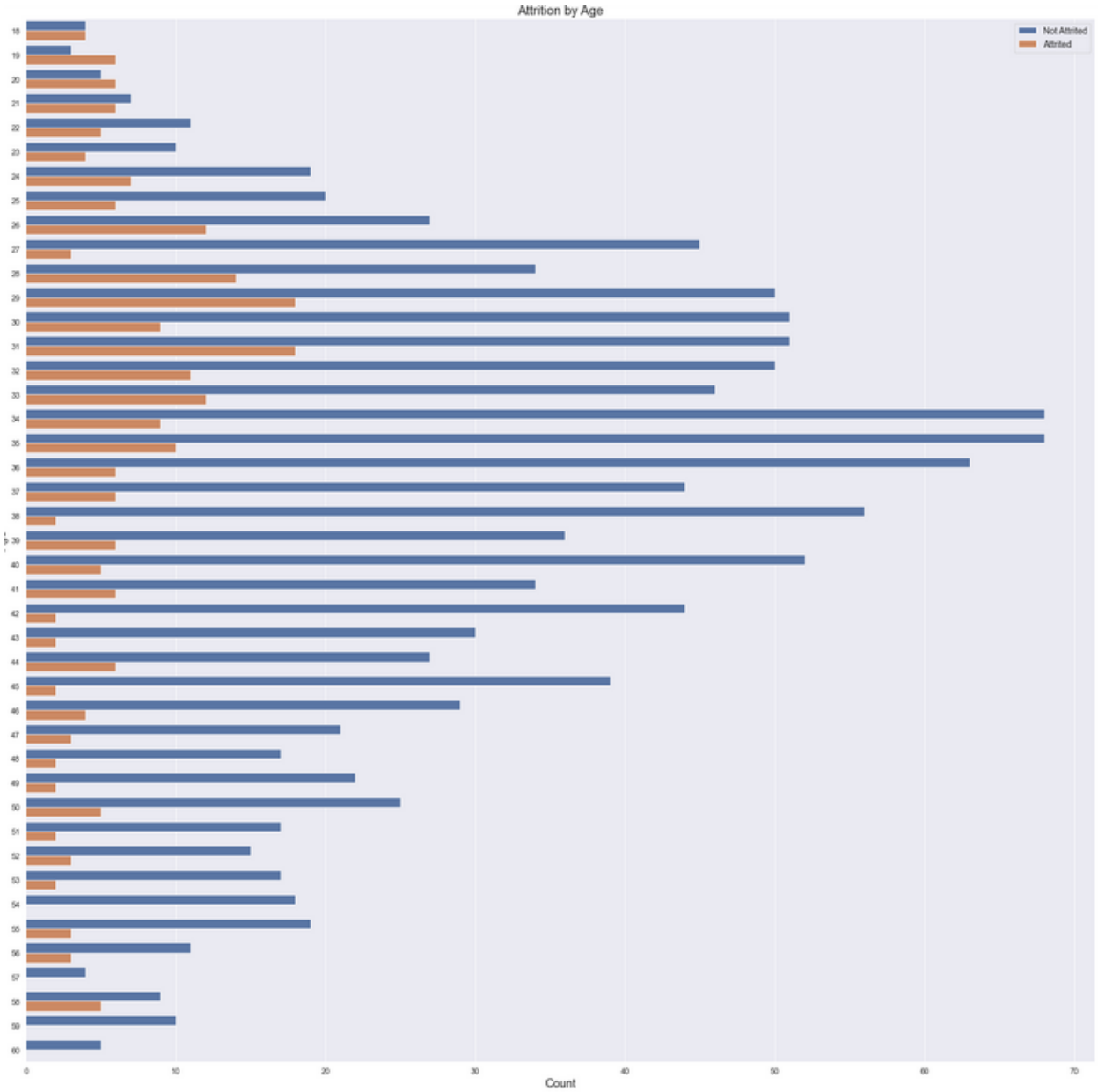

Figure 2: Above, a horizontal bar chart assessing attrition by age.

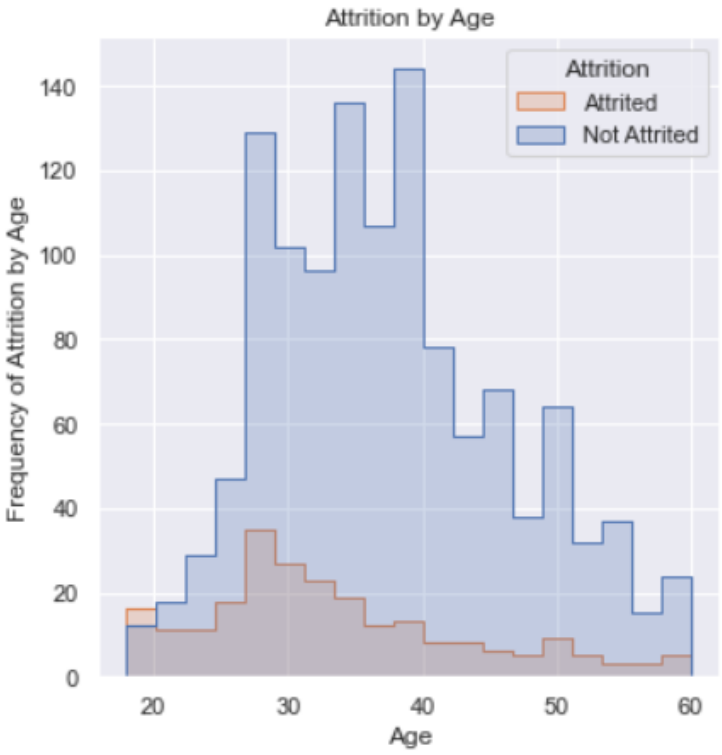

Figure 3: Above, a vertical bar chart assessing attrition by age.

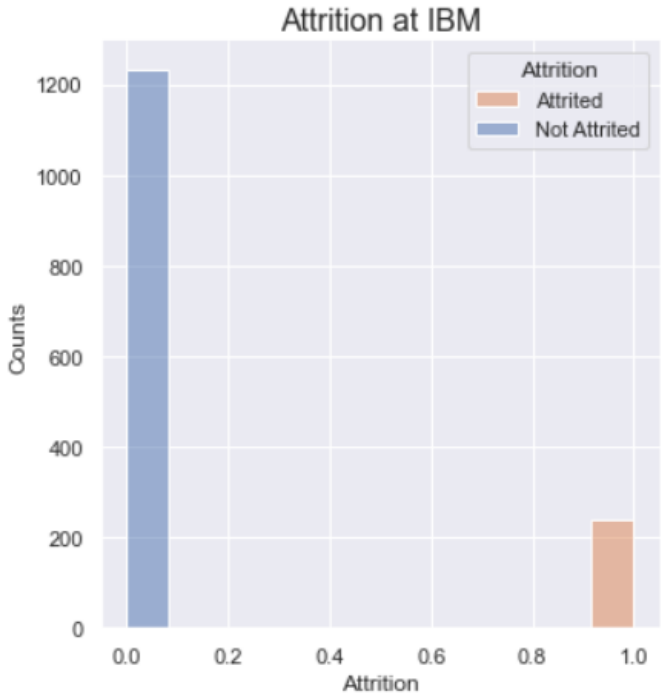

Figure 4: Above, a histogram of overall employee attrition.

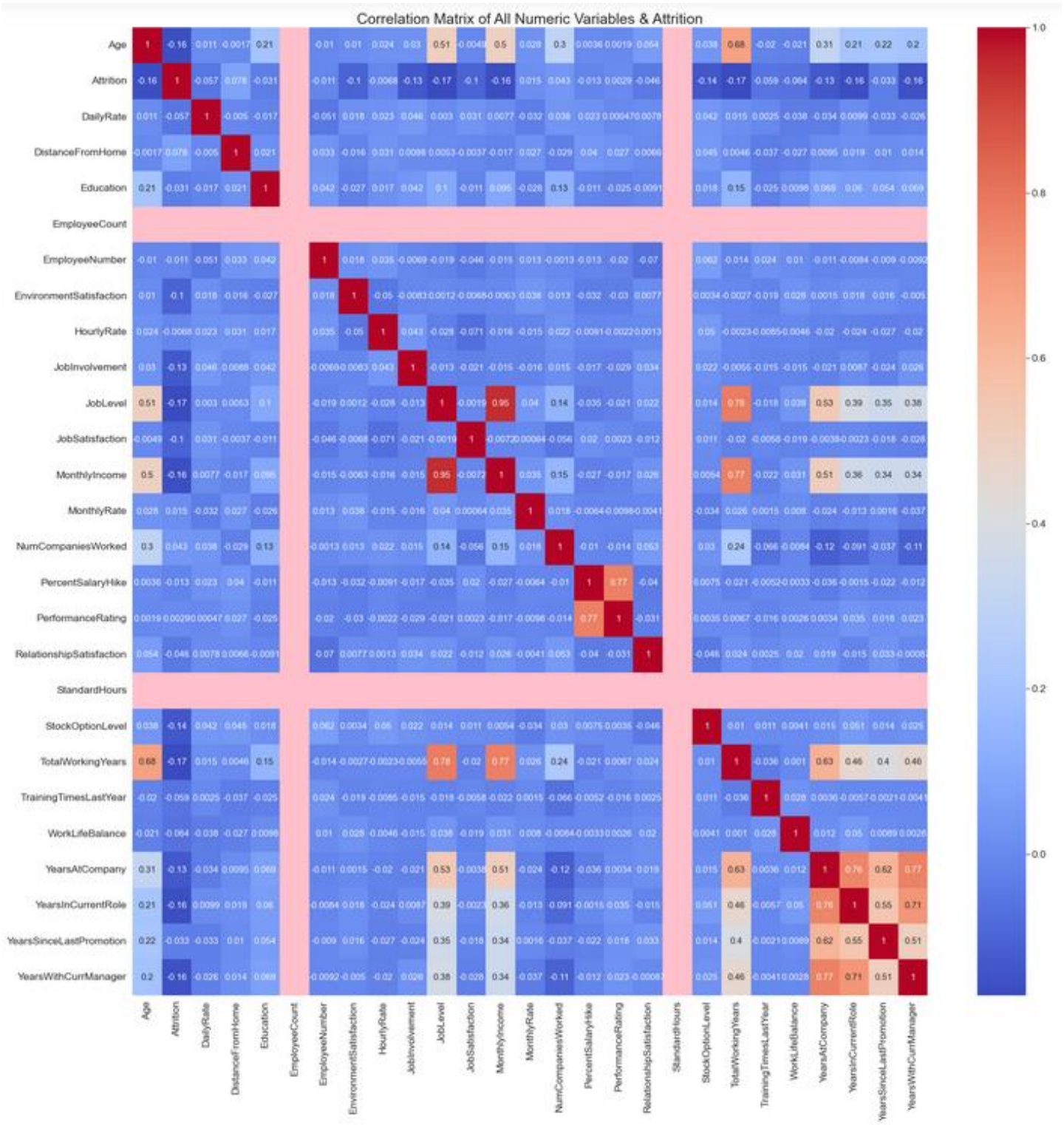

Figure 5: Above, a correlation heatmap of all variables within the dataset.

Data Preparation (Milestone 2):

Following EDA, my dataset required preparation, transformation, and cleaning prior to usage. In order to ensure that my data was ready to be used in a training and testing set for my model, I needed to complete some preparation steps including removing/dropping any unnecessary columns (those variables that I was not planning to use), replacing column headers with a more appropriate and standardized naming schema, assess for and handle any missing data, and ensure variable types were appropriate.

For my first step, I only needed to drop four variables that would not be of use to me. These variables were 'EmployeeCount', 'EmployeeNumber', 'Over18', 'StandardHours'. Both EmployeeCount and EmployeeNumber were ID-type variables that would not be of use to me. The Over18 variable was also not useful, as the dataset already had an Age variable that I was using for age (so Over18 would be redundant). StandardHours were all the same – there was no difference or variation, and each data point/entry for this variable was the exact same value – so this likely would not have any impact on the outcome of my classification/target variable.

Secondly, I amended all of the dataset names to this naming schema: name_name. Doing so would make this easier to work within my code, and it is my preferred naming schema when working with datasets in Python.

For my third step, I assessed whether any missing data existed within this dataset. Interestingly (and luckily!) enough, no missing data was present, so I was not required to remove or handle any missing values.

For steps four and five, in my initial milestone iterations (specifically Milestone 2), I extracted out the target variable from the rest of my dataset and ensured that I encoded any categorical variables needed. This step was then moved from Milestone 2 to Milestone 3 in preparation for the modeling step – it made the most sense to move it here and handle it in the splitting and training step of this project.

Model Building and Evaluation (Milestone 3):

In the final milestone step, it was time for me to train and assess my model. The steps for this specific milestone included splitting the data into training and testing sets, ensuring that the variables were encoded properly for use, fitting the model to the training set, running the model, and assessing/evaluating the model.

In the first step of this milestone, I went ahead and assigned my x and y values – y being my target variable (attrition) and x being the remaining variables (features) in the dataset. Next, I went ahead and split the data into training and testing sets. I took this a step further and split my columns into numerical and categorical training and testing sets, converted the categorical ones to encoded values using OneHotEncoder, fit and transformed to the encoder, and then concatenated the numerical and categorical features back together.

After completing those steps, I was ready to move forward to the model building step. For my project, I wanted to use a variety of hyperparameters and model types in an endeavor to

see which parameters were best for my classification problem. The four various estimators I used were:

- Random Forest
- Logistic Regression
- Decision Tree
- KNN Classification

The reasoning for choosing these is that my dataset included data that I sought to classify, and these models are often used in that specific instance. Alongside these, I initialized hyperparameters for each dictionary that was associated with one of the above estimators. Next, I went ahead and created my pipeline and grouped parameters, then used a grid search model to be trained and fit. Lastly, I went ahead and evaluated the model by assessing the best parameters and displaying a classification report as well as a confusion matrix.

At the end of this milestone, I was able to determine the following:

- The model accuracy was 86.17%, with the best parameters from the grid search model being the below:
    - RandomForestClassifier(max_features=2)
    - classifier__max_features: 2
    - classifier__n_estimators: 100
- Precision: 86% of the predictions were correct for the 0 class (Not Attrited) and 100% for the 1 class (Attrited).
- Recall: The percentage of positive cases caught/identified were 100% and 10%, respectively.
- F1: The percentage of positive predictions that were actually correct is 92% and 19%, respectively.
- Support: The number of actual occurences of the classes within the xtest dataset are 373 and 68, respectively.

While this was an initial “good look” at how my modeling methodology of choice worked with my dataset, I had come to find that my model was having issues classifying the minority class (attrition). Given this, I went a step further and employed SMOTE (Synthetic Minority Oversampling Technique) methodology as an additional step to balance the classes and so my model would not have as much of an issue predicting the attrited (minority) class. SMOTE is a statistical technique for increasing the number of cases in a dataset in a balanced way - works by generating new instances from existing minority cases that are supplied as input.

From the SMOTE adjustment, I was able to determine that by applying this method both the overall model accuracy and recall improved, coming in at 94% and 90%, respectively.

Conclusion:

Overall, this project was insightful. The analysis of the variables in this project indicated that employing a SMOTE methodology to be used with my model was a better outcome than not doing so.

From a model building standpoint, the largest takeaway was the issue of the imbalanced classes within the dataset. In my initial model, the model accuracy was acceptable and was not necessarily weak, however, it was having a difficult time predicting the minority class of the dataset (attrition). This is merely due to the fact that my dataset was imbalanced – the dataset possessed significantly more data for the "not attrited" class than it did for the "attrited" class. In order to combat/counteract this, I needed to employ an additional methodology that would take into account this drawback (SMOTE).

To be frank, I am not certain whether this model is ready to be deployed. I would argue that the model post-SMOTE methodology is more ready than the original model that I deployed, but the accuracy is still not as high as I would like. Additionally, the dataset is a particular case study dataset on a sample of IBM's employee population, thus I do not know if the dataset size is robust enough for this model to be employed (no pun intended) – let alone outside of IBM itself.

Given this, my recommendations for moving forward would be to try some additional model analyses, and to do so with a variety of employee attrition datasets – not just this single IBM case. I think the largest challenge would be finding that kind of data and using all of this data to build a model that is more applicable and appropriate in a variety of employee attrition scenarios. However, it might prove difficult to do so and to have a "generic" employee attrition model – especially given the varying differences between employers. Lastly, I believe that finding data that includes reasons for attrition would be an interesting next step – knowing the "why" would be insightful to employers who are seeking to retain employees.

I thoroughly enjoyed this project – it granted me a healthy amount of experience with machine learning techniques, and I look forward to the ways in which I will use the skills and experiences I have garnered from this course in the future.