**Data Science Final Project Report**

Allama Ikbal Sijan

Montclair State University

PSYC 546

Dr. Michael Bixter

May 6, 2023

**Introduction**

**Goal of the Data Science Project**

The goal of this data science project is to find out which machine learning (ML) algorithms best predict job satisfaction. According to international business machine (IBM), “Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy” (“What is Machine Learning,” 2023). In this project, I will be experimenting with various ML algorithms (e.g., logistic regression, KNN, and decision trees) to find out which algorithm does the best job at predicting job satisfaction. In addition, I will also perform hyperparameter tuning for certain models in order to enhance their accuracy.

**Overview of the Data Set**

The original data set is available at IBM Watson Analytics in IBM community. After discovering that the current version of a dataset was pre-cleaned before being made available, I conducted further research to locate a previous version that was still in its raw form. My investigation led me to a version of the dataset that was uploaded to Kaggle by David O’Keeffe (dgoke1 on Kaggle.com). According to O'Keeffe, it is an earlier version of the original dataset with the column “Employee Source” randomly added (O’Keeffe, 2018). The data set titled “IBM HR Data.csv” consisted of 37 variables/columns and 23532 observations/rows.

**Exploratory Data Analysis**

**Loading Packages**

I had to load a number of R packages for this data science project. In my final R script, I decided to put them all together in the beginning of the script. These are the packages that I used for this project: “tidyverse”, “psych”, “caret”, “caTools”, “ggcoorrplot”, “reshape2”, “randomForest”, “MASS”, “glmnet” and “purr”. I used options(scipen = 999) to prevent the automatic conversion of small numbers to scientific notation in output and plots.

**Descriptive Statistics**

Firstly, I loaded the data set to my R environment, and then I used the function summary to get descriptive statistics of the whole data set. The descriptive statistics revealed a lot of meaningful information about the data set that was helpful in the preprocessing stage. I discovered the presence of both numerical and categorical variables. Using the summary function, I obtained the minimum, median, and maximum values, as well as the missing values, for all the numerical variables. Moreover, I identified some of the numerical variables to be ordinal in nature (e.g., Job Satisfaction, Work life balance, job involvement, etc.). I also discovered the specific binary variables (Attrition, Age, Gender) that needed to be converted to 0, 1. I didn’t get much information about the categorical variables that are not in the Likert scale, so I conducted individual descriptive statistics for those variables. Among the 23,532 employees in the dataset, 96 were terminated, 3,709 resigned voluntarily, and 19,714 were still employed. Analyzing the gender distribution, the dataset comprised a larger number of male samples (14,120) compared to female samples (9,400). Furthermore, examining the variable "Over Time" revealed that the majority of employees did not work overtime. There were 2 observations with a “Y” response instead of “Yes.” I tackle that issue in the data cleaning and transformation section. Exploring the nominal variables, I observed that the majority of employees were employed in the "Research and Development" department, while the most common educational background was in "Life Sciences." Although the "job Role" variable displayed a wide range of options, the highest number of individuals in the sample held the position of “Sales Executive”. Additionally, the sample consisted of more married individuals compared to those who were single or divorced.

**Data Visualization**

I created some basic visualizations (histograms) for relevant numeric variables to understand the distribution and any trends. Here are the summary findings: Age displayed a normal distribution, while Daily Rate, Hourly Rate, and Monthly Rate exhibited uniform distributions with minimal variations. Distance from home was positively skewed, indicating that a greater proportion of individuals lived closer to their workplace. Monthly Income also showed positive skewness, with a median income of $4936. The number of companies worked was positively skewed, with the majority of individuals having worked at only one company. Other variables such as Percent Salary Hike, Total Working Years, Years at the Company, Years in Current Role, Years Since Last Promotion, Years With Current Manager (bimodal distribution), and Training Times Last Year were also positively skewed. On the other hand, Standard Hours displayed a block distribution, indicating no variation at all.

**Preprocessing**

**Data Cleaning & Transformations**

I conducted a count of the missing values in the dataset, which revealed a total of 367 missing values across various variables. After removing these missing values, the remaining number of observations in the dataset was 23,296. It's worth noting that this count considers the possibility of some observations having multiple missing values. Secondly, I checked the total number of duplicate observations in the data set. I found 14 duplicate observations, which I removed from my data set. As “Employee Number” is unique to every employee, I decided to check if there are any duplicate “Employee Number” in the data set. This was particularly crucial as some employees may have multiple observations at different times, making them less likely to be identified as overall duplicates (where the entire row is duplicated). Consequently, my examination revealed a total of 49 instances of duplicate "Employee Number" in the dataset. After deleting those duplicate values, my data set had a total of 23,233 rows/observations. Thirdly, I removed the unnecessary variables (“Employee Count,” “Standard Hours,” “Over 18”, “Application ID,” “Employee Number,” & “Employee Source” since it was randomly generated) based on my exploratory data analysis (EDA). Fourthly, I conducted an assessment for potential outliers in the dataset. Exploratory data analysis highlighted "Monthly Income" as a variable with a wide range ($18,990), where the median stood at $4,930. However, the presence of large values (potential outliers) skewed the mean to $6,506. To address this, I removed the outliers, resulting in a mean of $5,505, which was much closer to the median. Additionally, I eliminated a singular and non-meaningful observation categorized as "Test" in the "Education Field" variable. Following the removal of outliers, my final sample size comprised 21,429 observations. After eliminating unnecessary variables from the data set, the total number of variables was 31.

Next, I proceeded with data transformation to facilitate the implementation of machine learning models. Firstly, I converted the binary variables, namely "Attrition," "Gender," and "OverTime," into numerical values of 0 and 1. For the "Attrition" variable, which initially had three levels, I merged the "Termination" level with the overall attrition category due to its relatively small number of values (86 after data cleaning). Consequently, the transformed "Attrition" variable now consisted of two levels: "Yes" (n = 3,745) and "No" (n = 19,488). Then, I converted the appropriate nominal/categorical variables to factors and ordered the ordinal variables. Then I split the data set into two subsets which are training and test set. The training set contains 70% of the data, where the test set contained the remaining 30% of the data. Before splitting the data set, I set seed in the R environment so that I get the same results across different runs.

**Feature Selection**

To select the most useful features for the ML models, I first created a correlation matrix using Pearson’s correlation coefficient for all the numeric variables. To calculate correlation, I first created a data frame with only the numeric predictors (continuous data). I created a correlation plot which is attached in the Appendix section (See Appendix A). Then I ordered the correlation values in a descending order to find the predictors with highest correlations. The correlation matrix shows that “Years With Current Manger” has a strong positive correlation (0.77) with “Years At company”; “Total Working Years” has a strong positive correlation (0.76) with “Monthly Income”; “Years In Current Role” has a strong positive correlation with “Years At Company” (0.75) and “Years With Current Manager” (0.70). Based on these findings, I decided to drop the following predictors from my feature list “Years At Company” and “Years in Current Role”. This is because highly correlated features may lead to overfitting the data. The remaining predictors (that I didn’t drop) seemed more relevant to me for predicting “Job Satisfaction”.

Next, I created another correlation matrix where I included ordinal predictors along with numeric predictors to check if they are highly correlated to each other. The correlation plot is attached in the Appendix section (See Appendix B). I used Spearman’s rho to find correlations between ordinal and continuous predictors. The correlation matrix shows that “Job level” had a strong positive correlation with “Monthly Income” (0.91) and “Total Working Years” (0.71). So, I decided to drop “Monthly Income” from my feature list. Next, I used the function nearZeroVar to identify any predictors/features that have near zero variance. Any predictors with near zero variance will not be useful in our model. The result indicated that none of the variables had near-zero variances.

Lastly, I used LASSO Regression model to reduce the magnitude of certain coefficient values and select the best predictors/features for predicting job satisfaction. First, I created separate y and x variables for running LASSO model smoothly. I performed the function cv.glmnet to find the optimal lambda parameters using k-fold cross-validation. After performing the LASSO model analysis, I created a list of the most important predictors. I used a cutoff score of 0.005 to exclude predictors that had a coefficient closer to zero (meaning they were less important to predicting job satisfaction). The final list of selected features that contributed the most to predicting job satisfaction included 12 predictors, namely “Attrition,” “Business Travel,” “Department,” “Gender,” “Job Involvement,” “Number Companies Worked,” “Over Time,” “Performance Rating,” “Relationship Satisfaction,” “Stock Option Level,” “Training Times last year,” and “Work-Life Balance.” This list of predictors didn’t include any of the predictors that I dropped using the correlation matrix.

**Machine Learning Models**

**Logistic Regression**

Initially, I utilized a logistic regression model encompassing all the predictors. The purpose of this base model was to serve as a benchmark for comparison with a subsequent model that would utilize only selected features. In this model, I treated the dependent variable, Job Satisfaction, as a factor. Upon evaluating the model's performance on the test set, it yielded an accuracy of 0.33. Afterwards, I created another logistic regression model using only the selected variables from the feature selection process. This model resulted in similar accuracy (0.33) on the test set. Since the selected variables didn't contribute to improving the accuracy, I decided to experiment with cross-validation techniques. I first tried repeated cross-validation, but it didn't make any difference in the accuracy. I also attempted to use Leave-One-Out cross-validation, but unfortunately, my RStudio crashed after running for a while.

**K-nearest neighbors (KNN)**

On my first try, I used all the features in a KNN model with k = 5 to predict job satisfaction, treating it as a categorical variable. The resulting accuracy on the test set was 0.87, which is significantly higher than the logistic regression model. In my next attempt, I employed KNN with selected features and again set k = 5. However, this time the accuracy dropped to 0.70, which was considerably lower than when using all the features. This led me to realize that adding more features might enhance the performance of the KNN model. I added the following features in my next attempt to improve the performance of the model – “Age”, “Years with Current Manager”, “Total Working Years”, “Monthly Income”, “Marital Status”, “Job Role”, “Job Level”, and “Environment Satisfaction”. To select additional features, I relied on a correlation matrix conducted during the preprocessing phase, as well as my intuition, by experimenting with different combinations of features. This approach led to a remarkable increase in accuracy on the test set, reaching 0.94.

In order to assess whether cross-validation could enhance the performance of my KNN model, I implemented the repeatedcv method, as it appeared to be a better option than cv. For cross-validation, I utilized a 10-fold approach that was repeated 5 times in my model. During the cross-validation process, the model explored various values of k to identify the optimal configuration with the highest k value. The analysis concluded that the best-performing k value was k = 5, resulting in an accuracy of 0.91 on the cross-validated sets. Next, I used this KNN model to predict job satisfaction in the test set. This resulted in an accuracy of 0.94, similar to the non-cross-validated KNN model.

To assess the impact of different k-values on the KNN model's performance, I implemented a function that ran the model with k-values ranging from 1 to 20. However, I was aware that choosing smaller k-values increases the risk of overfitting the model. To analyze the results, I created a line graph plotting the cross-validation accuracy and test accuracy. It revealed an intersection point between k = 5 and 6 (See Appendix C), indicating that these values were the most optimal for my KNN model. This finding also supported the selection of k=5 in the KNN model with cross-validation. Despite this, I experimented with k=3. Unsurprisingly, the accuracy on the test set was 0.97, suggesting a high level of accuracy. However, it's crucial to note that this model might have indeed overfitted the data due to the low k-value. Thus, this model might not be generalizable to a different data set even though it performs well on the test set.

**Random Forest**

Finally, I decided to apply the random forest algorithm to the training data, utilizing the selected variables, in hopes of achieving improved model performance. The resulting accuracy of the random forest model on the test set was astonishingly high at 0.99. To validate these findings further, I performed 10-fold cross-validation on the random forest model. The outcome was remarkably consistent, with an accuracy of 0.99.

**Results & Conclusion**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** (only using selected features) | **Sensitivity/Recall** | **Specificity** | **Precision** | **Overall Accuracy** | **F1** | **Running Time** |
| Logistic Regression | 0.23 | 0.76 | 0.27 | 0.33 | 0.21 | Fast |
| Logistic Regression with repeated cv | 0.23 | 0.76 | 0.27 | 0.33 | 0.21 | Fast |
| KNN (k = 5) | 0.69 | 0.90 | 0.68 | 0.70 | 0.69 | Fast |
| KNN with added selected features (k = 5) | 0.94 | 0.98 | 0.94 | 0.94 | 0.94 | Fast |
| KNN (k = 5) with repeated cv | 0.94 | 0.98 | 0.94 | 0.94 | 0.94 | Slow |
| Random Forest | 0.99 | 1.00 | 0.99 | 0.99 | 0.99 | Moderate |
| Random Forest with cv | 0.99 | 1.00 | 0.99 | 0.99 | 0.99 | Slow |

*\*Most of the evaluation metrics in this table are average across four levels of job satisfaction. See Appendix D for metrics information about individual job satisfaction levels\**

Based on the evaluation metrics, it's clear that the random forest model outperformed all other machine learning models with an astonishing accuracy of 0.99 and an F1 score of 0.99. On the other hand, the logistic regression model performed the worst, showing an F1 score of 0.21. When I examined the confusion matrix closely, I noticed that the logistic regression model struggled to predict Job Satisfaction level 2. The logistic regression model predicted 0 values for Job Satisfaction level 2, but I am not sure of the underlying reason behind such predictions. The KNN model showed notable improvement after I added more relevant variables based on feature selection. It's accuracy and F1 score increased to 0.94. Interestingly, cross-validation didn't change these metrics. Despite KNN achieving high accuracy and F1 score, the evaluation metrics indicate that the random forest model outperformed all other models in predicting job satisfaction, making it the top-performing model among the models I tested.

However, it's essential to note that the results obtained from the random forest model might be difficult to generalize to other datasets (e.g., different industries, demographics). Although I tested the model's performance on a separate test set, it still belonged to the same dataset with similar patterns. Therefore, future researchers and data scientists should investigate the generalizability of random forest on entirely different datasets. They should also explore other machine learning models like Support Vector Machines and conduct further diagnostics to ensure the models' underlying assumptions are met. Future researchers should also consider experimenting with ensemble learning techniques such as Bagging and Boosting, which combine the predictions of multiple individual models to improve overall performance. In conclusion, despite the limitations of the models, it is evident that advanced machine learning models outperform regression in predicting job satisfaction. Therefore, the I/O psychology field can benefit immensely from introducing advanced data science topics and modeling techniques to its curriculum.

**References**

Irizarry, R. A. (2019). Introduction to data science: Data analysis and prediction algorithms with

R. CRC Press.

*What is machine learning?* (n.d.). IBM. Retrieved May 6, 2023, from

<https://www.ibm.com/topics/machine-learning/>

**Appendix A**

A picture containing text, screenshot, diagram, plot

Description automatically generated

**Appendix B**

**A picture containing text, screenshot, diagram, plot

Description automatically generated**

**Appendix C**

**A picture containing text, diagram, line, plot

Description automatically generated**

**Appendix D**

A screenshot of a spreadsheet

Description automatically generated with medium confidence