HR Analytics Capstone Project Document

Problem statement

- The main opportunity is to identify Data Scientists (target) ready to leave their current jobs.
- This opportunity is important because identifying the right target saves time and resources as opposed to the HR consultant calling one by one each candidate on the database. Target candidates ready to leave their current job would have their resumes done and more willing to go through the interview process since they have made up their minds to explore new opportunities. Non-targets are still undecided, which means the process could drag longer, requiring more time/resources/costs on the HR consultant.
- The project was featured in Kaggle but the results were different from this project.

Industry/ domain

- This project belongs to the Human Resource domain.
- The market demand for Data Scientists and its related field has grown exponentially in Singapore and globally. Filling those positions has turned into a lucrative proposition. Companies are increasingly prioritising Data Science in their roadmap for hiring.
- Thus, this project will be highly relevant to any companies looking to hire or set up a data analytics or data science team.

Stakeholders

- The stakeholder could be someone directly involved in the head-hunting business specifically a HR professional with some background in data science looking to leverage on the market demand for Data Scientists/Data Engineers/Data Analysts/Business Analysts/Research Analysts/Trainers/Machine Learning Engineers/Economics Analysts.
- Using the database, the head-hunter could be interested to leverage on identifying the right target to fill up vacant positions of their clients' companies. In the realm of head-hunting, speed and the right fit matters. "Get there before someone else does".

Business question

- The business value of this project is to find a pool of ready Data Scientists to fill up vacant positions in the market.
- For each Data Science role filled, the head-hunter earns a commission worth 1-1.5 month's salary of that job. The more ready contacts the head-hunter has, the sooner she/he can supply the market with the right candidate - which translates to more earnings for the HR agency.
- Those who are ready to leave but are identified as not quitting means losing opportunities to grab the right person to fill up vacancies. The project should mitigate the false negatives, make it as as low as possible. Secondly, as accurately and as MANY as possible, identify those who want to quit. In other words, extracting the highest possible TRUE positives.

Data question

- To fulfil the business question and the opportunity for the stakeholder, the dataset should be reliable and large enough to extract as many 'target' classified as 1 as possible.
- The dataset should have the relevant candidates with relevant background e.g. trained or have experience in data science – in terms of breadth (diverse industry/commercial experience/exposure) and scope (at minimum, intermediate level to advanced/proficient level of technical knowledge and skills).

The Data

- Given the limited availability of workable datasets in this domain, the dataset was retrieved from Kaggle.
- The original dataset had over 21,287 rows with 14 sets of columns.
- The dataset had numerous null values which makes it unsuitable unless some work is done to clean it up.
- The dataset has both numeric and categorical information about candidates' background including: work experience, relevance of experience, training hours, city development index, education background, previous company type and length of last job.

Data science process

Data analysis

- The process of checking and deciding what to do with the null values took up a substantial amount of time.

- Firstly, the large count of null values for certain categories posed a threat to the reliability of the dataset. There were more than 6000 null values for gender, more than 2000 null values for target, more than 3000 null values for education level, and other null values across company types and company sizes. To fix these issues, the null values in the target column were removed. By fixing this, the dataset had 19,158 rows which is still substantial. The rest of the null values were less dire and easily fixed using forward/backward fill function.
- Secondly, the dataset is highly imbalanced in almost every category including the target where only 25% were class 1 (ready to quit).
- To visualise the findings for better analysis, we used pie charts and bar graphs were employed. The graphs display the imbalance in the following areas:

1. Gender where male accounts for 90% of data scientists.
2. STEM as a major in education background.

- Not all features are equal, some were dropped and those which could be useful (relevant in the outcome) were further mapped into numeric rankings to make way for machine learning classifiers.
- The dataset had to be standardised/ scaled.
- To fix the imbalanced dataset, the SMOTE (oversampling) was deployed.
- The NearMiss (undersampling) was also used.
- As a base model, Logistic Regression algorithm was used to test the dataset before/ after SMOTE and NearMiss.

Modelling

- 10 features out of 14 were used for machine learning.
- The ExtraTree classifier was utilised to rank the feature importance. We identified the top 4 features:
    1. Training hours (number of hours spent training shows the greatest influence in whether a candidate stays or quits)
    2. City Development Index (the city of their current job plays a part in deciding factor whether they stay or quit)
    3. Work experience (to a minor extent, one of the deciding factors).
    4. Company size of their current job meaning prospects for upward mobility within the company could be a factor to quit or stay.
- The dataset requires classification to determine the business and data outcome, so the following algorithms were used:
    1. Random Forest (RandomForestClassifier)
    2. KNN (KNeighborsClassifier)
    3. Decision Tree (DecisionTreeClassifier)
    4. MLP (MLP Classifier
    5. Ada Boost (AdaBoostClassifier)
    6. Gradient Boosting (GradientBoostingClassifier)
    7. XGBoost (XGBClassifier)
    8. Stacking (StackingClassifier)

Pre-processing

- For optimisation, GridSearchCV was used to find the best parameters for each model. Since the datasets are imbalanced, SMOTE was used on the dataset without splitting it into training and tests sets. GridSearchCV function will split the dataset so I had to make sure the datasets will not re-split twice.
- The SMOTE training sets were used to fit each machine learning model.
- Stacking Classifier took the longest to run. Originally, the Support Vector Classifier took the longest but this model was scrapped. Not only was the SVC inefficient in use of time to run/fit the model, the score was also unimpressive.
- Besides the confusion matrix and classification metrics, specificity and sensitivity were added to mix to find the optimal model for this project.

Validation of modelling results

- To validate the models and their accuracies in terms of classification properties, the full classification scores were used:
    1. Confusion Matrix
    2. Accuracy, Precision, Recall, F1 Scores
    3. Specificity and sensitivity scores
    4. Auc Roc scores and plotting of their curves

Performance

- The findings from running the 7 algorithm or machine learning models:
    1. Accuracy scores won't necessarily mean much.
    2. Given the highly imbalanced dataset, the specificity score (for class 0) fared higher than the sensitivity score (for class 1).
    3. The specificity score in this case was high in each algorithm as the size for class 0 (not quitting) is larger (75%).
    4. The score of interest would be the sensitivity score where the findings should yield as many quitters as accurately as possible. But the sensitivity scores for each algorithm were below average.
    5. To optimise the sensitivity score, threshold function was initiated and used to determine if this can improve results.

Implementation

- In terms of their performance, these 3 models were selected to run the threshold function:
    1. K-Neighbors Classifier (KNN)
    2. Random Forest Classifier
    3. Extra Gradient Boost Classifier (XGBoost)

- From the threshold function, the decision is opened to the stakeholder to determine the right threshold for the 3 selected algorithms or any machine learning models for the project.

Data outcome

- With the minimal threshold of 20% we can take any selected Classifier models to run the function to produce a list of those who will be quitting.
- For greater accuracy/reliability, the list can be further refined by increasing the threshold to 25%, 30%, 35%, 40%, 45%, 50%.

Response to stakeholders

- The recommendation to the stakeholders is that it is the management's decision to select the thresholds to identify the candidates most likely to quit.
- The higher the threshold used, the more accurate is the list, but this also means the list gets narrower. The lower the threshold = the lower the accuracy = the larger the list.

Challenges

- Deciding which null values to delete was a major hurdle in the EDA process. Initially, we took out all null values and ended up with 8800+ rows. When discovered this compromised the accuracy scores, and went back to reassess the data cleaning. We decided to start with the 'target' column and delete those null values since these would be too precarious to predict. Subsequently, deleting the null values for 'target' meant working with 19158 rows, a stark improvement from just 8800+ rows.
- The scores for the models were not optimal when the GridSearchCV parameters were run on unbalanced datasets.
- After much investigation, this was corrected where the GridSearchCV used SMOTE resampled sets.
- It wasn't until we fit the StackingClassifier that we discovered Stacking doesn't work on split datasets. We had to run SMOTE without splitting the datasets for both StackingClassifier and GridSearchCV. The error back and forth took up considerable time but was worth it because there were improvements to the classification scores.