

School of Informatics and IT

Diploma in Business Intelligence & Analytics

# Major Project Final Report

Project Title:

Unlocking Potential: A Data-Driven Approach to Job Security in Northeast Singapore

Submitted By:

<Ho Wei Bin> <2202561C>

Supervisor:

Makino Michiko Senior Manager (Data) | GCD Office (Pasir Ris-Punggol GRC)

# Student’s Declaration of Originality

I hereby declare that the course work(s) submitted by me, is a result of my own effort. I affirm that there is NO plagiarism and copying, either partially or entirely, from someone else’s design and works, without giving proper credit and acknowledgement to the source(s)/author(s).

I am aware that I shall be subjected to disciplinary actions deemed appropriate by the School of Informatics & IT and Temasek Polytechnic if I am found to have committed or abetted the offence of plagiarism in relation to this submitted work.

|  |  |
| --- | --- |
|  |  |
| Ho wei bin  2202561C  15/7/2024 |  |

# Table of Contents

[Acknowledgements 3](#_Toc44597276)

[Abstract 5](#_Toc44597277)

[Chapter 1: Introduction 6](#_Toc44597278)

[Background of the project 6](#_Toc44597279)

[Problems Identified 6](#_Toc44597280)

[Objectives of the project 6](#_Toc44597281)

[Scope of the project 6](#_Toc44597282)

[Chapter 2 Data Understanding & preparation 6](#_Toc44597283)

[Data cleaning 8](#_Toc44597279)

[Data transformation](#_Toc44597279) 10

[Preliminary data visualization/correlation analysis](#_Toc44597279) 11

[Chapter 3 Modelling 17](#_Toc44597284)

[Evaluation metric selection 17](#_Toc44597279)

[Linear regression & regressor result 19](#_Toc44597279)

[Model Results (Classifier) 20](#_Toc44597279)

[Logistic regression 23](#_Toc44597279)

[Other classifier models 25](#_Toc44597279)

[Conclusion on modelling & champion model selection 29](#_Toc44597279)

[Chapter 4 Visualization & business intelligence 30](#_Toc44597284)

[Business intelligence 38](#_Toc44597279)

[Chapter 5 Additional configuration 40](#_Toc44597284)

[Chapter 6 Model Deployment 30](#_Toc44597284)

[Chapter 7 Conclusion 44](#_Toc44597285)

[Lessons learnt and Insights gained 44](#_Toc44597286)

Recommendations for Improvement 44

# Acknowledgements

‘I would like to thank Miss Michiko Makino for assisting me in securing the necessary dataset from the organization and giving me feedback along the journey of this project.’

‘I would like to thank Mr. Kok Yau Lim for offering me guidance and reminders for the administrative matters during the process of this project.’

‘I would like to thank Mr. Zaw Htet Wai for giving me suggestions and advise on small academic matters in the making of this project.’

‘I would like to thank Mr Jivaka Tei Ren Yi for his assistance when it came to troubleshooting the web development process during web development’.

‘I would like to thank Ms Teo Yun Fong for assisting me in providing me insights and suggestions when it came to the design for front end related matters in this project such as the website, dashboard and pitching slides.’

# Abstract

Objective: The objective of this project is to enhance Project SUCCESS's ability to make data driven decisions involving job placements for residents within certain regions in Singapore by implementing certain strategies and methods. Firstly, the project will introduce methods on automation technique in the data loading and cleaning process, which will help to increase efficiency for future data users. Secondly, a user-friendly and interactive dashboard will be created. This dashboard will include a predictive model to visualize and analyse data, allowing users to filter, drill down, and explore recruitment trends in detail. Finally, machine learning will be leveraged to identify patterns within the historical data. These insights can be used to inform future recruitment strategies and improve overall placement success for Project SUCCESS.

Scope:

* Data understanding
* Data wrangling & preparation (i.e., cleaning & transforming)
* Primary data visualization (Correlation analysis for modelling)
* Metrics selection & model consideration
* Modelling
* Model Deployment & integration
* Chart creation & dashboard assembly
* Insights drawing & actionable suggestions forming.
* Additional configuration to power BI workspace & dashboard.

Extent the objectives were reached:

Machine learning application: Thorough investigation on regressors and classifiers given the dataset’s nature with clear reporting on findings and speculations. Heavy relations were drawn between the model and the business context.

Business intelligence: Findings and insights drawn were converted into actionable insights for stakeholders to implement and consider as strategies to achieve the overall goal of the project.

Key findings:

* Base data structure is messy and difficult to manipulatable.
* Base target column is severely unbalance (ML application)
* Classifier is much more optimal as compared to regressors.
* Default logistic regression is a superior model in this context & dataset.
* Recommendation (More recommendations mentioned later)
* Targeted actions like outreach, recruitments and events will be extremely beneficial in the aim for higher successful placements numbers.
* Constant revisions and analysis needed to develop more concrete and efficient methods to boost awareness and participation.
* Conclusion: Data visualisation coupled with appropriate ML application can heavily assist in awareness of the situation and give new perspective for decision making

# Chapter 1: Introduction

Started in 2003, Project SINGAPORE UNITED AS A COMMUNITY TO CARE AND TO ENCOURAGE SELF-SUFFICIENCY (S.U.C.C.E.S.S) is a non-profit organisation supported by Northeast CDC that perform jobs matching to candidates. The offices are in various CC particularly in Pasir Ris and Punggol.

## Background of the project

In hopes of truly understanding and gaining utmost insights into the preference of local resident in both Pasir Ris and Punggol, a project was put in place to make a detailed analysis on the employment situation by utilizing the records that Project SUCCESS had. The dataset is created from records taken during individual interviews over the course year 2023.

## Problems Identified

* Limited data visibility into the different stages of the recruitment funnel
* Manually analysing large datasets to identify trends and patterns can be time-consuming and inefficient. Hidden points and insights hindering more concrete and efficient course of action.
* Repetitive tasks like data cleaning might be hindering productivity.

## Objectives of the project

To improve efficiency and gain deeper insights from my data by I'm looking to automate the data loading and cleaning process. In addition to develop a user-friendly dashboard that incorporates a predictive model for data visualization and analysis.

## Scope of the project

* Leverage machine learning models to identify trends and patterns within the recruitment data & prepare for future deployment should it be necessary.
* Develop an interactive dashboard that visualizes key metrics and insights derived from the machine learning analysis. This dashboard will allow users to filter, drill down, and explore the data in various ways.
* Automate workflows & additional BI features by using power query, alert system, power automate and other software or functions within power BI desktop/ power BI platform to enhance security, efficiency and aid the users’ convenience.

# Chapter 2: Data understanding & preparation

# Data Understanding

Data background: The data itself is listed by the companies that have applied for employee seeking at the organization and will therefore contain repeated companies since some of them do apply for employee seeking numerous times throughout the year. The original dataset contains 20 columns and 195 rows. The individual rows of data are instances where a job has been listed from Project success. These jobs are offered by different companies. This means that some of the companies can offer more than one position more than once.

Data dictionary:

|  |  |  |
| --- | --- | --- |
| Company | Data type | Name of company |
| Industry | Object | Industry that the company belongs to |
| Date | Date | Date of interview |
| Vacancy quantity | Int64 | Number of vacancies the company has for that listing |
| Interviewed | Int64 | Number of people interviewed |
| KIV | Int64 | Number of people dismissed but kept in view |
| Rejected | Int64 | Number of people rejected |
| Selected | Int64 | Number of people accepted |
| Rank & file | Object | whether the companies interviewed have employees in non-managerial positions |
| PMET | Object | whether the companies interviewed have Professional, Managerial, Executive, or Technical (PMET) positions, |
| Region | Object | The CC in which the interview was taking place. |
| Method | Object | The way that the interviewed was approached |
| Successful Placement | Int64 | Not to be confused with selected. Successful placement indicates the number of people that were **allocated into the position that they interviewed for and did not reject the opportunity after successful interviews.**  \*There are instances where the recorded number of selected interviewees were 0 but the number of Successful Placement is 1\* |
| Pasir ris east | int64 | Number of interviewees from pasir ris east |
| Pasir ris central | int64 | Number of interviewees from pasir ris central |
| Pasir ris west | int64 | Number of interviewees from pasir ris west |
| Punggol coast | int64 | Number of interviewees from Punggol Coast |
| Punggol shore | int64 | Number of interviewees from Punggol shore |
| Punggol west | int64 | Number of interviewees from Punggol west |
| Sengkang & others | int64 | Number of interviewees from Sengkang & others |

Data cleaning challenges:

* **Spreadsheets with Merged Headers:** Headers spanned across multiple rows, making it difficult to interpret the data.
* Inconsistent data formatting
* "✔" used instead of "Yes" or "No" for PMET and Rank & File.
* Unlabelled cells due to missing or inapplicable values.
* **Separate Interviewee Breakdown Sheet:** Data containing the number of interviewees from the 7 possible divisions were in another excel sheet.
* **Combined Interview Outcomes:** Numbers for Interviewed, Selected, Rejected, and KIV were combined with reasons in a single cell (e.g., a cell could be: "3 Interviewed, 2 x KIV, 1 x Rejected").

Formatting and primary wrangling:

* **Manual Sheet Creation:** A new, clean sheet was created by merging information from both original sheets. The merging method was very simply manually copy pasting the data from the original file into this new sheet and organized.
* Data Imputation: Missing "Selected" values were imputed as 0.
* Data Standardization: "✔" annotations were converted to "Y" and "N".
* Data Separation: Combined interview outcomes were separated into individual columns ("Interviewed," "Selected," "Rejected," "KIV").

Reason for manual data imputing rather than python manipulation:

Given the relatively small dataset size (195 rows), a manual copy-paste techniques was the most efficient for initial data wrangling. This strategy proved particularly effective for tasks like merging information from separate sheets and separating combined interview outcomes. Additionally, the complexity of the code required to split the combined interview outcome variable ("Interviewed, Selected, Rejected, KIV") within a single cell outweighed the benefits of automation via coding for this specific case.

Target evaluation & selection: While picking the target variable, there were a few options. I have considered picking either Interviewed or Selected as the column. However, there was a problem which is that the 4 columns, Interviewed, KIV, Selected and Rejected were too closely related. They essentially depend on one another. In fear of potential target leakage, I decided to go for **Successful Placement**. The bigger reason for this is because the Successful Placement column will be a more meaningful as a sign to show the preference in the different industries.

EDA process & data transformation considered:

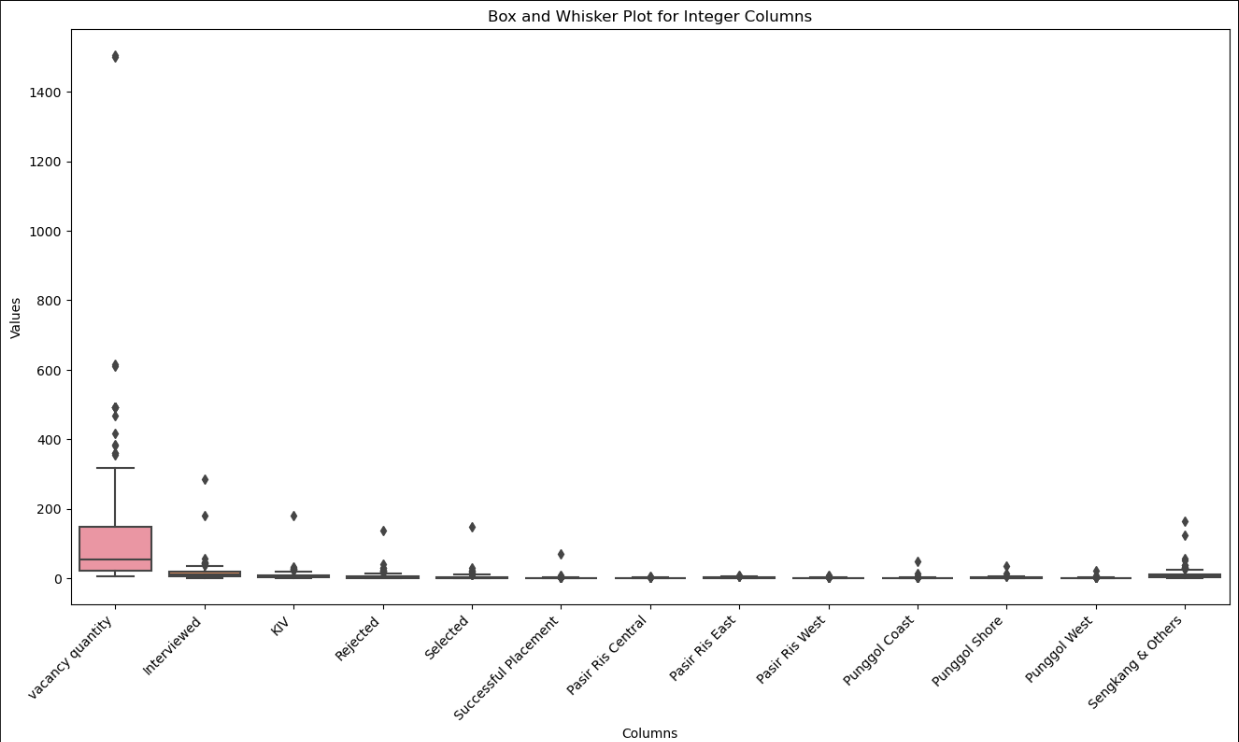
Checking for data cleanliness:

* Missing values: Missing values will be treated as 0 since it is all numeric and the data as explained by the provider is cleaned and should not be tampered with unnecessarily that might change the definition of the data itself.
* Inconsistencies: Due to most of our data being in numeric, I normally ensured that the numbers are in constant decimal places. As for few nominal data columns, I ensured that their capitalization is constant.
* Duplicates: Duplicates has also been checked for and came back with 0.
* Invalid/Incorrect data: After vetting through during the manual process of data wrangling as previously mentioned, I double checked that the data was appropriate and legitimate.

\***Outliers\*:** While potentially affecting model performance, outliers are acknowledged as valuable insights and will not be removed.

The benefit of this way of handling any dirty data is that it reduces complexity by verifying data quality before any other process and by retaining outliers, valuable and important insights are taken into consideration even if they might affect predictive model’s performance. The reasoning for not imputing missing values as the mean or median value is because the data provider confirmed that the dataset underwent primary analysis, reducing the likelihood of missing values due to unintended data omission. Since missing values were likely intended to represent zero values within the data itself. Imputing with a different statistic (mean/median) could potentially misrepresent the true meaning of these missing entries.

Visualizing outliers:



Data transformation (Additional columns):

[Punggol] & [Pasir Ris]:

* Method: These two columns are created summing the 3 individual parts values within each respective area. (Pasir Ris = Pasir Ris East + Pasir Ris West + Pasir Ris Central. Punggol = Punggol Coast + Punggol Shore + Punggol West)
* Reasoning: By combining the each of the outlets into a single measurement, it firstly reduces dimensionality. This will reduce the number of features for the model. This can be particularly beneficial for models like decision trees, as it can lead to a simpler and potentially more interpretable model structure. Aside from the possible ML improvements, presenting consolidated data provides a broader perspective on performance trends within each area; Punggol and Pasir Ris, offering valuable insights for stakeholders.

[Day] & [Month]

* Method: I extracted month and day information from the existing year-based date format. Discarded the year information for this analysis.
* Reasoning: Since the data Since data wasn't collected consistently throughout the year (not daily or monthly), using the year might not provide meaningful insights into trends or seasonality but more importantly, this data only contained one year of data. In addition, extracting month and day allows me to analyse **relative** changes within the dataset, focusing on patterns within the collection timeframe. For example, you might see if there are more interviews on weekdays compared to weekends.

Rejected Transformations:

* Selection Ratio: due to the presence of listings with zero interviewees, leading to undefined ratios. Alternative options (flag column, filtering) were considered but rejected to maintain data integrity. Considering our potential target column for machine learning, this has a high risk of being target leakage.
* Acceptance Rate: the same reason as the Selection Ratio. Listings with zero selected candidates would create undefined values.

Target variable classification:

* **Model Exploration:** I will be exploring and experimenting with both classification and regressor models. For regression model using the original "Successful Placement" value is planned. For classification, I have considered separating it into different bins. But given that the target variable's range (0-71), binning into low, medium, and high was initially considered but rejected due to severe class imbalance. Many rows (117) have a value of 0, leading to models biased towards the majority class. To address this class imbalance, the target variable was transformed into a binary classification. Values other than 0 were converted to 1, resulting in a "Yes" (Successful Placement > 0) vs. "No" (Successful Placement = 0) classification. This approach reduces the imbalance to a 1:1.5 ratio (No vs. Yes) compared to the original data.

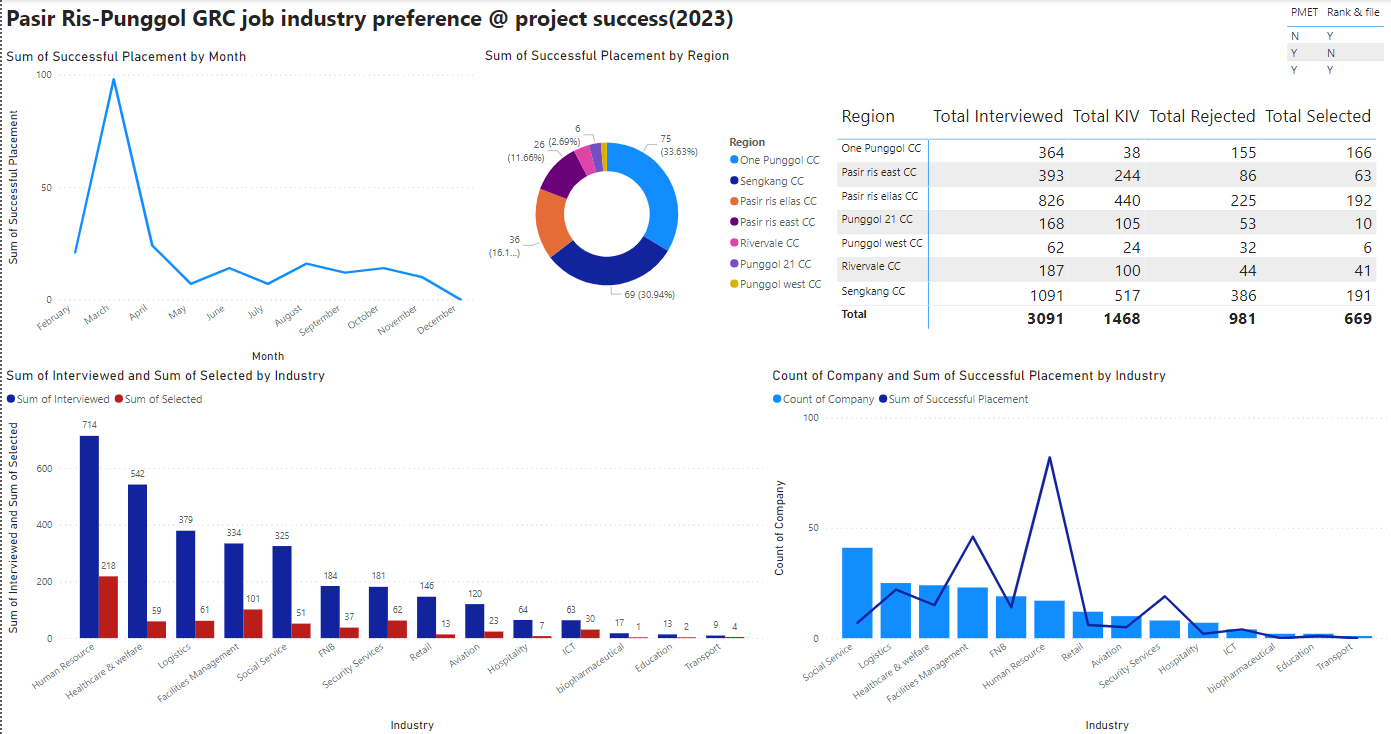
Scaling & normalization:

While scaling & normalization are common and quite reliable preprocessing technique to encourage an interpretable but still reliable model, I ultimately rejected them for this specific case.

Reason:

* With a relatively small dataset (195 rows) and binary classification, the impact of scaling/normalization on model performance might be small.
* Since most of the features represent count, the scale holds an inherent meaning. By scaling or normalizing this type of data, it could be distorted and ultimately make the model less straightforward leading to cases of underfitting.

Preliminary data visualization:



Analytics insights:

* Interviewed vs selected:
* Observation: According to the clustered column chart, there seems to be a positive but weak a weak correlation between the number of people interviewed and the number ultimately selected by companies.
* Implication: This suggests that a high number of interviews might not directly translate to a high number of successful placements. Other factors might be influencing company selection decisions.
* Listing vs Placements:
* Observation: According to the line & bar chart as shown, there is no direct relationship between the number of companies listed per industry and the number of successful placements within that industry.
* Implications: This shows that simply having a high number of companies in that industry does not guarantee successful placements. Other factors, such as industry-specific skills required or company reputation, might be playing a role.
* March Placement Spike:
* Observations: There is an unusually high number of successful placements occurred in March.
* Potential causes: Further investigation shows a specific employment event held at One Punggol CC, where a division within Certis achieved 71 successful placements. Being a positive outlier.

Target evaluation and selection:

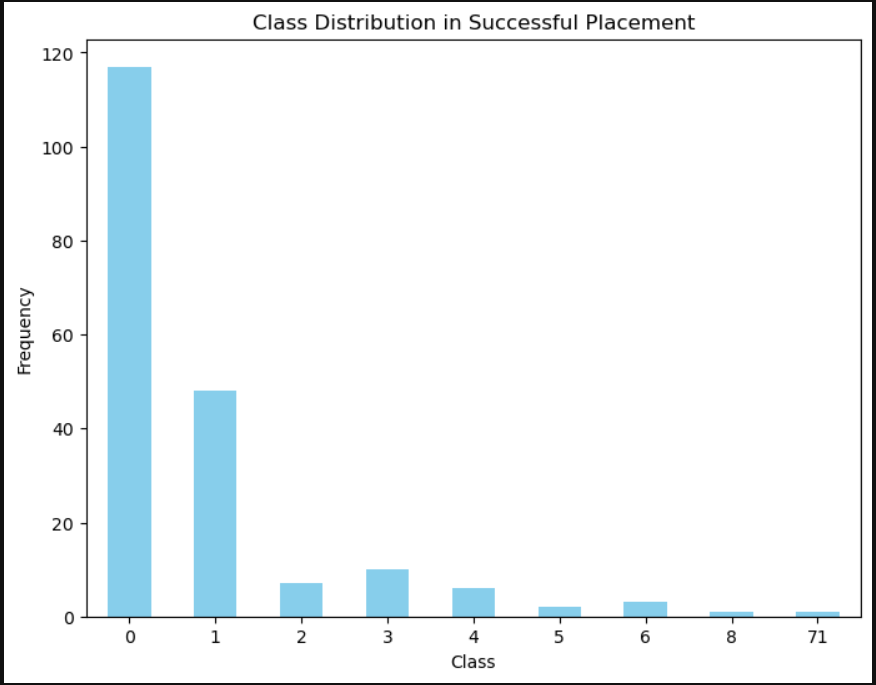
While picking the target variable, I have considered picking either Interviewed or Selected as the column. However, there was a problem which is that the 4 columns, Interviewed, KIV, Selected and Rejected were too closely related. They essentially depend on one another. In fear of potential target leakage, I decided to go for Successful Placement. The bigger reason for this is because the Successful Placement column will be a more meaningful as a sign to show the preference in the different industries.

Selected vs Successful Placement addressing:

In combination with a heat map that I plotted which will be discuss later, I have also plotted a scatter plot to show that statistically the column “Selected”, and “Successful Placement” are not dependent on one another. By plotting a scatter plot to get a clearer picture, it is fair to say that while it is not strictly a directly proportional relation, the selected column and successfully placed column is generally quite proportionally related. But to reiterate, while the Successfully placed column is closely related to the Selected column, they are not a direct inference of one another. Even without statistics, fundamentally, the two columns are not derivative of one another. To reiterate, the selected column represents the number of people who have gone through the stages of interview and have been selected by the company. It does not mean that the position is definite. Sometimes due to undisclosed negotiation and discussion, the interviewee might be allocated to another position in which case, it will not be marked as successfully placed. There is also another case where it a selected interviewee does not mean successful placement. That is when the interviewee withdraws their application last minute after they have already been marked as selected. There are cases when the opposite happens i.e., even if a company position has no selected personnel, there can be people still successfully placed. This is because they have been accepted after they have been kept in view (KIV) and potentially due to more discussion on the company’s end, they have been confirmed and successfully placed.

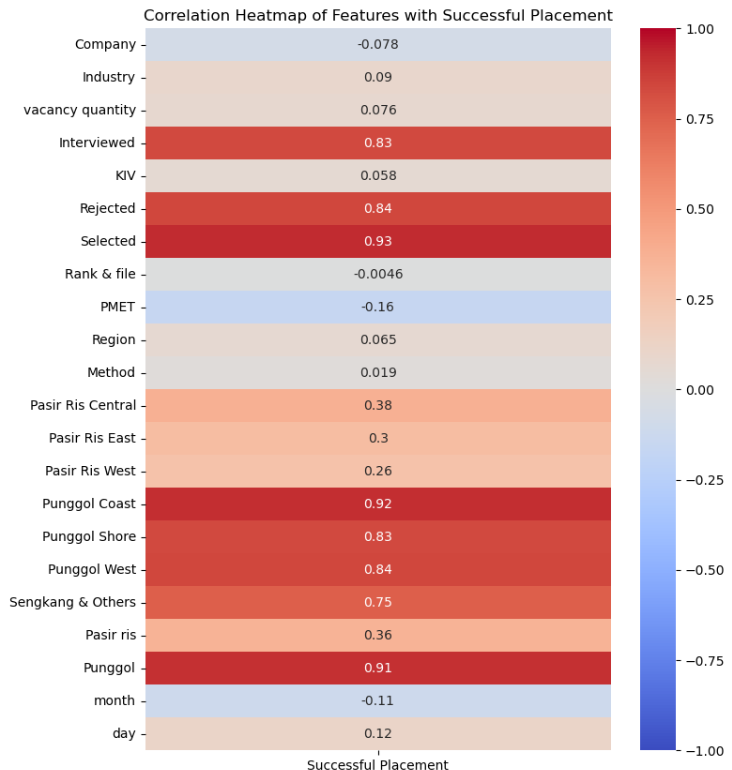
Target leakage addressing: While target leakage, where the target variable influences other features, is the main concern in ML application, it is not an issue here. My target, "Successful Placement," is the final outcome that I am predicting, and the other interview stage counts (rejected, selected, KIV and Interviewed) appear to be recorded independently. My target is the final outcome of my prediction, it is not being derived from or influenced by the interview stage counts. However, to further ensure model integrity, I have maintained consistent data collection practices and using techniques like separate training and testing sets.

[Bar graph to illustrate class imbalance]



Class imbalance detection: as mentioned in the transformation section above, there is a huge class imbalance particularly in the value of 0 in Successful Placement. Even if I choose to do a multi class classification involving maybe 3 classes, it would be impossible to balance out the severe class imbalance especially if I choose to add the value of 0 and 1 together.

[Heat map for correlation analysis]



Correlation analysis: Due to numerous models being heavily influence by the correlation between one variable and another, this chart will provide more insights on which features are potentially more correlated to our target.

Insights:

* Contrary to my initial dashboard insight, the heatmap shows a positive correlation between "Selected" and "Successful Placement." This suggests a generally proportional relationship, though not necessarily a direct one to one correspondence.
* Among the highly correlated variables, KIV (kept in view) has a correlation value below 0.8. This might indicate a more complex relationship between KIV and successful placements compared to the other variables.

Checking correlation of transformed columns:

* In terms of the regional differences, Punggol region has a significantly higher correlations (> 0.8) with successful placements compared to the Pasir Ris region. Which shows that there is a different dynamic in these areas.
* The heatmap indicates a weak correlation between "Day" and "Month" with successful placements. These factors might not be significantly influencing placement outcomes in this dataset.

Multi collinearity analysis:



Multicollinearity & its significance:

* Negative Impact on the model’s performance
* High multicollinearity would cause the model to struggle to differentiate the individual impact of each correlated feature on the target variable.
* High multicollinearity could also inflate the variances of estimated coefficients, leading to misleadingly low p-values and potentially unreliable conclusions about feature importance.
* High multicollinearity would also make it difficult to understand how each feature influences the target.

Addressing multicollinearity in my context:

* The inf in some of the column is the abbreviation for infinity found in my regional data i.e., the different parts of Punggol and Pasir Ris and even the transformed column. Because of that, I have decided to exclude individual region columns from further analysis. Instead, the model will consider the consolidated "Punggol," "Pasir Ris," and "Sengkang and Others" columns.
* Reason: While the 4 other measurements method have high multicollinearity due to a simple reason. These features represent counts at different stages of the interview process. A high number of interviewees would naturally translate to a higher number of both selected and rejected candidates. Similarly, a high number of interviewees could also lead to a higher number of KIV candidates.
* Justification: The reason for still considering them in the model is because **a lot of machine learning algorithms are robust to multicollinearity to a certain extent.** They can learn the underlying patterns in the data even if features are correlated. This allows the model to potentially capture complex interactions between the different outcomes that might not be readily apparent with simpler feature selection techniques.

# Chapter 3: Modelling

Evaluation Metrics:

Regressor: [Mean absolute error & mean squared error]

* MSE:
* Reason: Our target column is a count variable making it always a positive integer. This is distinct from continuous variables that can take any real number value within a specific range.

Optimality:

* MSE is optimal in this scenario because MSE focuses on the squared difference between predicted and actual placements. This emphasizes larger errors as they contribute more to the overall error score.
* Numerous machine learning algorithm in particular linear regression are trained using loss functions based on squared errors. Evaluating with MSE directly matches with the model's training objective, allowing for a more reliable assessment of its performance.
* MSE also has an added positive byproduct. MSE normalizes the errors by squaring them, making the metric less sensitive to the scale of the target variable (number of placements). This allows for easier comparison between models even if the target variable is measured in different units.
* MAE:
* Reason: Our target column is a count variable making it always a positive integer. This is distinct from continuous variables that can take any real number value within a specific range.

Optimality:

* MAE focuses on the absolute difference between predicted and actual placements, directly addressing the size of the errors, giving a concrete image of the situation.
* MAE is also beneficial in the sense that it gives equal weight to all absolute deviations. This makes it less sensitive to extreme outliers that can significantly inflate MSE.
* MAE also helps with interpretability. Since the units of MAE are the same as the units of my target variable, it allows for a straightforward interpretation of the error. For example, an MAE of 2 indicates an average prediction error of 2 placements.

Benefits of combining:

* Complimentary skills: MSE emphasizes large errors which is important for identifying significant deviations from predicted placements and MAE is Less sensitive to outliers, providing a robust picture of the overall error distributions
* Covering flaws
* MSE’s sensitivity to large errors can be overly influenced by outliers. MAE mitigates this by focusing on absolute differences, giving equal weight to all deviations.
* MAE does not differentiate between large and small errors. MSE addresses this by penalizing larger errors more heavily, ensuring they don't get overlooked.

Together, MSE and MAE provide a more comprehensive view of my model's performance. I can see how well it handles both average errors and large deviations from predicted placements.

Evaluation metrics (Classifier)

Main: \*\*Weighted F1 score & AUC ROC\*\*

Supporting: [F1, Recall, Precision, Accuracy]

* Accuracy: provides the overall proportion of correct predictions, making it easy to understand and clear.
* Precision: shows the proportion of true positives from the total positives predicted which will reduce false positives.
* Recall: shows the of actual positives that are correctly identified which will reduce false negatives.
* F1 score: provides a balance average of precision & recall.
* AUC ROC: focuses on ranking ability, measuring how well the model distinguishes between classes.

Additional consideration: Accuracy can be misleading in situations with class imbalance which is why I considered some alternatives.

Reasoning:

* Both false positives and negatives can be detrimental:
* False positives waste time and resources on unsuitable candidates when companies devote manpower and time to handle admin matters and training towards unsatisfied manpower.
* False negatives miss out on potentially successful candidates making it a shame and

Solution: F1 score which provides a harmonic mean of precision and recall, giving equal weight to both.

Weighted F1 score: Through some research, I realized that there are different versions of F1 scores; weighted and macro. But ultimately, I choose to use weighted F1 score. This is because it takes class distribution into account and gives more weight to the minority class (successful placements) you care about most.

Reason for weighted F1 not macro F1:

Mild Class Imbalance: The difference between weighted and macro F1 is insignificant in my case.

Focus on Minority Class: However, I am primarily interested in identifying successful placements which is the minority class. Weighted F1 prioritizes the minority class by assigning it higher weight, providing a more relevant measure for my specific needs.

Threshold setting: While the specific purpose and predicted results of the model might not be the sole focus, achieving a strong **weighted F1 score** remains crucial for our evaluation. A threshold of **0.75** is targeted as it shows a **robust performance level** that balances precision and recall effectively. This score will mean that a model that can reliably identify residents interested in working for specific companies, without raising concerns about potential overfitting or memorization of past trends. This balance ensures the model's predictions are **meaningful and trustworthy** for the stakeholders.

Model Consideration + discussion [Regressor]:

To begin I will be setting a baseline linear regression using the original version of the target column instead of the transform binary column. The base linear regression consists of all the features that exist in the dataset with no hyperparameter or modification. The result that I have retrieve is that the MSE has a value of ≈ 1.67 and a MAE of ≈ 0.85. Well, it is not a terrible start with the model incorrectly predicting about 2 person per company, but it is also important to remember that the target is counting number of people with majority of the column being 0 and 1 which means there might be times where the prediction can be off. For example, a company could have successfully placed 0 person and the model could predict 1 or 2. After a bit of consideration, I realize that there is a limitation to my two metrics of MSE and MAE which is that these metrics are less informative for skewed data. Large errors for the few higher values can significantly inflate them, even if the model predicts most 0s correctly.

Model result:

[**Regressors]**

|  |  |  |
| --- | --- | --- |
| Model name | Mean squared error | Mean absolute error |
| Linear regression(base) | 1.6704589942495465 | 0.8531358231708333 |
| Linear regression (No inf VIF features) | 1.5563587872594091 | 0.9044952984006022 |
| Ridge linear regression | 1.5558275219479512 | 0.918485493930495 |
| Lasso linear regression | 1.7203739404208078 | 1.033652830375971 |
| Elastic net linear regression | 1.7203739404208078 | 1.033652830375971 |

Validation & evaluation(regressors): Since regressors are not my main priority, I will only reflect some of the data nature in a regressor as well as some conclusive evaluation. The relatively high MSE values across all models suggest there might be some inherent noise or unexplained variance in the data. This is because even the best performing models have a moderate MSE.

Additionally, the difference in performance between models is subtle, indicating that the features might not have a strong linear relationship with the target variable. This suggests that the data may have some underlying complexity that a simple linear model might not capture fully. Based on the results, the baseline linear regression or the model with VIF filtering might be suitable starting points for further exploration.

Their performance suggests a potential linear relationship between features and the target variable, but there's room for improvement, especially considering the high MSE values. Regularization techniques like Ridge might be helpful, but Lasso and Elastic Net seem less effective in this case, potentially due to the specific characteristics of the data.

<End of regressors>

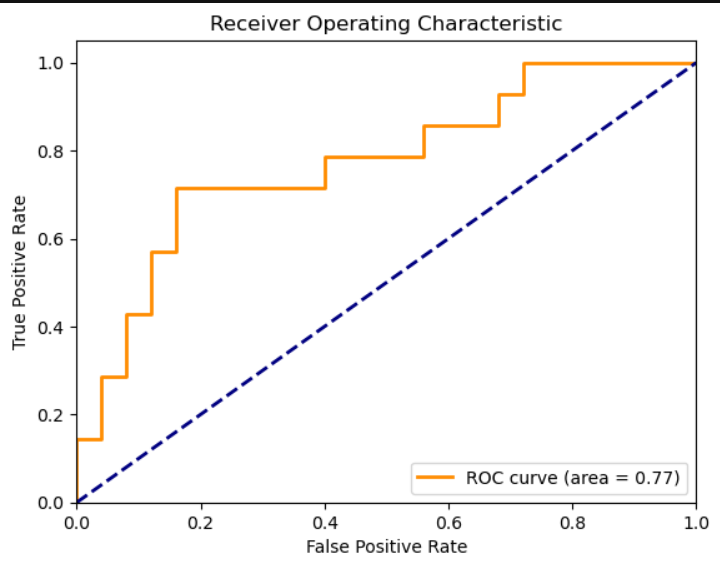
[Classifiers results]

[**Best Classifiers]**

|  |  |  |  |
| --- | --- | --- | --- |
| Default logistic regression | Precision | Recall | F1-score |
| 0 | TR:0.74 TE:0.79 | TR:0.87 TE:0.88 | TR:0.80 TE:0.83 |
| 1 | TR:0.75 TE:0.73 | TR:0.56 TE:0.57 | TR:0.64 TE:0.64 |

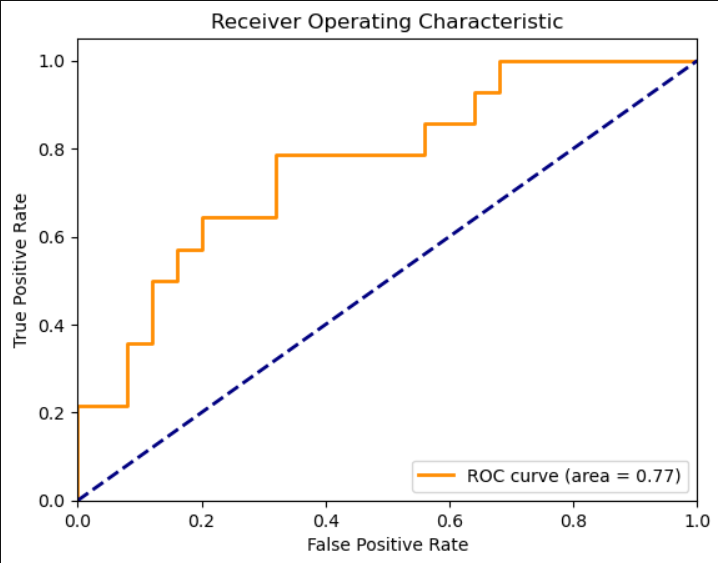
Accuracy train – test: 0.74 – 0.77 AUC-ROC train – test: 0.7741 – 0.7743

ROC chart:



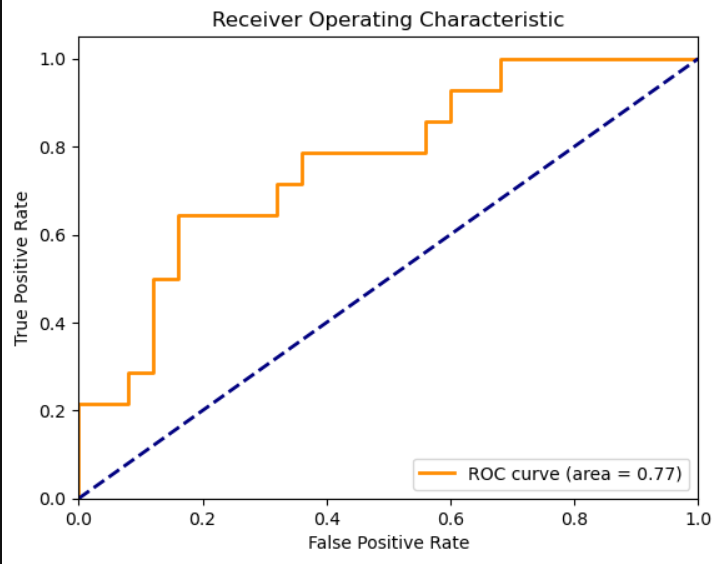
|  |  |  |  |
| --- | --- | --- | --- |
| Default lasso Log regression | Precision | Recall | F1-score |
| 0 | TR:0.73 TE:0.76 | TR:0.89 TE:0.88 | TR:0.80 TE:0.81 |
| 1 | TR:0.77 TE:0.70 | TR:0.53 TE:0.50 | TR:0.63 TE:0.58 |

Accuracy train – test: 0.74 – 0.77 AUC-ROC train – test: 0.7707 – 0.7657



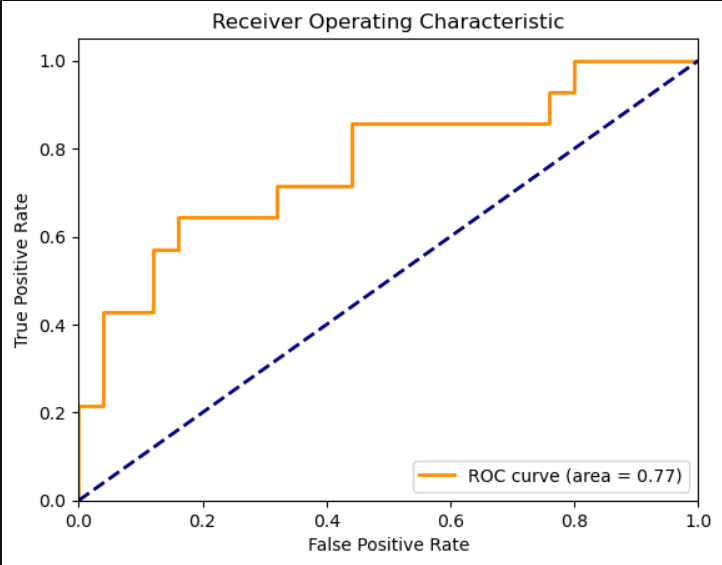
|  |  |  |  |
| --- | --- | --- | --- |
| Default ridge log regression | Precision | Recall | F1-score |
| 0 | TR:0.73 TE:0.76 | TR:0.89 TE:0.88 | TR:0.80 TE:0.81 |
| 1 | TR:0.77 TE:0.70 | TR:0.52 TE:0.50 | TR:0.62 TE:0.58 |

Accuracy train – test: 0.74 – 0.74 AUC-ROC train – test: 0.7372 – 0.7657



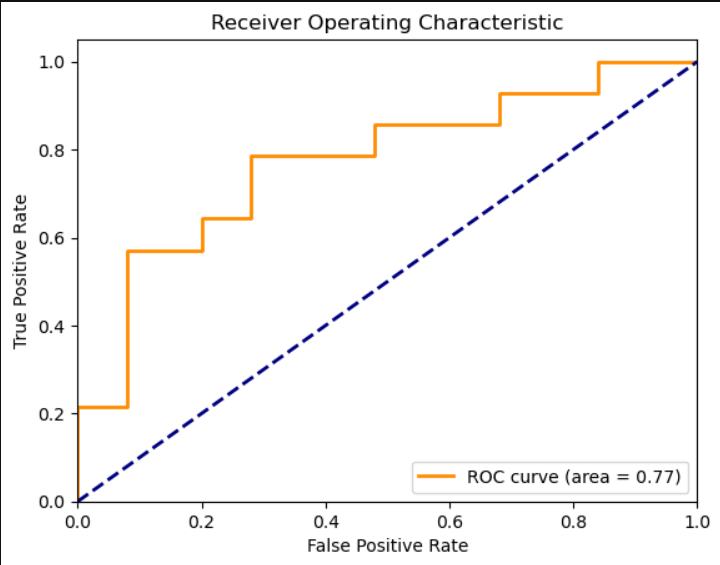
|  |  |  |  |
| --- | --- | --- | --- |
| Hyper parameter Gradient Boosting | Precision | Recall | F1-score |
| 0 | TR:0.85 TE:0.81 | TR:0.97 TE:0.84 | TR:0.90 TE:0.82 |
| 1 | TR:0.94 TE:0.69 | TR:0.75 TE:0.64 | TR:0.83 TE:0.67 |

Accuracy train – test: 0.88 – 0.77 AUC-ROC train – test: 0.9700 – 0.7657



|  |  |  |  |
| --- | --- | --- | --- |
| Baseline logistic regression | Precision | Recall | F1-score |
| 0 | TR:0.76 TE:0.78 | TR:0.89 TE:0.84 | TR:0.82 TE:0.81 |
| 1 | TR:0.79 TE:0.67 | TR:0.59 TE:0.57 | TR:0.68 TE:0.62 |

Accuracy train – test: 0.77 – 0.74 AUC-ROC train – test: 0. 0.7940 – 0.7743



\*Best SMOTE classifier\*

|  |  |  |  |
| --- | --- | --- | --- |
| SMOTE decision tree classifier | Precision | Recall | F1-score |
| 0 | TR:0.78 TE:0.75 | TR:0.90 TE:0.84 | TR:0.84 TE:0.79 |
| 1 | TR:0.82 TE:0.64 | TR:0.64 TE:0.50 | TR:0.72 TE:0.56 |

Accuracy train – test: 0.79 – 0.72 AUC-ROC train – test: 0. 0.8697 – 0.7143

|  |  |  |
| --- | --- | --- |
| Test weighted F1 | Train weighted F1 | Difference |
| 0.7090 | 0.7895 | -0.0805 |

Ranking: According to F1 to strike a balance on preventing both false positive and false negatives (Tie breaker will be an overall evaluation of all F1 score across 1 and 0 on both train and testing)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Position | Name | Test weighted F1 | Train weighted F1 | Difference |
| 1 | Default log reg | 0.7619 | 0.7355 | 0.0264 |
| 2 | Hyper parameter Gradient boosting classifier | 0.7672 | 0.8753 | -0.1081 |
| 3 | Default lasso reg | 0.7317 | 0.7324 | -0.0007 |
| 4 | Baseline log reg | 0.7387 | 0.762 | -0.0233 |
| 5 | Default ridge reg | 0.7317 | 0.7249 | 0.0068 |

Validation & evaluation(classifiers):

Validation method: 80/20 Train-Test split for model validation.

Reason for choice: The low number of rows in my dataset which can cause two problems.

1. The performance estimate on each fold can be more variable due to the limited data in each training set. This can make it difficult to get a reliable picture of the model's true performance on unseen data.
2. Since each training fold has less data, the model might be more prone to overfitting to the specific examples in those folds. This can lead to poor performance on unseen data.

Disclaimer on this validation method:

**I will acknowledge that the 80 – 20 split validation method has its limitation such as a high variance in performance estimates and potential overfitting. To tackle the overfitting risk, I have some hyperparameters and even models itself in mind that could softened the impact.**

\*Report will mostly be referring to main metrics: weighted F1 score & AUC ROC\*

First logistic regression [Baseline]: I first start with a regular logistic regression which will serve as our baseline model. The testing F1 score was 0.7387 and my training F1 score is 0.762. While it is a good start, the model was a little too complex with too many coefficients and I was certain that the performance could be improved.

Second logistic regression [Default]: After taking consideration from things like linear correlation and mixing and matching, I have decided that the column Company, ‘vacancy quantity’, the two columns that I have made which is ‘day’ and ‘month’ and lastly, the column ‘Selected’ which might be confused as target leakage will be excluded. That created our best model, default logistic regression. With the test set having a weighted F1 score value of 0.7619 and a training set that has the F1 score of 0.7355, this model has one of the lowest differences in F1 score between the training and testing set while still maintaining a high F1 score.

[Lasso]: Moving on, I tried out both Lasso and Ridge as well as elastic net which is basically a combination of Lasso and Ridge since regularization is crucial to avoid overfitting, especially when dealing with limited data. While none of the three new models manage to outscore the default model much less improve the interpretability, it still produces a rather remarkable result. The lasso regression model produced a model that is the closest between the testing and training set. With the test F1 score of 0.7317 and the train F1 score of 0.7324, this model shows its minimal overfitting potential, and it is a good sign.

[Ridge]: The next model which is a ridge regression has a F1 score on 0.7317 on testing and 0.7249 on training. This puts a difference of 0.0068 between training and testing which while is worse than the Lasso model, it is still quite acceptable. However, this is the start of a whole series of model producing the exact same result in the F1 score.

[Elastic net]: Case in point, the Elastic net model produced the exact same output in terms of the weighted F1 score. The Elastic Net regression performed similarly to the Ridge regression because the L1 ratio, which controls the balance between Lasso and Ridge regularization within Elastic Net, might have been set to a high value, emphasizing L2-like shrinkage, and reducing the impact of feature selection.

[Hyperparameter induced model]

Hyperparameter used in the final tuned model:

|  |  |  |
| --- | --- | --- |
| Hyperparameter Name | Purpose | Set value |
| Penalty | To discourage model from assigning excessively large weights to features, helps prevent the model from becoming overly complex and susceptible to overfitting the training data. In turn, penalty term encourages the model to learn more generalizable patterns from the data. | ‘L1’, ‘L2’ |
| C | controls how much the model is penalized for complex features, allowing you to fine-tune the balance between fitting the data and avoiding overfitting. | ‘0.001, 0.01, 0.1,1,10 & 100’ |
| Solver | determines the training algorithm, with some prioritizing speed for large datasets, while others prioritize guaranteed convergence which is finding the optimal solution and efficient handling of sparse data which are datasets with many irrelevant features. | ‘Saga’ |
| Maximum iteration | Reason for high max iteration: through some primary tries, the model often runs into convergence runtime issues | ‘10000’ |

Outcome: Despite this change of setting, it still produced the same results. I tried other hyperparameter like playing around with the solver method like changing it to IBFGS and Liblinear and of course change up the penalty appropriately since IBFGS is mismatch with L1 on penalty. But despite those changes, my model did not see any changes. After discovering the amount of hyperparameter that I can explored, I did my best to explore all options, but it still did not change much in terms of my main metrics of weighted F1 score.

General reasoning and explanation for performance: Some reasons why I assume the models perform the way it did in this series of logistic regression model is.

* The initial feature selection in the default model might have already reduced overfitting, leaving less room for further improvement by regularization techniques.
* As for why the model still did not perform as optimal as hoped even with the fine tuning and regularization is because the chosen regularization strength in Lasso and Elastic Net may not have been aggressive enough for significant feature selection. Ridge, by nature, doesn't perform explicit selection.

<End of logistic classifiers & hyperparameter exploration>

Support vector machine:

Reason: Firstly, I was still unsure whether my data might have non-linear relationships. Second reason I made the decision to explore support vector machine next is because SVC is less susceptible to the curse of dimensionality. This is because with appropriate decision on the Kernel selection, it focuses on identifying the most important data point.

Exploration: To start, I created a default Support vector classifier. The testing set produced a weighted F1 score of 0.6781 and the training set produced a score of 0.7383. While it is not ideal and an even worse model as compared to our baseline logistic regression model, I decided to give it another try by implementing some hyperparameter. By adjusting the C value as well as the Kernel and gamma setting, the model still did not see any improvement. It is about this time I started to wonder why no matter my modification to the models, the performance never seemed to increase or decrease from its original. After a bit of investigation, I decided to draw the conclusion that there might be two factors that causes this. Firstly, my data has a very linear relationship between features and the target variable, which explains why the SVC might not be able to outperform Logistic Regression. Due to extended runtimes of the SVCs, I decided to explore a less computationally expensive algorithm.

General reasoning and explanation for performance:

* Linear relationships and choices in models
* Logistic regression generally perform so well could be because it is Well-suited for modelling linear relationships between features and the target variable. (While this is the assumption for now, there will be points throughout this exploration where I might think different)
* As for the support vector classifier, it can be powerful for capturing non-linear relationships through kernels. However, it can be less effective when the underlying relationships are inherently linear.
* Hyperparameter tuning & performance:
* SVC performance didn't significantly improve or worsen with hyperparameter changes shows more evidence of the possibility of inherent linearity in my data.
* Computational cost:
* Between the two types of models, SVCs can be computationally expensive, especially during hyperparameter tuning. Logistic regression is generally faster to train and evaluate, making it a more efficient choice for linear problems.

Decision tree:

Reason: One reason I decided that to explore decision tree after SVCs is also because it offers insights into feature importance, which might be valuable for understanding how the model makes predictions. Additionally, it is far easier to interpret compared to a

Exploration: The first decision tree I create which was default, but it was overfitted with the value of 1 across all metrics. That’s when I begin to manually prune the tree. After a bit of tuning with the max depth and min sample size, the model produce a weighted F1 score of 0.7090 on the testing and 0.7651 on the training set. It was a small but still significant improvement as compared to the SVC. While there is not much more hyperparameter options for me to tune, I attempted to try using entropy as the criterion instead of python default Gini criterion. With the small change, the test score increases to 0.7224 but the training score dropped to 0.7618. I see this as an improvement as it closes the gap between the training and testing set, dropping the possibility of overfitting. I decided that while the best model among the decision tree was not too bad, I still hope to increase the performance a little more.

General reasoning and explanation for performance:

* DT vs LR for linear relationships
* Decision tree can capture complicated non-linear relationships through their branching. However, for linear data, they can become overly complex and prone to overfitting, as seen in my initial default tree.
* Logistic regression is simpler and more efficient for capturing linear relationships. my initial success with logistic regression suggests this might be the case for my data.
* Tuning vs pruning: While pruning and hyperparameter tuning in my decision tree helped reduce overfitting and improve performance, it might not be enough to be better than a well-tuned logistic regression model for linear data.

Random forests:

Reason: The reason I decided to move onto random forests is because decision trees can be susceptible to variance and overfitting, especially on smaller datasets. Random forests address these limitations by averaging predictions from multiple trees with randomness injected during training, resulting in more robust and generalizable models.

Exploration: While I was primarily hoping to increase the overall performance of the model, I was also hoping to close the gap between the train and test score to decrease overfitting which is where random forests have an edge over decision tree. Random forests address these limitations by averaging predictions from multiple trees with randomness injected during training, resulting in more robust and generalizable models. Just like the decision tree, executing random forest without any hyperparameter causes overfitting with the values of the metrics all coming out with 1.00. Just like last time, I begin to experiment with the hyperparameter. The parameters that I tune are very similar to the random forests, max depth, minimum sample split, minimum sample leaf and the max features. However, it proves to be pointless when the test set produced a weighted F1 score of 0.7090 and the train set came out with a score of 0.8466. This is one of the worse models in my opinion not because it has a bad base score but rather, it has too big of a gap between the test and training score.

General reasoning and explanation for the performance:

* Underlying data linearity: When a linear relationship is present between the target and the features, logistic regression is still stronger but Random forests, while robust, can be overly complex for linear problems. Their strength lies in capturing non-linearities through ensemble learning. In a linear setting, this complexity might not translate to significant performance improvement over logistic regression.
* Overfitting and hyperparameter: Despite tuning hyperparameters like max depth and minimum sample size, the large gap between training and testing F1 scores in my random forest suggests overfitting. Random forests generally require more careful tuning compared to logistic regression to avoid overfitting, especially with smaller datasets.

Gradient boosting machine (GBM):

Reason: To combat the overfitted score of my final random forest classifier, I decided to explore gradient boosting models. GBMs are sequential learning models where each tree learns from the errors of the previous ones, resulting in models that can generalize better and potentially achieve higher performance than the random forests, especially when the overfitting issue is severe.

Exploration: As always, I begin with a default model which just like the pass few models are experiencing overfitting with almost all the metrics reaching values of 1.00. To move on, I created another model with necessary hyperparameters set. By tuning the n estimators, learning rate and max depth, I hope to close the gap between the test and training score while improving the overall performance. That produced a test score of 0.7672 and a train score of 0.8753. While it did increase the overall performance as compared to the other models, it has one of the highest differences between the test and train set indicating severe overfitting compared to the other models. After this, I looked at the records of all my models together.

General reasoning and explanation for the performance:

* Pure performance vs overfitting: Even though my GBM might have achieved a higher test score, the significant gap between training and testing scores suggests severe overfitting.

Reason for such

* Each tree in a GBM builds on the errors of the previous one. This can lead to the model memorizing noise in the training data if not carefully controlled.
* Tuning hyperparameters like learning rate, n estimators (number of trees), and max depth can be more complex in GBMs compared to logistic regression. Finding the right balance to prevent overfitting can be challenging.

Overall evaluation of current models (pt 1): The consensus was that the values for my weighted F1 score would stay between 0.6 – 0.8 with the modified gradient boosting model having the highest F1 score of the training set with a value of 0.8753. In our context where I am identifying the number of successful employee placement in different companies, an average of 0.7 of weighted F1 score suggest a rather reasonable balance between precisions and recall meaning that it can strike a very good balance when it comes to detecting false positives and false negatives. Considering that this model which will be implemented into a forecast visualization stage, a slight miss rate in either precision or recall might be tolerable for initial exploration. It can still provide valuable insights into placement trends.

Class imbalance exploration & mending techniques:

While exploring methods to fix class imbalance does not guarantee model performance improvement, I felt that it would be beneficial to do so. While researching methods that I can use to help with class imbalance, I came across two main types of methods, Oversampling and under sampling. But I quickly decided that oversampling would be the way to go since my dataset was very small. By using oversampling, I can “create” more data by increasing the number of samples in the minority class which will be able to capture the underlying relationships within the minority class. I then dug deeper on the different methods that follow the idea of oversampling. The two methods I saw was random oversampling as per the name means that duplicate data points from the minority class randomly. While this is simple, it can lead to overfitting. This did not seem very optimal which made me use the second method, the Synthetic Minority Over-sampling Technique (SMOTE) method. This method Creates synthetic data points for the minority class by interpolating between existing minority class samples. While reading up on SMOTE, I learn of a combination method called SMOTE and ENN. Essentially, this approach combines SMOTE for oversampling the minority class with Edited Nearest Neighbours for under sampling the majority class. ENN removes majority class neighbours that are too close to minority class samples. However, that did not turn out as intended and therefore I decided to stick with the default SMOTE method.

SMOTE infused model exploration:

Baseline model: I begin the application of SMOTE onto my baseline model, just to observe the general effectiveness. That produced a result of 0.6399 on the testing set and 0.7411 on the training set. With one of the worse test set results, it did not seem to be performing as well as intended.

Further consideration: While the immediate assumption I had was that the synthetic data itself was noisy or even producing nonlinear relationships, I decided to evaluate one of each of my different models. Among the 6 different models that I tried to apply SMOTE to like gradient boosting, Random Forest, SVM, Log reg, Entropy decision tree classifier and regular decision tree. It is important to note that I chose the best performing models out of the different model classes and stacked SMOTE on top of it.

Reason: I chose not to apply it to every single one of my existing models is because while I do acknowledge that there is class imbalance in my data, I do not think that the dataset has too severe class imbalance to begin with, the benefits of SMOTE might be minimal. Applying it to already balanced data might not significantly improve performance and could even introduce noise. In addition, SMOTE while mostly beneficial, can have different impacts on the model performance itself. If the existing data doesn't capture the key factors influencing the target variable, the synthetic data points created by SMOTE might not be informative for all models. For example, logistic regression assumes a linear relationship. If SMOTE creates data points with complex non-linear relationships, logistic regression might not benefit from them.

Result:

The Gini decision tree ended up performing best out of the SMOTE bunch. With F1 testing score of 0.7090 and F1 training score of 0.7895. This is the best model for two reasons. One, it has the lowest gap between the training and testing score of 0.08 and second, it has the highest on both training and testing F1 score. The ranking goes as the Gini decision tree first, entropy decision tree, SVM, random forest, logistic regression and lastly the gradient boosting. I decided to explore further to find the reasoning behind the performance difference when SMOTE is applied VS when it is not. Due to horrible consistency, the only explanation I could draw from this difference in performance is because SMOTE can improve performance for models dealing with class imbalance, but overfitting and model complexity can lead to unexpected results. The actual ranking change after applying SMOTE depends heavily on the base nature of our data.

<End of all model exploration>

Conclusion: To reiterate, I evaluated model performance using weighted F1-score, aiming for a balance between precision and recall in identifying successful placements. The default logistic result with feature selection is the best performing model out of everything I have tried. This shows that the data likely exhibits a linear relationship between features and the target variable, limiting the benefit of more complex models. This could explain the way that the more complex models like SVC, Random Forests and Gradient boosting tend to overfit. I also think that the small dataset size also contributes to this phenomenon. And lastly for SMOTE implementation, the already overfitted and complicated model might have contributed to the inconsistent but general suboptimal performance. For any future improvement, the primary goal is to attain more volume of data since one of the reasons why the models perform quite erratically especially in large dataset is because of the smaller volume of data size.

***FINAL MODEL SELECTED & REASON:*** Default logistic regression.

Reason:

* The strong and consistent performance across various evaluation metrics, shows that it can effectively captured the underlying relationships in my data.
* The simple interpretability of the logistic regression model makes it perfect for my stakeholders who are inexperience in the field of ML applications but can understand its implication.
* Compared to more complex models like random forests and gradient boosting, it exhibited less susceptibility to overfitting, making it a more robust choice, especially for smaller datasets.

Coefficients & additional information:

Industry: 0.0860415001425646

Interviewed: 0.09652157639788385

KIV: -0.07355498187746679

Rejected: -0.04441902082480365

Rank & file: -1.7841005220576318

PMET: -1.246152326452792

Region: 0.10689176161838747

Method: 0.28242651142206654

Sengkang & Others: 0.027400911851762192

Pasir ris: -0.0008663352045159323

Punggol: -0.03902708687694495

Method to implement model result: we opted to utilize the currently available data for model development and prediction. To do this, I transformed the target variable into a binary format suitable for the chosen model. Subsequently, the model was trained, and a new column was generated containing the predicted values for the binary target variable.

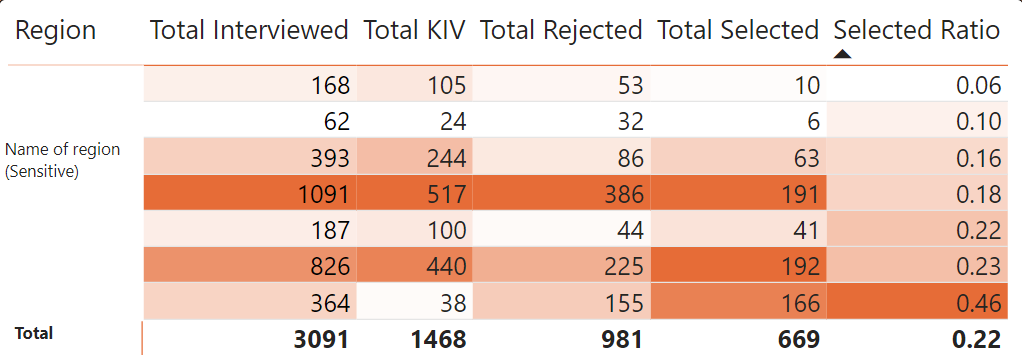
***With the model prediction set in the form of a new column and the data tidied, I was ready to start with data visualization.***

# Chapter 4: Visualization & business intelligence

Dashboard Creation

Dashboard creation thought process: Under the guidance of my supervisor, I was informed that analysis and sharing is often done through simple screenshots, I had to try to cover as much ground as I could in a single dashboard in power BI to provide a concrete but still overall picture to the project SUCCESS employment results. To start, I created a simple heat map, in the form of a table chart. This chart contains information on the 6 divisions involved and 4 different measurements. Total interviewed, total KIV, total Selected, Total Rejected and selection Ratio. The last measurement is created by creating a new measure using the formula: Selected Ratio = DIVIDE(SUM('Sheet1'[Selected]), SUM('Sheet1'[Interviewed]), 0)

This is to put into perspective how many people did manage to get selected for the job that they interviewed for making it a much simpler process for stakeholders to view it. I am aware that I rejected this transformation back before ML application, but it was as mentioned, this would have been too closed to target leakage. The reason for this format of creation of chart is for an easy overall analysis since it contains the few key measurements that the higher ups are concerned with. The presentation format of a heatmap is a way to ensure straightforward interpretation and understanding. The purpose of the chart is to provide a quick overview of hiring trends across the 6 divisions involved in Project SUCCESS. It allows higher-ups to identify potential areas for improvement in the recruitment process. One of the main insights that this chart can provide is that by comparing the intensity of colours for "Total Interviewed" and "Total Selected" can reveal interview: Selected ratios across divisions. This might indicate interview effectiveness or potential hindrance in the selection process.



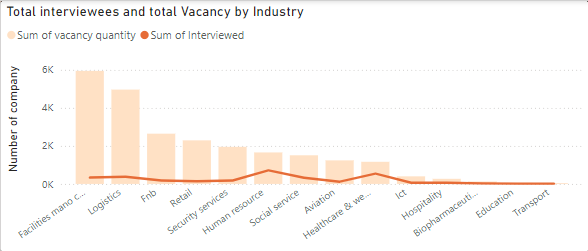
Key insights from heatmap: It reveals significant regional variations in the number of interviewed personnel at Project SUCCESS offices.

* **High Demand in Sengkang CC:** Two regions stand out with a high number of interviewed personnel – Sengkang CC being the highest. This can be attributed to several factors.
* **Large Population:** Sengkang is one of the most heavily populated districts in Singapore, creating a natural demand for job opportunities. (Source: The Business Times, 2023)
* **New Hospital:** The recent opening of Sengkang General Hospital likely contributed to the high demand, as they might have held career fairs with Project SUCCESS to recruit staff.
* Low demand in Punggol West CC: Punggol West has the lowest number of interviewed personnel. This could be due to:
* Non-Mature Estate: Punggol West is a relatively new residential area, leading to a smaller resident pool compared to more established areas.
* Small office: As compared to the remaining CC which have a far bigger building as its community centre, Punggol west CC is only a small office located at the void deck of a flat. This might make it difficult for people to notice and in turn learn about the project SUCCESS office in the CC.

Selection Rate and Candidate Pool: However, the number of selected candidates doesn't strictly follow the number of interviewed personnel.

* **High Selection Rate at One Punggol CC:** Despite a moderate number of interviewed personnel, One Punggol CC exhibits a high selection rate (0.46 compared to Punggol West's 0.18). This suggests a strong candidate pool with skills and experience closely aligned with job requirements.
* Potential reason for high selection rate:
* **Largest Branch:** One Punggol CC being the biggest and "official" branch in Punggol might attract higher-quality applicants.
* **Demographics:** Punggol has a high concentration of residents in the 30-39 age range, a demographic known for peak performance and experience, making them ideal candidates for job placement. (Source: City Population)

Line & stacked column chart: Next, a line and stack column chart. This chart’s X axis is the different industry in which the companies are categorized into, the Column chart Y axis is the total vacancy available, and the line chart Y axis is the total number of people interviewed for that industry. The purpose of this chart is to understand the dynamics between the industry. It can also be used to identify potential skill gaps by showing industries with high vacancies but low interview numbers. This could indicate a shortage of qualified candidates in those specific sectors. Since the stacked column chart represent relative demand for workers in each industry by showing the total number of vacancies available and the line chart indicates the overall candidate availability for those industries by showing the total number of people interviewed, this provides a clearer picture of the supply and demand.



Insights & implication:

Industry demand analysis: The bar chart reveals Facilities Management, Logistics, and F&B as the top three industries with the highest job vacancy count.

* **Facilities Management:** A major corporation within this category likely skews the data due to its diverse service offerings requiring a large workforce. E.g., one major company offers numerous services like disinfection service, smart facility management etc.
* **Logistics:** An article from The Straits Times highlights the industry-wide manpower shortage, particularly for "rank and file" positions. This shortage is likely due to a lack of young people entering the field and an aging workforce.
* **F&B:** As reported by Today Online, long hours, physical demands, and relatively low wages contribute to the high manpower demand in this sector. As education levels rise and alternative opportunities emerge, attracting younger workers becomes challenging. F&B also suffers the same issue as logistics which is that the present workforce is aging rapidly.

Interview trends: The line chart shows a lower number of interviewees for F&B and Logistics compared to vacancies, further supporting the high demand of these industries.

High interview rates industry:

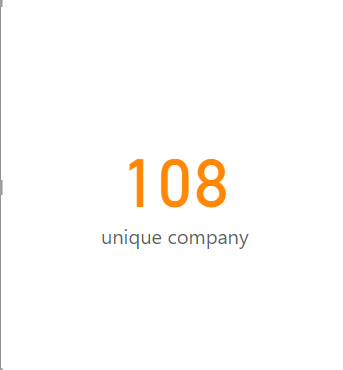
* Human resource: This industry has the highest number of interviews, most likely due to the diverse and engaging nature of the work. In the form of meeting new people and potentially engaging in unique job roles on a daily if not monthly basis. An article by BBC shows the importance of job fulfilment for young professionals, which HR roles often provide.
* Healthcare: The high number of interviews in healthcare likely stems from Singapore's aging population, leading to increased job demand and a stable job market. Government investments in the sector further contribute to its attractiveness, offering competitive salaries, good working environments, and benefits.

Low interview rates industry:

* Pharmaceutical and education: These sectors likely exhibit a balanced supply and demand for workers, resulting in fewer open positions. Alongside high probability of strict education and experience qualifications.
* Transportation: Since the only company that has listed a job position for the transportation industry is a public transport company, the selection processes and requirements might be stricter due to security and responsibility concern.

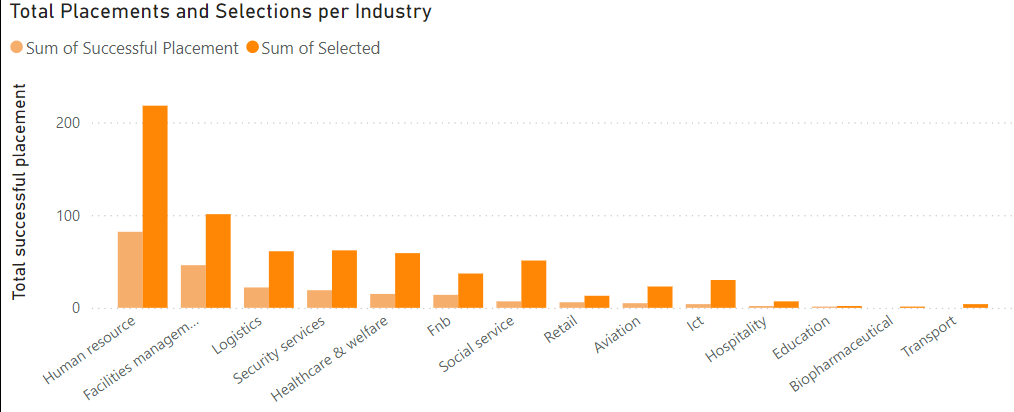
Overall, the data shows a mismatch between job vacancies and available talent in certain industries. Understanding the factors behind high demand lack of young talents such as unattractive salaries/benefits, long hours and high physical demand can inform targeted solutions to bridge this gap.

Card chart: The next chart I introduce is a simple card chart that just labels the sum of company. Due to power BI full intractability between all the charts, should the higher ups choose to narrow done the number of companies that fall under certain condition, it will be available. This serves as a headline figure, providing a quick overview of the total number of companies involved in Project SUCCESS and establish the overall scope of the data analysed in the subsequent charts within the dashboard. While the card chart itself doesn't offer extensive insights, it provides a foundational data point for understanding the reach of Project SUCCESS. A high number of companies could indicate a wider impact on the employment landscape. [Default card value]



\*Indicator insights\*

Clustered column chart: Moving on, I created a clustered column chart with the industry as the X axis, the total number of Successful Placement as the primary Y axis and the Total number of Selected as the Secondary Y axis. Some potential insights that can be gain from this is that it allows comparison of the success rates of the recruitment process across different industries. Industries with a high ratio of successful placements to selections indicate efficient hiring practices and good job-candidate matches. The gap between selections and placements highlights industries where candidates might be declining offers. This could be due to factors like better offers elsewhere, mismatch with expectations, or industry-specific reasons.

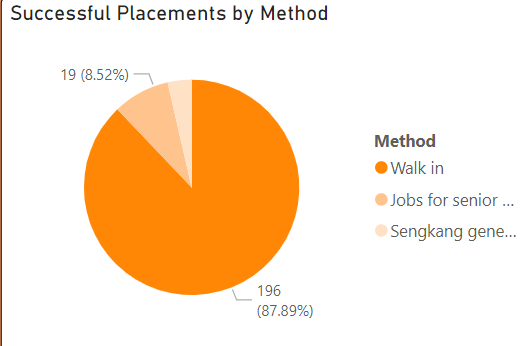


Insights & implications:

Successful placements by industry compares the number of "selected" candidates which are those who were offered the job to those "successfully placed" who are the ones that accepted the job.

* Human Resources: This industry still tops the chart for successful placements, likely due to the high number of selections and the diverse, stimulating nature of HR roles as I have explained in the line and bar chart above.
* Facilities management: Although Facilities Management has high selections, the number of successful placements is lower compared to HR. This can be very simply due to much stricter technical and potentially educational qualification which leads to a higher number of drops – off rate after selection. In comparison, human resource is most probably more heavily reliant on soft skills such as communication and management making it a less demanding industry.
* **F&B:** F&B falls in the middle, likely reflecting the challenges discussed previously – long hours, physical demands, and low wages deterring potential employees. However, it is also fair to say that it is the go-to industry for people with lower educational qualification or people looking for a short-term part time.

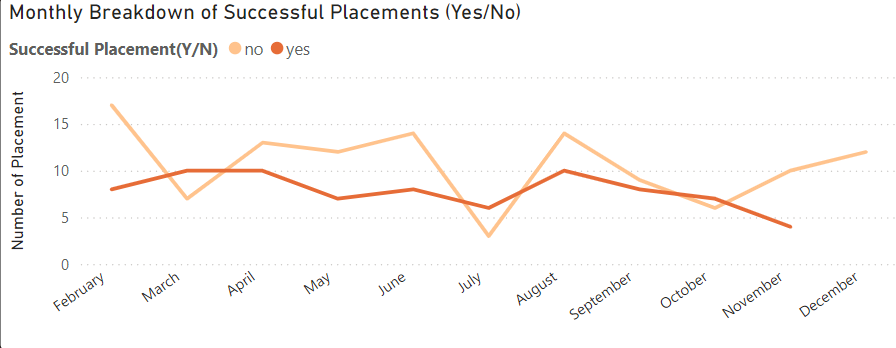
Pie chart: Following that, a simple pie chart that represents the proportion of people who learn of project SUCCESS through the 3 different methods. I chose a simple pie chart for this since it can be easily understood and easy to focus on the composition. In the labels of the Pie chart, there is a percentage calculation to give an easy numeral representation of the value of each factor in the overall composition. Insights that are gained from this chart could be used to boost the awareness of the project itself for more people to visit should they need a job in the future. It can also provide insights in underutilized channel.



Insights & implications:

* Methods of approach & successful placement: This pie chart analyses the percentage of successful placements by their approached method at Project SUCCESS (Walk-in, Sengkang General Hospital Career Fair, Jobs for Seniors Fair).
* **Walk-in:** As expected, walk-ins account for the majority of successful placements with over 75%. This likely reflects the convenience of multiple branch locations and the simplicity of walk-in processes.
* Targeted fairs: While the two other methods have a smaller share, they still contribute to successful placements. These fairs offer focused outreach to specific demographics (medical professionals and seniors) and achieve decent placement rates.

Line chart: The next is a monthly breakdown of the number of Successful placements in total. It is represented with Yes and No so there will be two lines on the line chart. A time chart is always represented in a line chart it can reveal trends, patterns, and outliers in a clear and intuitive way. I decide to split them up into Yes and No instead of just using one simple line because with the different lines representing yes and no, stakeholders and higher ups can help identify potential causes for fluctuations in success rates and inform strategies for improvement.

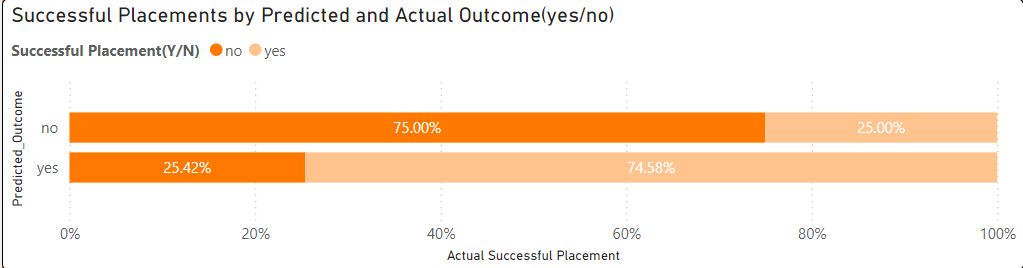


Insights & implications:

Placement trends and anomalies: The chart displays the number of successful placements compared to failed placements over time. While the number of unsuccessful placements is generally higher, there are anomalies in March, July, and October.

* March spike: This is due to the two career fairs I have mentioned previously. Targeted outreach through fairs can lead to a significant increase in successful placements, as shown by the data.
* July & October rises: While there are no major events are signs in the data that could clearly indicate the reason, I think some possible explanation is that the quality of applicants during these months could contribute to higher success rates. Second, Companies might have implemented more efficient or effective hiring processes during these periods, leading to better selection choices.

100% Stacked bar chart: Next is a 100% stacked bar chart. This is the part of the dashboard where I reflect to the higher ups and stakeholder on my model accuracy. This graph is created Using the Y axis as the predicted outcome from my model, the X axis is the count of Successful placement in binary and the legend is the same as the X axis. Essentially, it plots out the idea of true positive, false positive, true negative and false negative into an easy layman representation. The purpose of this chart is to give a straightforward breakdown of my model performance to the higher ups through a simple and easy to understand visual representation of my model performance. By assessing the model performance, the higher ups could potentially put the model into use on future data and to use it in a more practical sense. This could lead to future decision making and proper course of action suggested by the model’s prediction. Though they will also need to be warned and reminded of the limitation of the model’s performance that all of its prediction should be taken as a prediction rather than matter of fact.



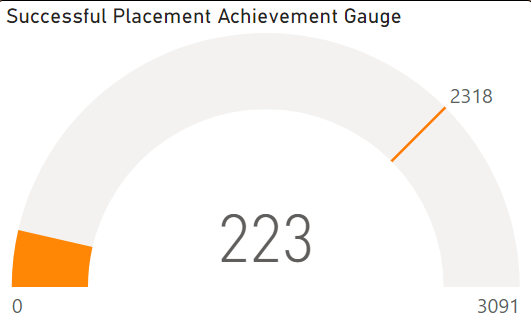
Method & breakdown: My model’s performance It is represented like a regular confusion matrix. However true positive, false positive, true negative and false negative might be a little confusing, so I decide to just represent it as such and in a more layman sense.

There are two bars, the Y axis is the predicted outcome separated into yes and no and the X axis is percentage. Instead, each of the bar are separated into two colours. In the predicted Yes bar, there is a shaded portion where the actual outcome is Yes. The unshaded portion is where the actual outcome is No. This is the general appearance for the other bar as well. So, when user hovers over the different part of each chart, they will be able to see which is which. In a more technical term, the model’s is as such.

|  |  |
| --- | --- |
| True positive | False Positive |
| 74.58% | 25.45% |
| True negative | False negative |
| 75% | 25% |

In general, this is good news for my stakeholders with whether the individual job listed will be able to successfully get people placed for it. In most cases, the model can correctly identify both people who will and won't get the job. (High True Positives and True Negatives). Of course, this ratio generally means that about a quarter of the time, the model’s prediction will be wrong.

Gauge chart: The next chart is a simple gauge chart which shows the number of successful placements as the values from the number of people who went for interviews which is the total of the chart. As per the organization’s target, the optimal scenario is that 75% of all interviewees can be successfully placed into the position. To create the target value, I created a quick value by simply writing a code that calculate 0.75 of the successful placement value and place that in the gauge chart. This way, it is fluid in the sense that whenever a new set of data is introduced in the future, the target will change accordingly and is not just a stagnant number. This can create an easy point of reference for the user and stakeholders to view their progress on the situation and the target that the organization has set.



\*Indicator insights\*

Scatter plot: The last chart I created was a scatter plot that has the total number of people interviewed as the Y axis and the total number of people who are successfully placed as the X axis and the points are separated according to their industries. This scatter plot reveals the relationship between the number of candidates selected and the number who ultimately achieved successful placement. The insights that this can provide is that it can highlight imbalance issues between people who were selected but ultimately failed to allocate into the position they interviewed for. It highlights industries where candidates might be declining offers. This could be due to factors like better offers elsewhere, mismatch with expectations, or industry-specific reasons. Through analysis of this scatterplot, they can identify industries where a high number of candidates are interviewed but successful placements remain low which could suggest a potential issue with offer declines in those specific industries. There is also opportunity of potential comparison between the placement success rates across different industries. In addition, this chart helps to support the supply and demand image for our stakeholders to truly understand the general preference industry and the availability of each industry in correspondence. Last is outlier identification. The scatter plot can help identify outlier data points that deviate significantly from the overall trend. These outliers might represent specific companies or industries that require further investigation to understand the underlying reasons behind their placement success rates



Insights & implications:

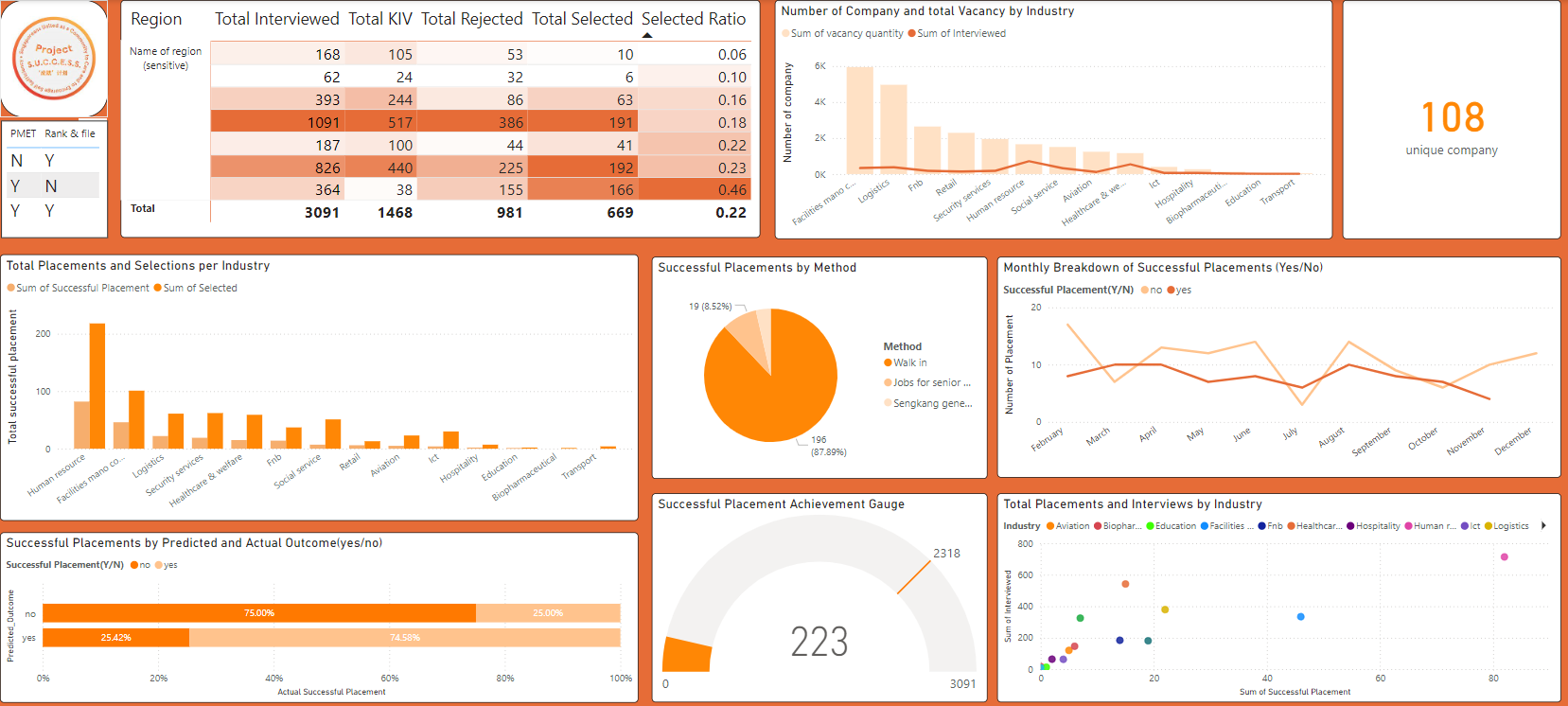
* Supply & demand in successful placement: The main purpose of this scatter plot is to examines the relationship between the number of people interviewed and the number successfully placed across different industries.
* **As to be expected, human resource** dominates the chart with both the highest number of interviews and successful placements, confirming its strong demand and ample qualified applicants.
* Next which also does not come as a surprise, despite showing half the interview volume of HR, Facilities Management achieves a respectable placement rate. Compared to industries like healthcare and logistics, its success rate is notably higher.
* Lower demand industry: Biopharmaceutical and Transport, previously identified as having stricter requirements or limited vacancies, are reflected here. While interview numbers are lower, some qualified applicants are present, suggesting the high standards for these positions. The public transport company in particular likely has a rigorous hiring process due to safety concerns such as medical checks and background checks and requires specific skills like punctuality, technical knowledge, customer service.

<End of dashboarding>

Business Intelligence (Actionable insights):

* Investigate the reasons behind unsuccessful placements across all industries. This could involve exit interviews or surveys to understand why selected candidates withdraw in the event a company has selected them, but they choose to withdraw.
* Develop targeted strategies to retain selected candidates based on industry-specific challenges. This could involve offering optimal compensation packages, career development opportunities, or promises of a more positive work environment.
* Consider factors beyond qualifications that might impact placement from the point of view of the interviewees. This could include commute times, company culture, or work-life balance.
* Partner and engage with employers to address industry-specific challenges that contribute to low placement rates.
* Ensure that branches remain accessible and welcoming for walk-in candidates. Consider partnering with other CCs to see if they are willing to participate in the project SUCCESS exercise.
* Explore alternative outreach methods like online job portals or social media campaigns to attract a wider audience, particularly younger demographics.
* Analyse reasons for unsuccessful placements which could involve improving candidate matching, interview processes, or job descriptions to ensure better alignment with employer requirements.
* Consider external factors that might have influenced hiring patterns in July and October, such as economic conditions or government initiatives impacting specific industries.
* Utilize the model selected with caution, while its performance is optimal, do operate under the assumption that it is a strong predictor rather than a definite result.

Final dashboard:



# Chapter 5: Additional configuration

Uploading dashboard onto power BI personal workspace as report -> Set up a community workspace for shareholders, higher ups, and related parties -> Uploaded the dashboard onto the workspace + data -> Set up power automate for basic automation processes.

Purpose for upload onto Power BI workspace:

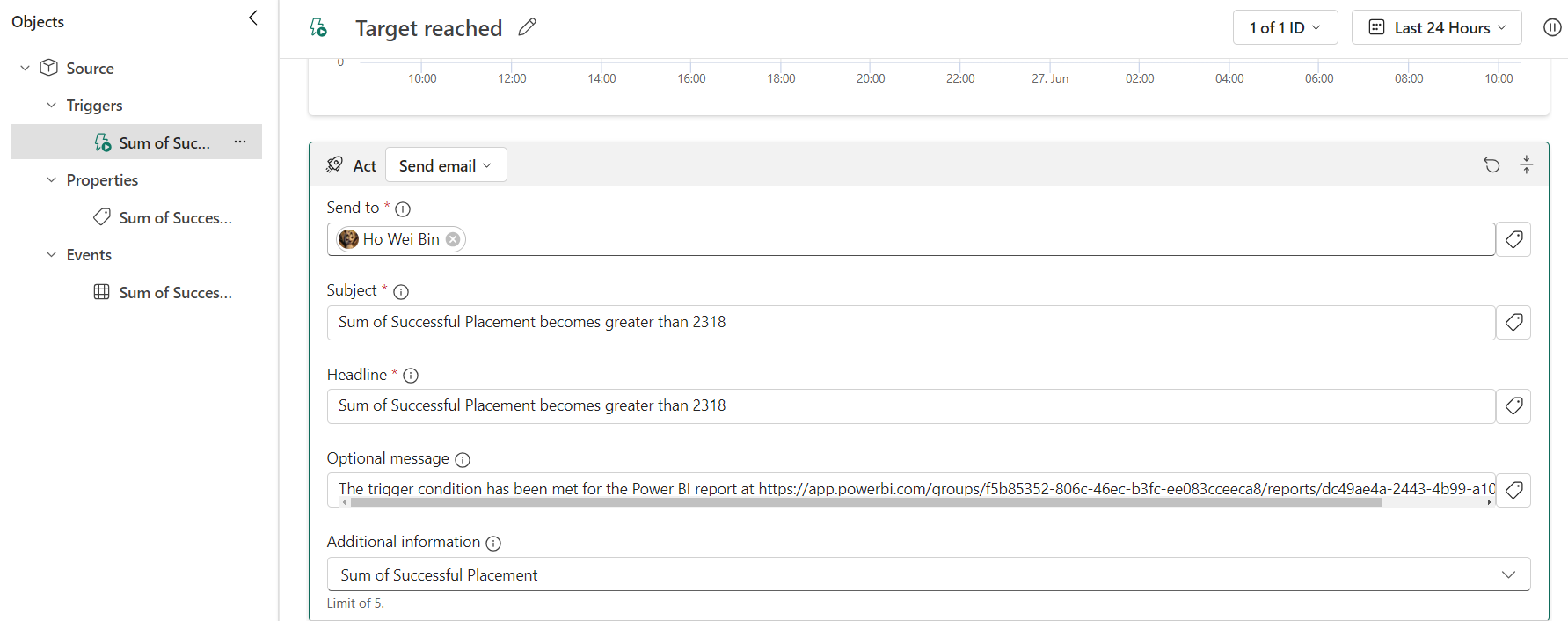
* Centralized Access and Collaboration:
* Everyone accesses the latest data and insights in one central location, minimizing confusion from scattered files or data version.
* Row-level security, a function of power BI platform, restricts access based on user permissions, ensuring only authorized personnel can view or modify reports. Increasing security of the data in turn, eliminating the chance for unauthorized personnel to gain access to the information.
* Power BI tracks changes made to data and visualizations, allowing users to revert to previous versions if needed, valuable for collaborative work granting well informed version control.
* Improved collaboration and efficiency:
* Multiple users can access and interact with the same reports and at the same time, allowing for better teamwork and knowledge sharing.
* Potential automation: Power Automate can schedule data refreshes, ensuring timely updates and maintaining accurate information.
* Configurable Alerts: Notification alerts can be set up for specific charts when they reach critical thresholds, allowing for quick responses to positive or negative trends.

Alert system: I incorporated a notification system triggered when the gauge chart reaches its target.

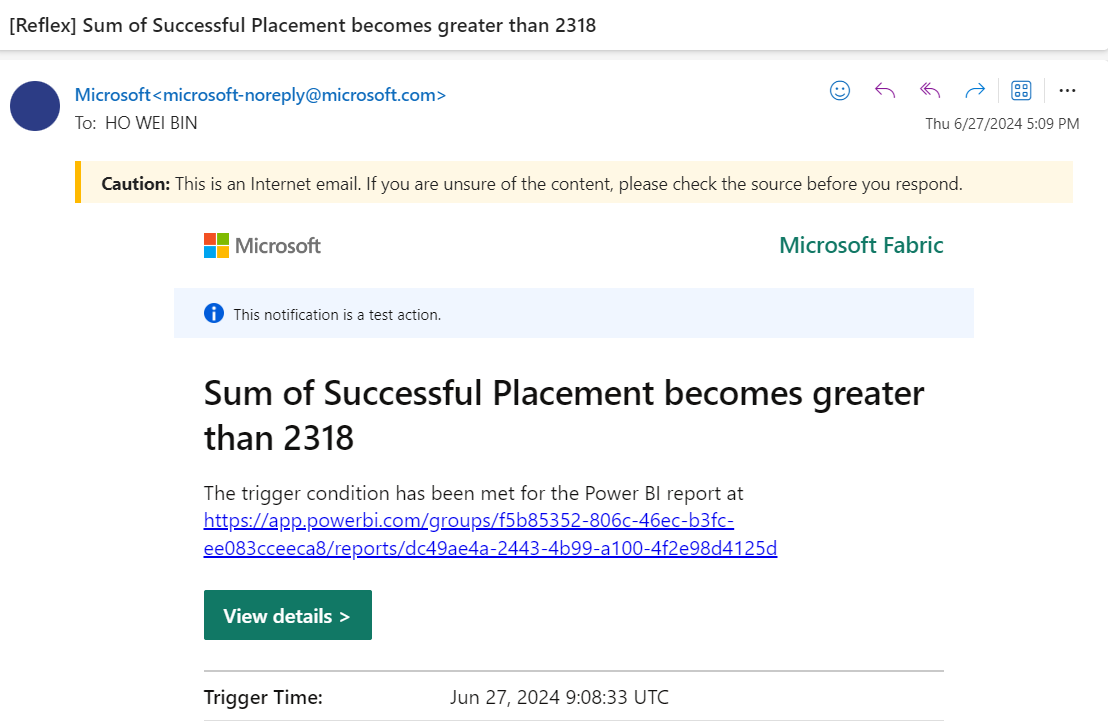
Purpose:

* Reduced Monitoring: Removes the need for constant monitoring by stakeholders, saving them time and effort.
* Actionable Insights: Notifies prompt users to analyse successful practices or plan strategies to maintain the achievement.
* Clear Communication: Gives a clear and concise message about reaching the target, reducing any confusion about progress.
* Enhanced Transparency: Alerts shows active monitoring of progress and achieving goals, building trust with stakeholders.

Configuration:



Test notification:



Auto data refresh(scheduled):

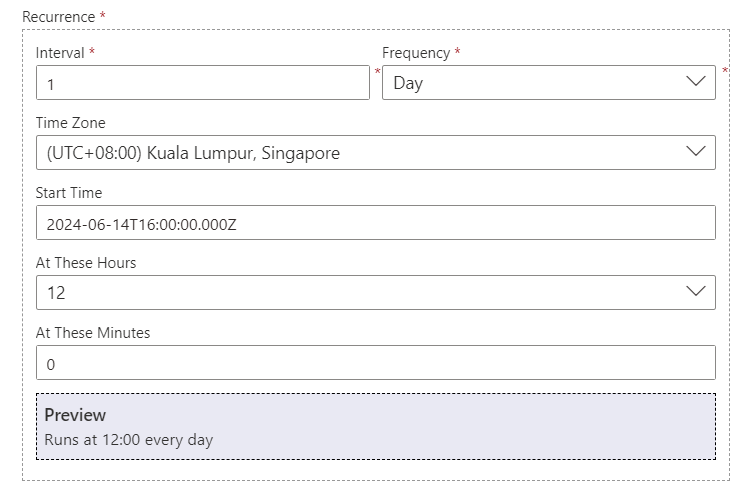
Process

* **Scheduled Flow:** A Power Automate flow is created with a recurrence trigger set to daily at 12:00 PM Singapore/KL time zone.
* **Community Workspace:** The flow refreshes the dataset within the designated community workspace, ensuring all authorized personnel have access to the latest data. **(Note: Due to limitations, this explanation assumes a temporary personal workspace setup with the same configuration)**

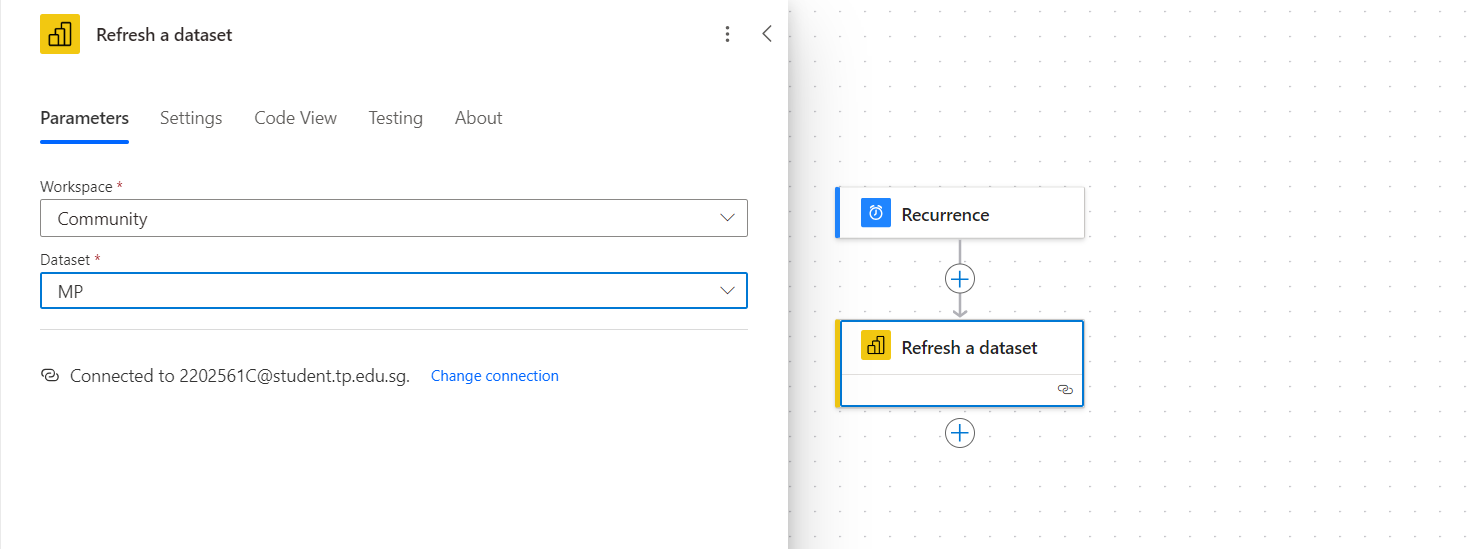
Benefit

* Gives valuable time for data engineers and analysts to focus on other tasks.
* Users have immediate access to the latest data, reducing wait times and allowing them to work more efficiently.

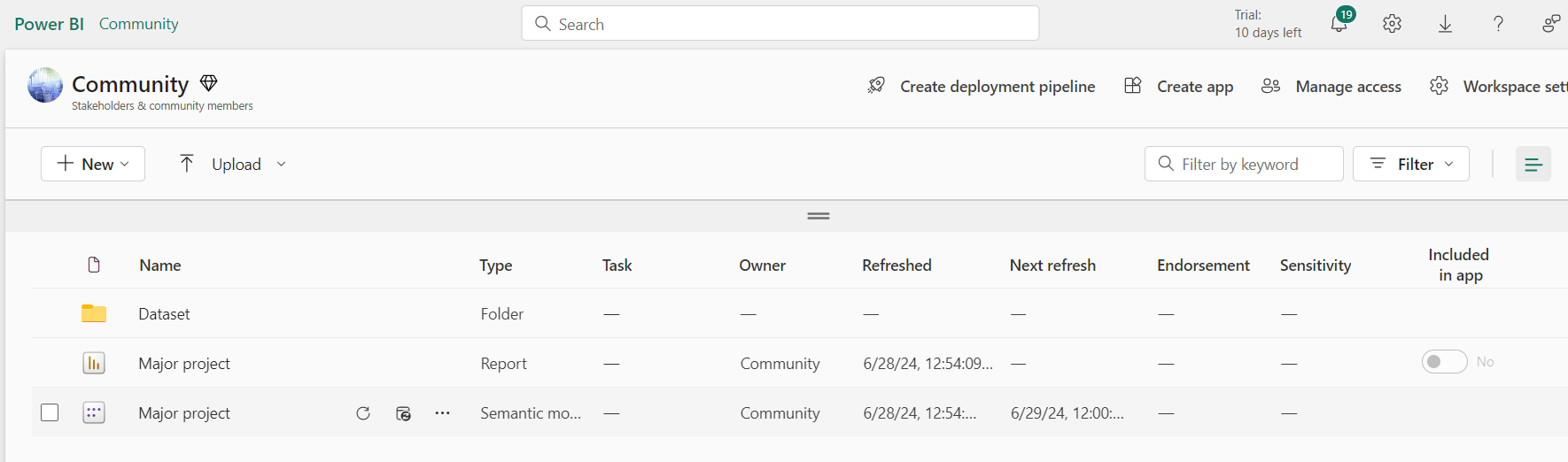
Recurrence node configuration (power automate):



Refresh a dataset node configuration (power automate):



Outcome (power BI):



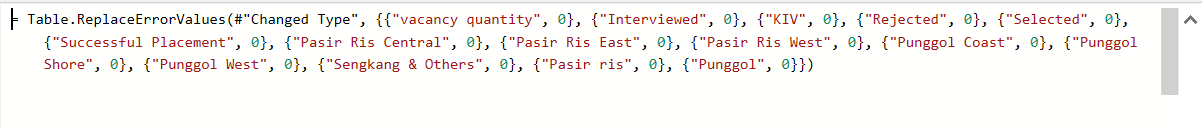
Data imputation via power query:

Process: Utilizing power query in power BI desktop to write my M function for data imputation. Since the data is always vetted and checked by the organization, the only acceptable missing values or invalid values for numeric data is to be imputed as 0, as to not bleach the original message behind the data.

Benefit:

* **Automates Repetitive Tasks:** No need to manually impute missing values every time you work with the data. To save time for both data analyst and end user.
* Standard approach and improved data quality: Ensures consistent data imputation logic across all reports based on my M function definition. This minimizes the risk of errors or inconsistencies in how missing values are handled. By imputing, it fills in missing values, potentially leading to more accurate analysis and insights.

Configuration:



Row level permission:

Process:

* Access manages roles under modelling tab in power BI desktop. Create new roles to define different data access levels for users in my report. Each role represents a specific set of permissions to view certain rows of data. I created roles for each of my regions so that authorized personnel could only view the dashboard containing information of their own region.
* Defining the security level filters by naming the roles, selecting the table, and filtering the data according to the Region column. Repeat the same process for the other roles.

Benefit:

* Enhanced data security: restricting data access allows users to only see the data relevant to their roles and permissions, minimizing the risk of exposing sensitive information.
* Reduce risk of errors during analysis: By limiting data access, RLS prevents users from making decisions based on irrelevant or inaccurate data.
* Simplified analysis: When using this, Users only see the data they need, leading to cleaner and less cluttered reports. This allows them to focus on analysing the information relevant to their tasks and responsibilities. It also means that users don't need to apply additional filters themselves. This streamlines the analysis process and reduces the time spent manipulating data before getting to insights.
* Standardizing data access: Ensures consistent data access rules across my reports. This ensures everyone within the organization sees the data they should base on their roles, promoting data governance and compliance with regulations.

Subscription report configuration:

Process: Using power Bi platform service, I created a secure and customizable method for report delivery.

* Connect report to recipient’s data source.
* Configure dynamic filtering based on specific regions.

(E.g., employees in Pasir Ris Elias CC only receive reports with data filtered to their region.)

* set up a customized schedule for report delivery, ensuring stakeholders and employees receive timely updates.

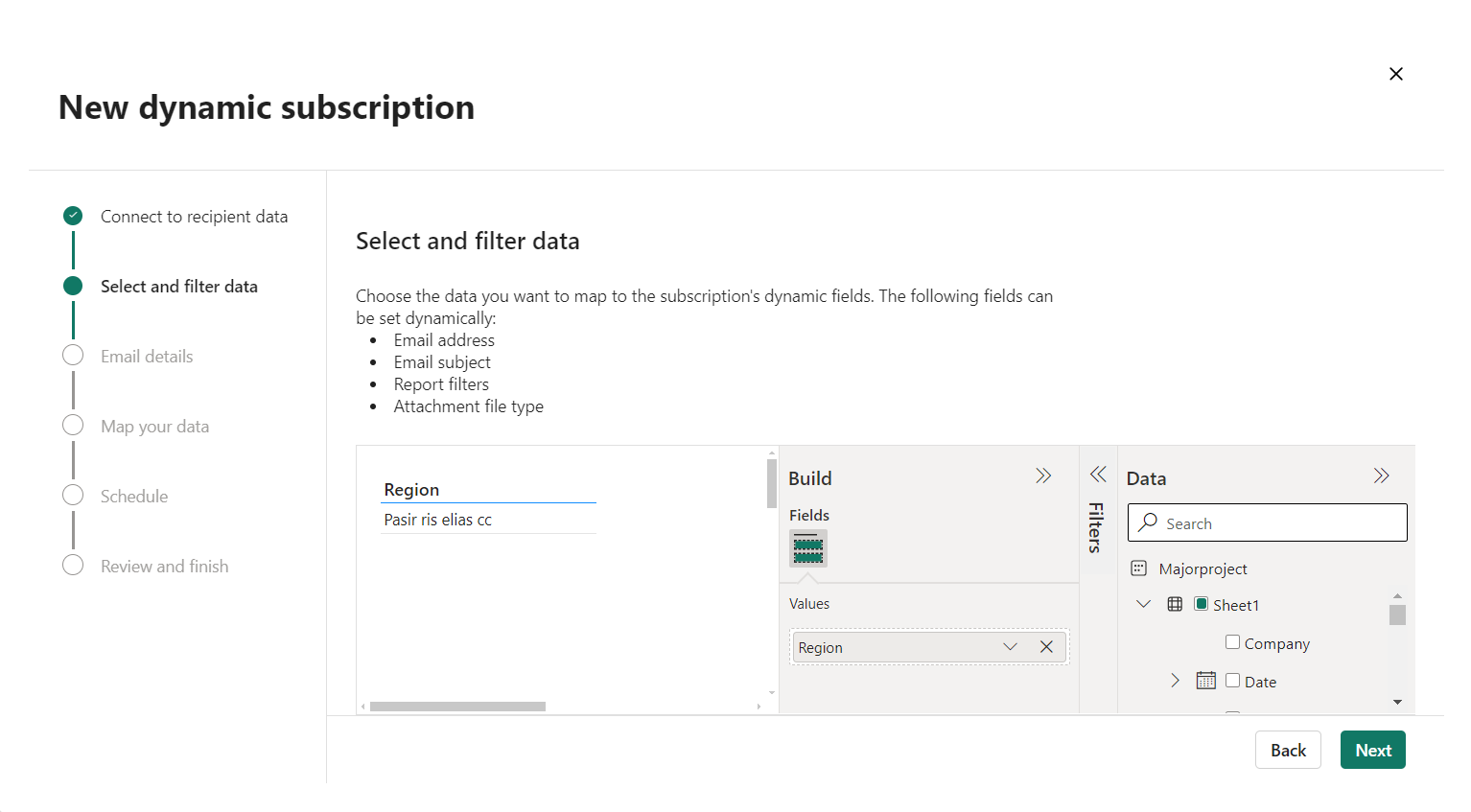
Benefit:

* Dynamic filtering restricts sensitive data access to each recipient's region, ensuring no unauthorized personnel gain access to data they are not permitted to view.
* Pre-filtered reports delivered directly to users save them time spent filtering data to increase efficiency for further business documentation.
* Customizable scheduling ensures stakeholders receive reports at their preferred intervals ensuring timely updates.

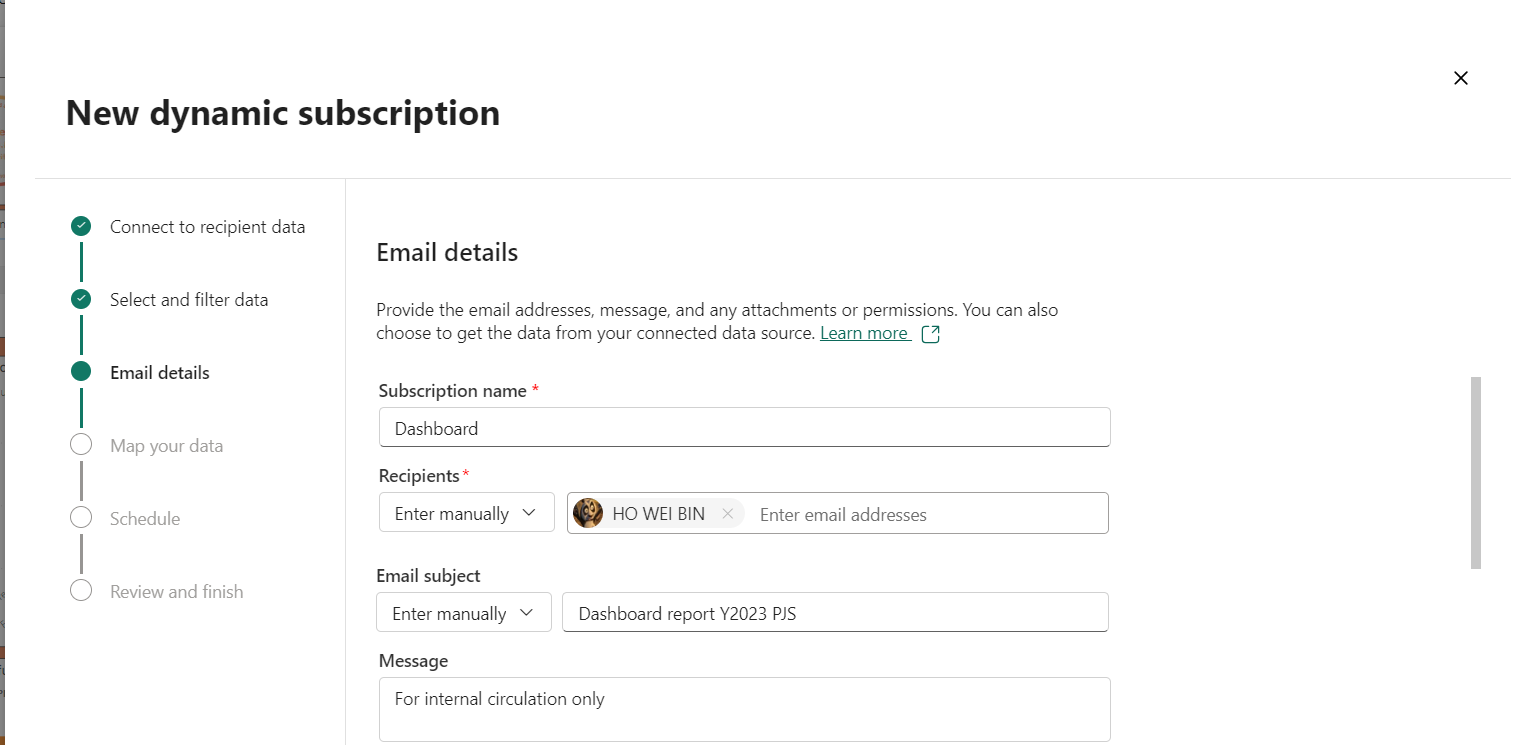
Real application: My first task during internship was to transfer a dashboard from power Bi to tableau and my supervisor had to manually email me. With a subscription system, the dashboard can be automatically sent to me without any manual performance and email storage consumption.

Configuration:

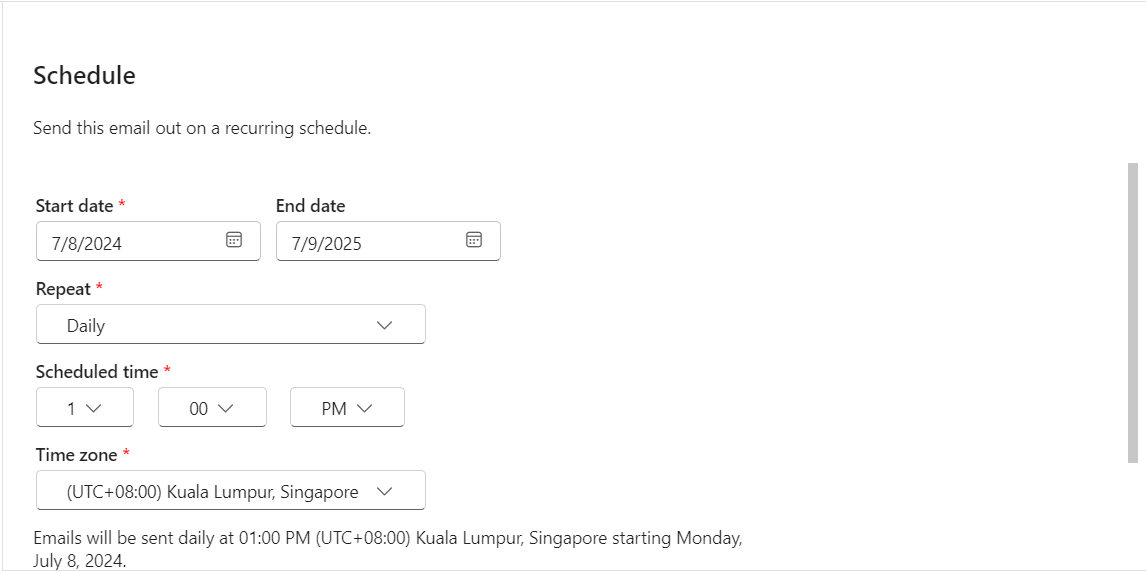
* Filter:



* Email details:



* Schedule:



(Report distribution scheduled after scheduled data refresh at 12pm for latest data)

# Chapter 6: Deployment

Web application: Originally, I demonstrated the model's effectiveness by applying it to existing data. While this approach was valid, it lacked practicality for widespread use. To improve accessibility and usability, I decide to develop a user-friendly website that eliminates technical barriers, enabling anyone to utilize the model conveniently.

Front end:

* Software used: Visual studio code
* Basic structure & content: HTML
* Overall visual appearance styling: CSS
* Responsive UI components: Bootstrap
* Minor animation: Javascript

Functionality:

* Code primarily focuses on the user interface and presentation of the web application.
* I have used a template syntax ({{prediction\_text}}) to dynamically display the predicted outcome, likely requiring additional backend code to handle form submission and prediction generation.

Back end:

* Software used: Jupyterlab

The reason behind the change from using jupter notebook to using jupyterlab is primarily because of the flexibilities and features that the platform provided. Most importantly, JupyterLab allows me to work with multiple notebooks, text files, and other documents simultaneously. Secondly, the platform has more features such as a file browser, terminal, and other tools for efficient workflow.

* Flask

Now that I had a html with CSS embedded ready, I needed to find a tool to bridge the front end of the application to the model itself which is where flask is used. It used for building web applications. It provides the essential tools and structure to handle web requests, process data, and generate responses.

Benefits:

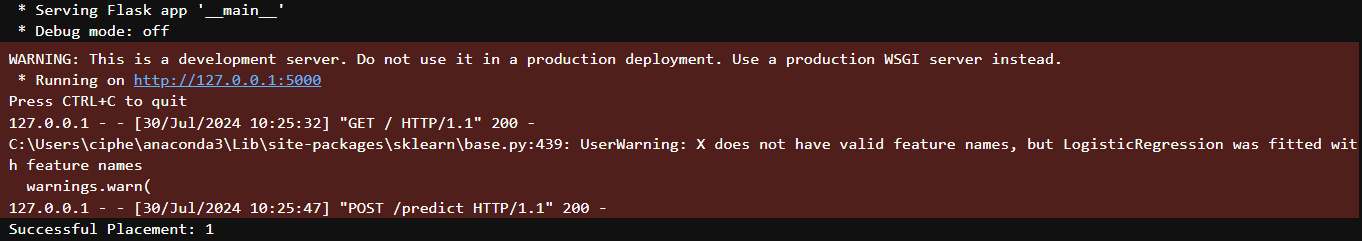
* The simple minimal core allows developers to focus on building their application without unnecessary overhead.
* Flask can also handle growing traffic and complexity by integrating additional libraries and components.

Flask role: When the user enters the form, flask receive the requests and takes the data from the HTML form and processes it according to the logic I have set in python. It accesses the model I have saved and uses it to make predictions based on the input data. Once the model has predicted, Flask takes the prediction results and sends them back to the HTML template, updating the webpage with the output.

Entire flow:

From the previous main IPYNB, save the model -> load the model into the deployment IPYNB folder -> Place the HTML file into a folder called: “templates” -> Load the app containing the HTML file using flask -> Establish the app route and render the appropriate template -> establish the core logic of the web application: Take user input, processes it, makes a prediction using your model, and then displays the result on the webpage.

Result (reflected in JupyterLab):



Result (reflected in the website):



Crucial situation to utilize the website: I realized that I have not set context where the model is useful for the organization.

1. Recruitment Strategy: Identify industries, regions, or employee types with higher placement success rates to focus recruitment efforts accordingly.
2. Talent Acquisition: Optimize the recruitment process by understanding which candidate profiles are more likely to lead to successful placements.
3. Performance Analysis: Analyze the factors contributing to successful placements to improve overall recruitment performance.
4. Market Expansion: Determine which industries and regions offer the best opportunities for business growth based on placement success rates.
5. Client Acquisition: Identify potential clients in industries with high placement success rates.
6. Resource Allocation: Allocate resources effectively by focusing on regions and industries with the highest potential for successful placements.
7. Labor Market Analysis: Understand regional disparities in employment placement to inform labor market policies and initiatives.
8. Education and Training: Identify skill gaps and training needs based on placement success rates in different industries.
9. Economic Development: Support economic development strategies by focusing on industries with high placement potential.

Reason for not uploading the model online:

* Due to the model handling sensitive data, keeping it offline can protect user privacy and comply with data protection regulations.
* By not exposing your model to the public internet, you reduce the risk of cyberattacks, hacking, or malicious interference.
* Hosting a website with a model can require significant computational resources and bandwidth. Keeping it offline can save costs. Even with free options like Netlify or Heroku, the scalibility might be an issue
* Lastly since the model is intended for internal use within an organization, there might be no need to expose it publicly.

# Chapter 6: Conclusion

## Lessons learnt and Insights gained.

Insights learned (Practical application of Data and ML):

* Understanding supply and demand imbalances between employers and potential employees helps identify needs for training programs or targeted outreach.
* Models can truly support organizations in identifying future job market trends, aiding decision-making, and resource allocation. There is an actual practical application for it.
* Working on this project made an actual demonstration of the value of data analysis in connecting residents with suitable jobs.

Technical challenges:

* Working at PA required a lot of resourcefulness, especially regarding this project's technical execution.
* The ambitious project premise relied heavily on technical skills like machine learning and data visualization, which required contextualization within the project scope.
* Navigating and troubleshooting the Power BI platform relied heavily on research and online communities for assistance since I was less experienced in it.
* Power automates itself also required a lot of researching and trial and error since it was a platform that I was foreign with.

Soft skills exposure:

* Conveying the pros and cons of machine learning application to my supervisor, who was unfamiliar with the concept, required clear and effective communication.
* Balancing an internship with a major project, especially while initially planning two separate projects, presented a significant time management challenge.
* Since this project was pitched together by me, it took a lot of confidence to ensure that the decision-making processes were optimal and valid.
* Since multiple of the platform and programmes I use were foreign to me, I had to conduct my own research and learning which really honed my independent learning skills.
* As per the marking rubrics, I had to bear in mind the novelty of my solution as well as the idea of broadening my project scope, I had to practice a lot of creativity, to create a project that exceed whatever is taught in the classroom.

## Recommendations for improvement:

1. Feed more data into the model for a more concrete and stable model.
2. Tweak the website to ensure a smoother process of prediction.
3. On the topic of my deployment, I am aware of the numerous improvement I can make to the site and my method of deployment to let my stakeholders have more convenience since my current method of deployment mean that they must have jupyterlab installed for them to be able to access the website.

* Develop a web-based application accessible only within your organization's network.
* Create a desktop application for users to interact with the model.

1. Explore more opportunity for more complex automation especially when new features are introduced.
2. Modify the backend of the and coordinate with the training process such that all naming conventions are matching to help with easy future maintenance.

References/ Bibliography

ChatGPT: OpenAI. (n.d.). ChatGPT [Large language model]. <https://chat.openai.com/>

City Population. (n.d.). Punggol (Planning Area, Singapore) - Population Statistics, Charts, Map and Location.

[<https://www.citypopulation.de/en/singapore/admin/303__punggol/>]

Gemini: Google AI. (n.d.). Gemini [Large language model]. <https://gemini.google.com/app>

Ministry of National Development Singapore [MND Singapore]. (2024, May 2). URA prepares massive site in Sengkang to house new residential estate.

The Business Times. Retrieved from <https://www.businesstimes.com.sg/property/ura-prepares-massive-site-sengkang-house-new-residential-estate>

Stack overflow: <https://stackoverflow.com>

# Appendix A1: Terms of Reference

Temasek Polytechnic

School of Informatics & IT

**Diploma in Big data & analytics**

AY2023/2024 first Semester Level 3

MP Terms of Reference

**Project Particulars**

|  |  |
| --- | --- |
| **MP Supervisor** | Mr Kok Yau LIM |
| **Project Title** | Project S.U.C.C.E.S.S complete analysis |
| **Student Matric Card Number** | 2202561C |
| **Student Name** | Ho Wei Bin |

**1. Introduction**

Project (SINGAPOREANS UNITED AS A COMMUNITY TO CARE AND TO ENCOURAGE SELF- SUFFICIENCY) S.U.C.C.E.S.S is a nonprofit organisation supported by Northeast CDC that perform jobs matching to candidates. This project will provide a complete picture on the interview outcomes held by project success on behalf of the companies.

**2. Objectives of the Project**

Main objective: Provide a full picture on the preference that the residents have on the different industries in the form of dashboards and charts.

Secondary objective:

* Utilize machine learning algorithm & technique to predict future applicants’ preference.
* Provide reasoning & speculation (with contextual understanding) as to preference from different region regarding their choice in industry.
* Deploy the model into a website for a more practical and easy access on the part of the end users.

**3. Scope of the Project**

* Tidy and organize data structure to ensure smooth ETL process.
* Create preliminary dashboard for base report on findings.
* Develop ML models that will assist in prediction for future job seekers.
* Incorporate ML results into existing data and create final dashboard to reflect ML effectiveness and finalize existing data findings.
* Conducting testing to validate the functionality and accuracy of developed models and dashboards.
* Create a frontend for the model and utilize flask to link the best model into the backend of the website.
* Documenting the project processes, methodologies, and outcomes for future reference and knowledge transfer.

**4. Project Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task no | Description | Start Date | End Date | Duration |
| 1 | Background & data understanding. Data cleaning | 13/5 | 17/5 | 6 days |
| 2 | Data transformation. Creation of preliminary dashboard draft & discussion with supervisor | 20/5 | 25/5 | 6 days |
| 3 | Finalize preliminary dashboard & model exploration | 3/6 | 10/6 | 8 days |
| 4 | EDA + Modelling | 12/6 | 21/6 | 10 days |
| 5 | Finalize modelling & incorporation into final dashboard | 24/6 | 3/7 | 10 days |
| 6 | Automation induction | 8/7 | 14/7 | 7 days |
| 7 | Testing + final troubleshooting | 15/7 | 21/7 | 7 days |
| 8 | Deployment | 22/7 | 27/7 | 6 days |
| 9 | Documentation | 29/7 | 7/8 | 10 days |

# Appendix A2: Weekly Progress Report

Week 1(13/5-17/5, 20/5)

Plan: Understand the standard operating procedure of the company where the dataset is from (Project Success). Evaluate and inspect the data’s definition. Derive a cohesive and effective data dictionary. Perform data cleaning.

Perform: The first 2 of the projects involves a deeper analysis of Project Success and how the organization operate, particularly how interviews are held between its employers and the residents. I read up on different procedures that were taken during the interview and had a conclusive understanding of the entire process. The next 2 days revolve around coming to a complete understanding on the content and definition of the dataset. There were some definitions that I was initially unclear off such as PMET, rank & file and difference between the crucial columns: selected and successfully placed. Once that was completed, I could come up with a proper and accurate data dictionary indicating the different definition of each column. The last part of the week involves doing essential data cleaning, ensuring that missing values, inconsistencies, outliers, and duplicates were absent from the dataset.

Monitor: In terms of timeliness, the first week went quite well. There were no unexpected hold ups or delays. Originally, I was hoping to handle data transformation and data wrangling to be completed by the first week as well. However, it was alright that I did not manage to get that far but it was still alright.

Reflect: Between networking with people outside of PA, which are the project Success employees as well as having more interaction with my supervisor regarding the dataset, it is quite refreshing to be able to bridge the concepts taught in the classroom with a real context. Aside from the contextual exposure, it is also interesting to learn the jargons and industrial practices which are put in place in the process of employment.