



**COSC2667 Data Science Postgraduate
Project**

Benefi: Predicting Employment

Master of Data Science

**Xian Jing Wong
(s3772149)**

**Juhal Siby
(s3792902)**

**DEPARTMENT OF SCIENCE
ROYAL MELBOURNE INSTITUTE OF TECHNOLOGY
MELBOURNE - 3000, AUSTRALIA**

ABSTRACT

Benefi is a new kind of group benefit that improves the financial wellness of its employees. For the loan system, Benefi hopes to create a risk prediction model that could assist them to decide the risk of certain loan applicants. The aim of this project is to predict the job tenure of the specified loan applicant and it will help Benefi to evaluate the loan application term with the predicted job tenure of applicant. To develop the prediction learning model, 12 different machine learning algorithms model have been fitted with the dataset given to predict the job tenure. As the best result, the XGBoost regression machine learning model obtained the best r squared value of **0.774** on the evaluation of testing set. Other than that, some interesting findings and insights have been explored to study the relationship between other features and the job tenure of individuals by plotting different kinds of exploration graph. To visualize the prediction model, the machine learning model has been deployed with the front end webpage application for client usage. Client could directly obtain the predicted job tenure period by inputting the applicant's basic details on the front end webpage.

Keywords: Predicting Employment, Benefi, Data Cleaning, Data Engineering, Multivariate Imputation by Chained Equations(MICE), Random Forest, XGB Regressor, Data Modelling, Correlation, Imputation, Data Visualization.

Contents

List of Figures	4
1 Introduction	6
1.1 Partner/client	6
1.2 Project Background, definitions	7
1.3 Project Objectives	7
2 Preliminary Work	8
2.1 Multivariate Imputation by Chained Equations(MICE)	8
2.2 Random Forest	9
2.3 Extreme Gradient Boosting(XGBoost)	9
2.4 Extra Tree Regression	10
3 Methodology	12
3.1 Dataset Features and Metrics	12
3.2 Data Preparation	13
3.2.1 Drop Unnecessary Features	13
3.2.2 Match Job Field Category	14
3.2.3 Missing Value Imputation	15

3.2.4	Data Encoding	16
3.3	Data Modelling	18
3.3.1	Base Model Training	19
3.3.2	Hyperparameter Tuning	20
4	Results, Evaluation & Analysis	21
4.1	Best Machine Learning Model Evaluation	21
4.2	Proposed Solution	22
4.2.1	Exploratory Data Analysis	22
4.2.2	Front End Webpage Prototype	26
5	Conclusion and Future Work	28
5.1	Conclusion	28
5.2	Future work	29
6	Roles and Responsibilities	30
6.1	Project Management Plan	31
6.2	Project Management Tool	32
7	Self Reflection	34
7.1	Xian Jing Wong	34
7.2	Juhal Siby	35
	Bibliography	37

List of Figures

1.0	Null Values Count Existing in Experience 6 - Experience 9 . . .	14
2.0	Unmatched Job Field Category in Experience 1 - 5 Before Matching	14
3.0	Job Field Category Manual Match in Experience 1 - 5	15
4.0	Unmatched Job Field Category in Experience 1 - 5 After Matching	15
5.0	Missing Value Count	16
6.0	Correlation Heat Map of Features Considered	17
7.0	Data Encoding for Hierarchy Features	18
8.0	Encode Categorical Variable into Dummy Variable	18
9.0	Comparison Plot of Test Set R Squared Value among Different Models	20
10.0	Test Set R squared score for XGBRegressor model	21
11.0	Scatter Plot of Exact Value versus Predicted Value (XGBRe- gressor)	22
12.0	Top 8 Job Field With Longest Staying Period in Their Job Role	23
13.0	Top 8 Job Field With Shortest Staying Period in Their Job Role	24

14.0	Top 6 Type of Degree With Longest Staying Period in Their Job Role	25
15.0	Job Role Hierarchy Ranking With Staying Period in Their Job Role	26
16.0	Front End Webpage Application Prototype	27
17.0	Microsoft Teams Task Planner Example	33

Chapter 1

Introduction

1.1 Partner/client

Benefi, a new kind of group benefit that improves the financial wellness of employees. Benefi aims to help employees keep more of their money, escape debts, and improve their financial status. Other than that, Benefi believes that employees who have well financial status are better employees.[1] By providing low-interest loans and financial guidance to the employee, the employer could have financially empowered employees. The company's main concept is in its name, Benefi, which stands for 'Beneficial' and 'Financial'.

Titash Neogim, CTO from Benefi has supervised us in this project. He has decade of experiences in engineering management and product architecture. Other than that, Titash has strong background and knowledges in REST APIs and cloud infrastructures.

1.2 Project Background, definitions

For the loan system to the employee, Benefi hopes to create a risk prediction model that helps them decide the risk of a specific borrower by defining different loan application conditions such as rate offered, maximum loan term, and maximum loan amounts.[1] To determine the appropriate loan amount and loan term for applicants, Benefi has requested our help to create a predictive model. Benefi is interested in the average tenure of employees, the company's financial position, and the company's reputation.

1.3 Project Objectives

This project aims to predict the length of time that the specified loan applicant will most likely stay in his/her current place of employment with strong prediction accuracy on new applicants. We create a machine learning model to predict the job tenure of the specified loan applicant and help to evaluate the loan application term with the predicted job tenure of an applicant. **Other than that, we aim to find specific features which may impact the employability and valuable insight by plotting out different kinds of exploration plot.**

In addition, we have been informed that the best r squared value obtained from their current prediction model designed previously is **0.62**. Therefore, develop a machine learning model that could obtain a higher r squared value will be our further objective.

Chapter 2

Preliminary Work

The below sections gives a brief explanation on the main algorithms that we have applied during the data preparation and modelling stage.

2.1 Multivariate Imputation by Chained Equations(MICE)

In this project, we have used the Multivariate Imputation by Chained Equations(MICE) algorithm to fill in missing values. The main advantage of MICE compared to other standard methods on filling values is that it considers the correlation of variables and makes imputations for each missing variable.

To impute missing value, the Predictive Mean Matching (PMM) technique was used in MICE. What is Predictive Mean Matching (PMM)? It is a regression model with infinite and finite-dimensional components. It fills values by building a small subset of observations where the outcome variable

matches the outcome of the observations with missing values.

The MICE algorithm can be used to fill missing values for all types of variables, including continuous, binary, and categorical data. [2]

2.2 Random Forest

The Random Forest is an extended or particular type of decision tree algorithm that can be used for classification as well as regression problems. It is a supervised learning model that uses multiple learning algorithms. A number of the decision trees are created in the random forest model. As they are not correlated to each other, it helps to give better prediction accuracy than the standard decision tree algorithm. This technique is called bagging, and it helps in reducing the overfitting issue.

The trees that are built separately in the random forest are combined. The most common value among the output from trees is taken as the result for classification problems, and the mean of all the separate outputs is taken for regression problems. In general, it could be said that the final output is obtained by combining the aggregated results of the decision trees.[3]

2.3 Extreme Gradient Boosting(XGBoost)

Gradient Boosting reduces prediction errors in classification regression problems by building decision tree models one after the other to rectify the errors

made by the previous trees. This algorithm is also known as an ensemble boosting Machine Learning algorithm. It acts a similar concept to the working of a neural network. Observations will be weighted equally, and the training step was carried out in a sequential manner. After every round of training, the weight will be redefined, and the training step will be carried out again by the new weight.

Extreme Gradient Boosting or XGBoost is one of the efficient applications of the Gradient Boosting algorithm. We developed XGBoost Regressor in our project because of its execution speed and improved model performance compared to other algorithms. We found that it is highly effective and gave better accuracy compared to other regression algorithms.[4]

2.4 Extra Tree Regression

The extra tree algorithm is an extension or advanced version of the standard decision tree, which uses multiple learning algorithms in modelling that allows better prediction accuracy.

As mentioned earlier, the extra tree is similar to the Random Forest algorithm. The extra tree algorithm can also be applied for classification as well as regression problems. It is also a supervised learning algorithm. Several decision trees are created in this model and they are not at all correlated to each other, which helps to give better prediction accuracy than the standard decision tree algorithm.

The extra tree algorithm uses the bootstrap aggregation technique, also called the bagging method, to improve prediction accuracy and reduce bias. The significant difference between Random Forest and Extra Tree algorithms are:

- i. As in the case of random forest that uses sub-samples from the original one, the other uses the whole original sample that is given.
- ii. They both differ in the way on how they make the cut points for splitting the trees. The random forest algorithm choose the most appropriate one from the samples whereas the other choose it randomly.

As a result, the extra Trees algorithm is a lot faster than the random forest as they randomly choose the cut points. This action also helps to reduce the bias and variance while modelling.[5]

Chapter 3

Methodology

The detailed description of all these methods are described in the following sections.

3.1 Dataset Features and Metrics

For the given dataset from the client, a range of features were crawled from LinkedIn by the client's previous student team. They have crawled the information of individuals who come from various job fields in Canada. The dataset consists of 4996 rows of data with 107 features. The basic information of individual profiles have been included, such as job experiences with the respective starting date and ending date, educational history, number of connections, Etc. Other than that, several supporting features like social and economic factors were added into the dataset such as unemployment rate[6] and average job tenure period by different kinds of job field. Table 1.0 below summarises the attributes considered together with the metrics.

Attribute	Metric
LinkedIn Basic Information	Data retrieved from LinkedIn website and limited to individual profiles from Canada. (Working experience (1-5), educational history (1-3), job position (associate, senior, manager, president), length of period (job experience and educational history), etc.)
Unemployment Rate In Canada for Aged 15-64	Data studied by OECD[6] and obtained from Economic Research Website.
Average job Tenure period per different fields of job	Data obtained from Canada Government Website.[7]
Average attrition rate per different fields of job	Data obtained from Canada Government Website.[7]

Table 1.0: Variables investigated and their metrics

3.2 Data Preparation

3.2.1 Drop Unnecessary Features

As the dataset was extracted from the website, different people could have different quantities of past job experience. During the data crawling phase, Experience 1 to Experience 9 were recorded. However, we found out that most rows have null values in their Experience 6 to 9 columns because people don't change their job so frequently in general. The value count of null values in Experience 6 to 9 columns is shown in Figure 1.0 below. Therefore, Experience 6 to 9 columns were dropped due to the high quantity of null value existence.

```
In [20]: for i in range(6,10):
          null = df[f'Experience_{i}'].isnull().sum()
          print(f"Null Value in Experience {i}: "+null.astype(str))

Null Value in Experience 6: 2279
Null Value in Experience 7: 2925
Null Value in Experience 8: 3479
Null Value in Experience 9: 3961
```

Figure 1.0: Null Values Count Existing in Experience 6 - Experience 9

3.2.2 Match Job Field Category

Experience columns that were directly crawled from online contain the long text of occupation description such as 'it recruitment - account manager.' It is unable to be used now during the modeling phase without any processing. Therefore, the previous team had created new columns (Experience 1Matched - Experience 5Matched) with their respective job field category (IT, Human Resources, Etc.) by matching the experience content with the online job field directory. From the observation, we found out that many rows of the Experience 1-5 job field have been mismatched and categorized as 'unmatched' (refer to Figure 2.0).

```
In [34]: for i in range(1,6):
          unmatched = df.loc[df[f'Experience_{i}Matched']=='unmatched'].shape[0]
          print(f"Unmatched job field in Experience {i}: "+str(unmatched))

Unmatched job field in Experience 1: 582
Unmatched job field in Experience 2: 783
Unmatched job field in Experience 3: 801
Unmatched job field in Experience 4: 761
Unmatched job field in Experience 5: 678
```

Figure 2.0: Unmatched Job Field Category in Experience 1 - 5 Before Matching

Therefore, manual data imputation was carried out to match more mismatched job fields with the correct job field in the newly created columns (refer to Figure 3.0). This step will help us obtain a better dataset quality

and have a better prediction performance later on. As a result, the unmatched value for Experience 1 to 5 has been successfully deducted (refer to Figure 4.0).

```
In [37]: for i in range(1,6):
df.loc[df[f'Experience_{i}'].str.contains('entrepreneur') & (df[f'Experience_{i}Matched']=='unmatched'),
        f'Experience_{i}Matched'] = 'entrepreneur'

In [38]: for i in range(1,6):
df.loc[df[f'Experience_{i}'].str.contains('ai\s') &
        (df[f'Experience_{i}Matched']=='unmatched'), f'Experience_{i}Matched'] = 'it&network_administration'
```

Figure 3.0: Job Field Category Manual Match in Experience 1 - 5

```
In [42]: for i in range(1,6):
unmatched = df.loc[df[f'Experience_{i}Matched']=='unmatched'].shape[0]
print(f"Unmatched job field in Experience {i}: "+str(unmatched))

Unmatched job field in Experience 1: 274
Unmatched job field in Experience 2: 423
Unmatched job field in Experience 3: 417
Unmatched job field in Experience 4: 360
Unmatched job field in Experience 5: 392
```

Figure 4.0: Unmatched Job Field Category in Experience 1 - 5 After Matching

3.2.3 Missing Value Imputation

We continued the null value check on the remaining columns of the dataset. However, there is still much missing value in almost every feature (refer to Figure 5.0). Those column values could be significantly different to each user, such as length of duration for job experience, length of duration for educational history, Etc.

To handle multiple null values existing in the dataset, we chose to impute the missing value with a suitable value instead of dropping them. However,


```

In [62]: for i in range(0,df.shape[1]):
          null = df.iloc[:,i].isnull().sum()
          column = df.columns[i]
          if null != 0:
              print(f"Null Value {(column)}: {null}")

Null Value (latest_industry): 1
Null Value (number_of_connections): 2
Null Value (Experience_1): 1
Null Value (Organization_1): 5
Null Value (degree_1): 224
Null Value (start_1): 224
Null Value (end_1): 224
Null Value (Organization_2): 2
Null Value (Period_2Start): 20
Null Value (Period_2End): 108
Null Value (Period_2Duration): 108
Null Value (Period_2StartUnemployment): 20
Null Value (Period_2EndUnemployment): 108
Null Value (Experience_3): 302
Null Value (Organization_3): 302
Null Value (Period_3Start): 325
Null Value (Period_3End): 406
Null Value (Period_3Duration): 406
Null Value (Period_3StartUnemployment): 325
Null Value (Period_3EndUnemployment): 406
Null Value (Experience_4): 791
Null Value (Organization_4): 791
Null Value (Period_4Start): 815
Null Value (Period_4End): 876
Null Value (Period_4Duration): 876
Null Value (Period_4StartUnemployment): 815
Null Value (Period_4EndUnemployment): 876
Null Value (Experience_5): 1493
Null Value (Organization_5): 1493
Null Value (Period_5Start): 1524
Null Value (Period_5End): 1638
Null Value (Period_5Duration): 1638
Null Value (Period_5StartUnemployment): 1524
Null Value (Period_5EndUnemployment): 1638

```

Figure 5.0: Missing Value Count

we did not impute the missing value by the mean/median value as most of the columns are highly correlated with each other (refer to Figure 6.0). We chose to use the Multivariate Imputation by Chained Equation (MICE) [6] method to impute the missing value. This method will impute the missing value by considering multiple columns instead of the single column value.

In order to explore how other features affect job tenure, data exploration will be carried out in the current stage. However, valuable findings will be discussed later on in Chapter 5.

3.2.4 Data Encoding

Before the dataset could be fitted with any machine learning model, data transformation must be applied to the dataset. There are two parts of the

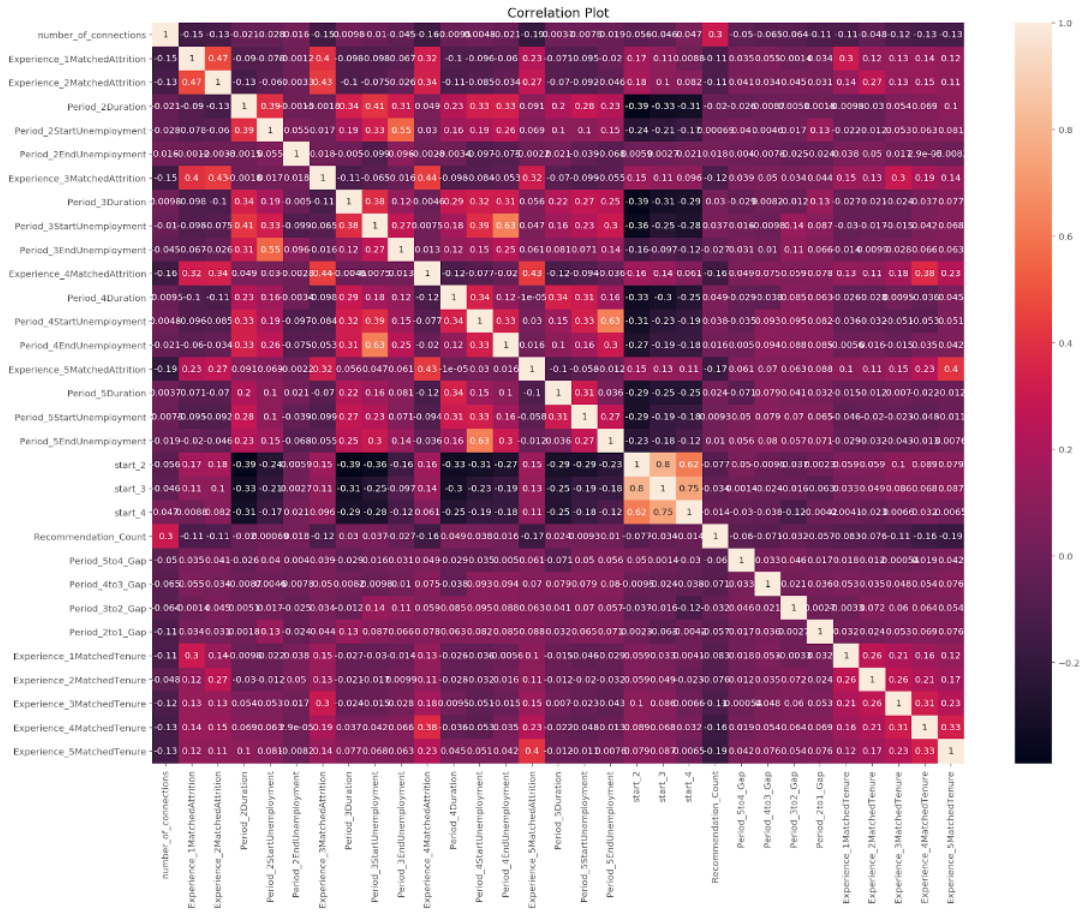


Figure 6.0: Correlation Heat Map of Features Considered

encoding process that we have applied.

Firstly, the hierarchy columns of Experience 1 - 5 were converted into ordinal variables by encoding hierarchy into four different levels (refer to Figure 7.0) depending on their role(associate, manager, chief/chairman, and president/vice president). The level of the hierarchy of role will be ordered numerically in the ascending order their level of job role from 1 to 4.

```

In [18]: print('Ordinally encoding columns')
tic = time.perf_counter()
replacementDict = {'1': 'associate',
                  '2': 'manager|senior|supervisor',
                  '3': 'chief'|'chairman',
                  '4': 'president|vice president|founder|director|partner'})
for i in range(1,6):
    for level, jobs in replacementDict.items():
        df['Experience_{i}Hierarchy'] = df['Experience_{i}Hierarchy'].str.replace(jobs, level, regex=True)
        df['Experience_{i}Hierarchy'] = df['Experience_{i}Hierarchy'].astype(float)
toc = time.perf_counter()
print('Columns ordinally encoded. (Took {toc - tic:0.4f} seconds)')

Ordinally encoding columns
Columns ordinally encoded. (Took 0.0933 seconds)

```

Figure 7.0: Data Encoding for Hierarchy Features

Secondly, we applied one hot encoder on all categorical variables such as Experience 1-5 Matched and degree 1-5 matched.[8] This encoder will convert all these categorical variables into dummy variables.

```

In [22]: print('Converting to dummies')
tic = time.perf_counter()
dummyDF = pd.get_dummies(df)
xColumns = dummyDF.columns.tolist()

```

Figure 8.0: Encode Categorical Variable into Dummy Variable

Until this stage, the dataset was well structured and it is ready to be fitted into any machine learning model.

3.3 Data Modelling

In this project, we aim to predict the staying period of an individual that will remain in his/her current job role. Thus, the target feature that we predict is the length of duration for the second latest job experience, which is the closest ended job experience (Period_2Duration). The r squared value will be the testing metric to evaluate the model performance. R squared score represents how well or close that the data points to the regression line. This result could directly give us a straightforward justification on how good our

model on predicting the job tenure.

3.3.1 Base Model Training

In the modelling stage, we constructed the dataset by splitting 80 % of the dataset as the train set and 20% of the dataset as the test set. The model will be trained by the train set, and the test set will evaluate it. Both the train set and the test set were standardized by applying the standard scaler.

For the modelling stage, 12 different algorithms of machine learning model which are Ridge Regression Model[9], Lasso Regression Model[10], Elastic Net Model[11], Random Forest Regression Model[3], Extra Trees Regression Model[5], Bagging Regression Model[12], Huber Regression Model[13], Bayesian Ridge Regression Model[14], XGB Regression Model[4], Decision Tree Regression Model[15], K Neighbors Regression Model[16] and Gradient Boosting Regression Model[17] have been fitted with the dataset by constructing the machine learning pipeline function.[18] The comparison plot below shows the r squared value obtained on the evaluation of test set for each machine learning model mentioned above. (refer to Figure 9.0)

From Figure 8.0 above, it can be observed that three models have similar r squared values on the evaluation of the test set, which are RandomForestRegressor, ExtraTreesRegressor and XGBRegressor. Therefore, hyperparameter fine-tuning will be applied to these three models to obtain better prediction performance.

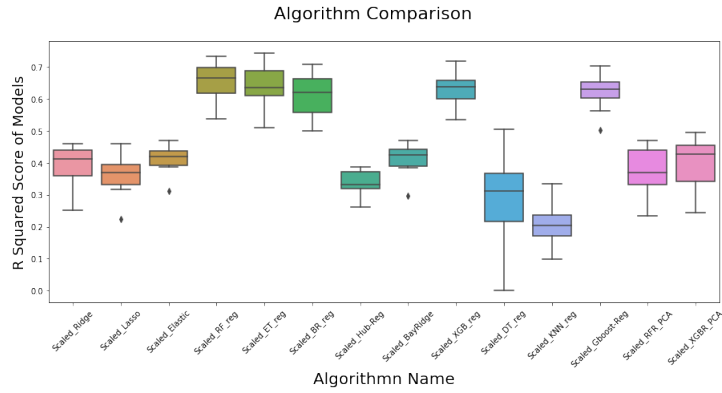


Figure 9.0: Comparison Plot of Test Set R Squared Value among Different Models

3.3.2 Hyperparameter Tuning

In this project, we have used the random search tuner[19] to apply hyperparameter tuning for RandomForestRegressor, ExtraTreesRegressor and XGBRegressor. To justify the tuning performance, we will continue to use the r squared value as the testing metric to evaluate parameter tuning. It could help us know how close is the unseen data fitted with the predicted regression line. It could help us to explore the prediction performance of the model after applying the hyperparameter tuning.

Chapter 4

Results, Evaluation & Analysis

4.1 Best Machine Learning Model Evaluation

The best-tuned machine learning model with the highest r squared value is **XGB regression model**. The r squared value on the test set obtained is 0.774 (refer to Figure 10.0), followed by 0.761 for the Extra Trees regression model and 0.756 for the Random Forest regression model.

```
In [104]: print('Training model XGBRegressor with Best Parameter')
best_model_XGB = xgb_grid.best_estimator_
best_model_XGB.fit(x_train, y_train)
score = best_model_XGB.score(x_test, y_test)
toc = time.perf_counter()
print(f'Test Score: {score}. (Took {toc - tic:0.4f} seconds)')

Training model XGBRegressor with Best Parameter

/Users/jack/opt/anaconda3/lib/python3.7/site-packages/xgboost/co
nd will be removed in a future version
if getattr(data, 'base', None) is not None and \

Test Score: 0.7739306999451394. (Took 6026.8878 seconds)
```

Figure 10.0: Test Set R squared score for XGBRegressor model

The plot of comparison between the exact value and the predicted value is shown below (refer to Figure 11.0). We can see that the data point is fitted

well with the reference line, especially for the short staying period.

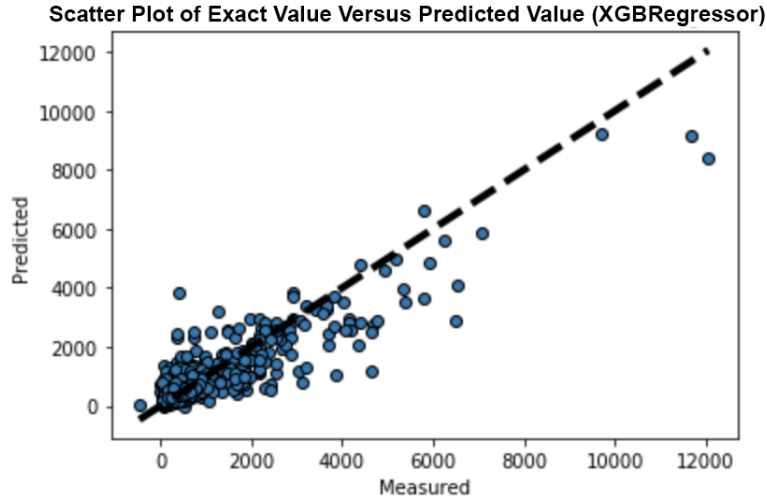


Figure 11.0: Scatter Plot of Exact Value versus Predicted Value (XGBRegressor)

4.2 Proposed Solution

4.2.1 Exploratory Data Analysis

We have plotted different types of exploration graphs to study how other features of individuals will affect job tenure. It could help us gain a deeper understanding of our project. The following graphs were based on the individual's second latest job experience (target variable) to obtain a more accurate study.

Average Job Tenure by Job Field Category

Figure 11.0 below demonstrates the top 8 job fields with the longest average period for employees to stay in their job roles. Employees in the therapy job field tend to remain the longest period in their job role comparing with other job fields from the plotted figure (refer to Figure 12.0). On the other hand, we have also plotted the top 8 job fields with the shortest average period to stay in their job role. Based on observation of the graph, we found out that health care administration employee will have the shortest period to remain in their job role (refer to Figure 13.0). These findings could provide a basic concept for the company when they received the loan application based on the job field.

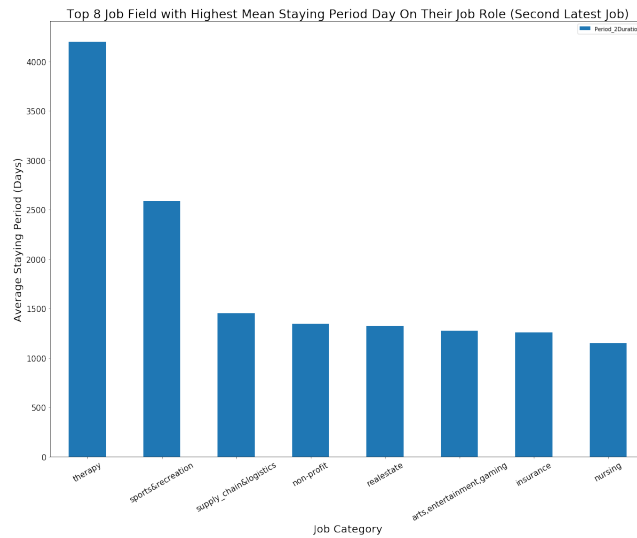


Figure 12.0: Top 8 Job Field With Longest Staying Period in Their Job Role

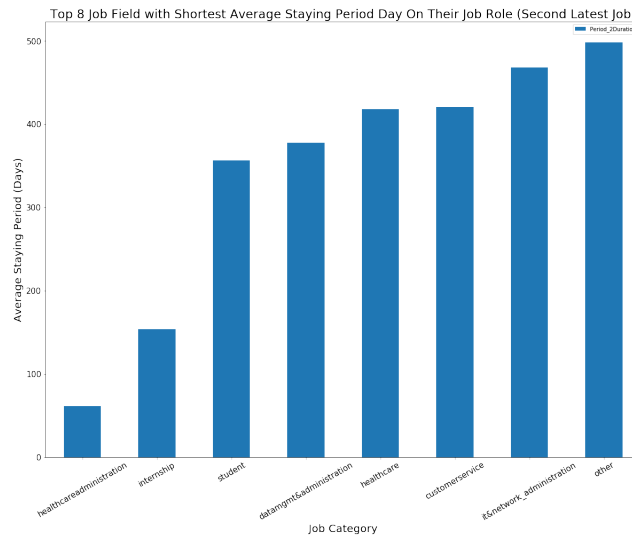


Figure 13.0: Top 8 Job Field With Shortest Staying Period in Their Job Role

Average Job Tenure by Latest Educational History

Next, we are interested in studying how different educational history levels that an individual held will affect the length of period to stay in their job role. Therefore, Figure 13.0 below shows the top 6 types of latest degree that the individual hold with the longest period that they will stay in their current job role. We can observe that the 'master of research' degree holder tends to keep up the longest period in their job role compared with other degree holders (refer to Figure 14.0).

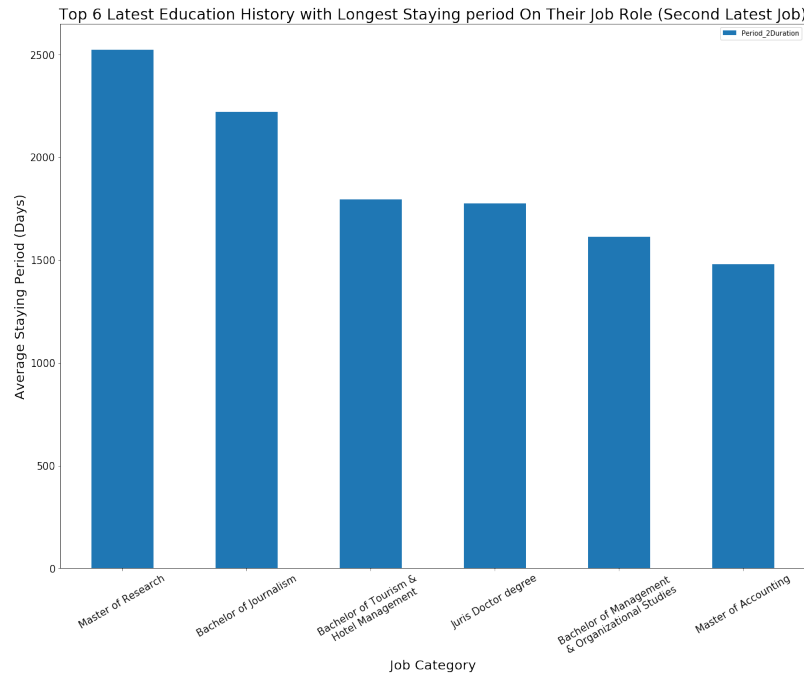


Figure 14.0: Top 6 Type of Degree With Longest Staying Period in Their Job Role

Average Job Tenure by Job Hierarchy

Lastly, we have also studied how job tenure will be affected by the job hierarchy. As mentioned above in chapter 4.2.4, we have encoded all job roles into four different levels. In the figure below, the average period that an individual will stay in their job role for different hierarchy levels. Based on the plotted graph, it can be concluded that individual tends to stay longer in their job role when they have a higher level of job role hierarchy in their company (refer to Figure 15.0).

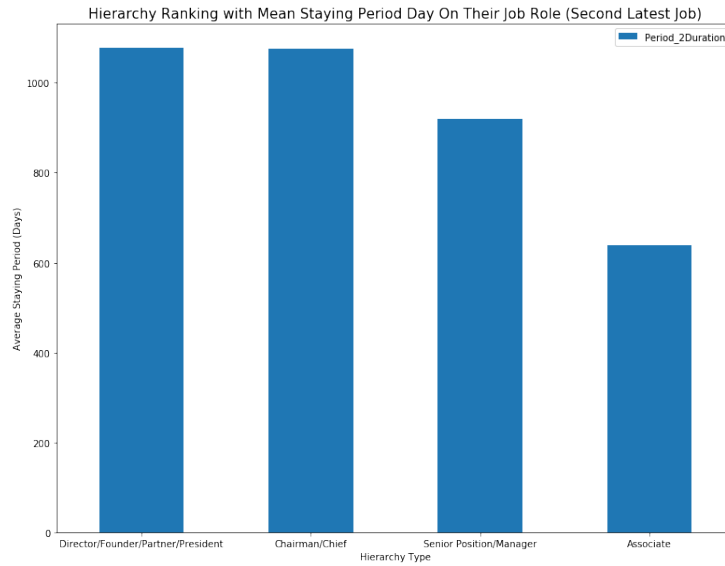
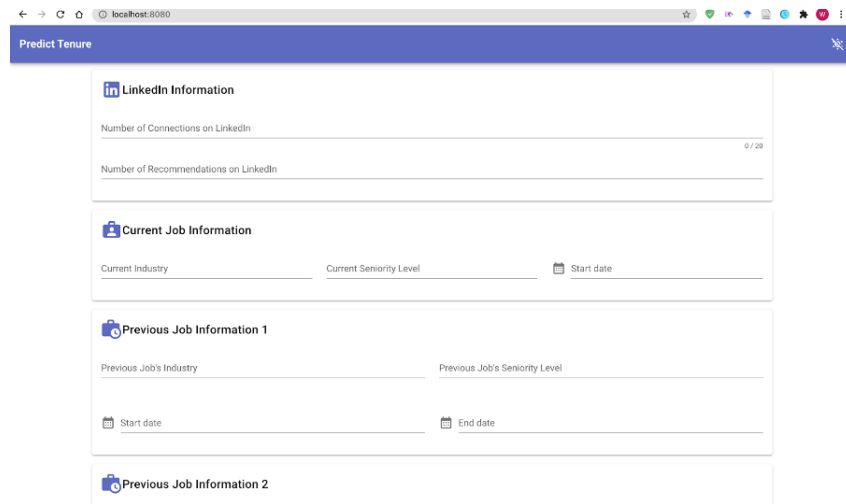


Figure 15.0: Job Role Hierarchy Ranking With Staying Period in Their Job Role

4.2.2 Front End Webpage Prototype

The machine learning model developed has been deployed in the backend with the front-end webpage application. The client could have a straightforward control panel for predicting the job tenure of the applicant. By inputting all the applicant's details, including their past job experience and educational history, the client could obtain the predicted job tenure of the applicant by just clicking the submit button. This application will be much convenient and user-friendly to all types of users. The model could be retrained anytime when we have obtained more training datasets. The figure below shows the basic design of the front-end webpage prototype.

As a result, the front-end webpage for predicting the job tenure and



The image shows a web browser window with the address bar displaying 'localhost:8080'. The page has a blue header with the text 'Predict Tenure' and a close button. The main content area is divided into four sections, each with a blue icon and a title:

- LinkedIn Information**: Contains two input fields. The first is labeled 'Number of Connections on LinkedIn' and the second is labeled 'Number of Recommendations on LinkedIn'. The second field has a character count '0 / 20'.
- Current Job Information**: Contains three input fields. The first is labeled 'Current Industry', the second is labeled 'Current Seniority Level', and the third is labeled 'Start date' with a calendar icon.
- Previous Job Information 1**: Contains four input fields. The first is labeled 'Previous Job's Industry', the second is labeled 'Previous Job's Seniority Level', the third is labeled 'Start date' with a calendar icon, and the fourth is labeled 'End date' with a calendar icon.
- Previous Job Information 2**: This section is partially visible at the bottom of the form.

Figure 16.0: Front End Webpage Application Prototype

several exploration plots have been developed to approach and achieve our project objectives.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In conclusion, to approach the project objectives, we have developed a machine learning model to predict how long an employee will stay in a company depending on the employee's details, including educational history and past career experiences. The best model with the highest r squared value is the XGBoost Regression model that gave a final r squared value of **0.774**. This means that we have successfully achieved our project's further objective of developing a better prediction model than the client's current machine learning model. To visualize the developed model, the model has been deployed in the back-end with a front-end web page application. The client could have a more straightforward control panel to have the prediction value by inputting the applicant's details. On the other hand, several exploration graphs have been plotted to better understand how other features will affect job tenure.

5.2 Future work

Although our objectives were successfully achieved, we believe that some improvements could be carried out to increase the prediction accuracy.

Firstly, improvements could be made to the data set by extracting more data from different kinds of data sources. We can obtain these additional informations from other job search and networking sites such as Seek, Indeed, etc.

Secondly, we suggest to include more sensitive features such as 'gender' and 'age' into the dataset features. It could help the enhance of prediction model and help to enrich dataset features with multi-category variables.

Lastly, the exploratory graphs that have been plotted beforehand could be visualized by creating an automated interactive dashboard. The client could have a more efficient way to check all the exploration graphs by obtaining all insights in a dashboard. The front-end interactive dashboard will be updated automatically whenever the back-end dataset was updated.

Chapter 6

Roles and Responsibilities

As mentioned in the initial project plan, the project has separated into four stages which are 'Partner Approach & Data understanding stage', 'Data Preparation & Data Exploration stage', 'Data modelling stage' and 'Final Project Wrap Up stage'.

Our team consists of two team members, who are Xian Jing Wong and Juhal Siby. When the project was just started, Juhal Siby has worked on the exploration of the dataset given. He worked on understanding the dataset provided and recorded down the description of each feature in the metadata file of the dataset.

After we had a basic understanding of the dataset, Xian Jing started to work on the data preparation stage, including missing value imputation, data encoding, Etc. At the same time, Juhal worked on the data exploration by studying the possible exploration graph to help us achieve the project objec-

tives. Juhai also carried out the research on the machine learning model. He researched and listed down all the machine learning models that we could apply during our modelling phase.

For the modelling stage, Xian Jing was in charge of modelling pipeline construction, and Juhai researched the hyperparameter tuning information. He also helped to build the tuner with a suitable parameter range for the top three machine learning models. After the modelling stage, both of us created the front-end webpage prototype by fixing the details of the back-end design from the provided application given template.

Lastly, the final stage was completed together by both of us. We both constructed the final report and presentation together.

6.1 Project Management Plan

The whole project was carried out in the CRISP-DM[20] with agile methodology. We regularly shared the progress among members, especially in the modelling stage. Like the example, we will frequently modify the modelling stage by adding a new machine-learning algorithm to the pipeline of the machine learning model. To carry out the project in the agile methodology, healthy communication is a must. Therefore, weekly meetings were carried out among the team, and we will send out the progress update email to the client every two weeks after our meeting. The meeting usually will last for an hour, and as promised to the client, we had spent at least 30 hours per

week working on this project.

6.2 Project Management Tool

Two major platforms helped us have excellent project management and keep track of the project plan, which are Microsoft Teams and Github. We have carried out our weekly meeting on the Microsoft Teams by scheduling the appointment earlier. We will upload all paper works into the Teams drive. We shared the latest update and had a discussion in the group chat to maintain healthy communication. To fully utilize the Microsoft Teams, Microsoft Teams Task Planner plugin was added to the group. It can help us keep on track with the detailed task weekly by creating the task and assigning it to the appropriate member based on their key skills (refer to Figure 17.0). Hence, it could also help us to make sure all jobs were distributed equally among all the members.

Other than that, all coding files for the project were pushed to the shared repository regularly. This action could ensure that all team members could get informed on time with the latest version of the code.

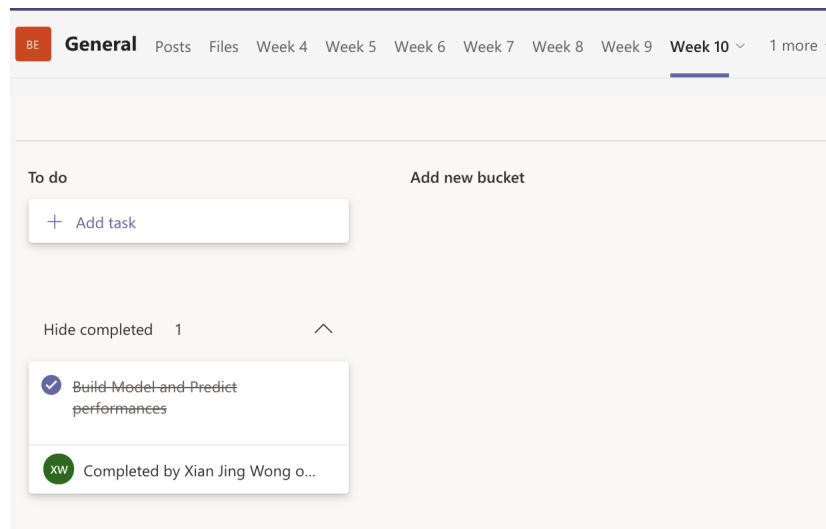


Figure 17.0: Microsoft Teams Task Planner Example

Chapter 7

Self Reflection

7.1 Xian Jing Wong

This postgraduate project is a super unforgettable and essential experience that could help me know the data field's actual career situation. Throughout the whole project, I have made plenty of mistakes, especially during the data preparation phase. The first mistake that we could discuss will be the missing value imputation issue. Without doing much extra homework on the missing value imputation issue, I have chosen to drop all rows with null values in any column. I decided to drop the rows because I thought that the data quality would be affected if impute with any value. However, after spending weeks proceeding with the dataset until the base model building stage, we realized that the accuracy is low. This experience has taught me that I shouldn't limit my mind to my current knowledge. A data scientist should always follow the latest study and journal. This could lead me to keep on track with new pieces of knowledge.

Other than that, I think there are some improvements to this project management. The communication between the client and us could be healthier and more frequent. For this project, we have chosen to update our client with our latest progress through email every two weeks. As a result, we didn't get to know the client's situation until the final stage.

In short, healthy and regular communication is the key to the success of good team management. On the other hand, I have clearly understood my lack of practical knowledge when handling the real-life dataset. I should do more profound research on the relevant task before really carrying out the job. Continuously self-directed learning will build us on becoming better at my career plan. Strong knowledge in machine learning models and mathematics will only help us to build a robust prediction model.

7.2 Juhali Siby

These past four months have been a really wonderful and eye opening experience for me. Being a data science master student, I thought I knew how to deal with large data sets and analyze them. However, I was completely wrong. I made mistakes in aspects like cleaning the data and visualizing it, which I thought was pretty simple. But these aspects had a huge importance in the project. As a result of my thorough research and study, I was able to rectify these mistakes. One of the mistakes that I made was with the Data visualization part. I thought that by simply plotting the graphs, we would get valuable insights. But to gain valuable insights, we need to see which

variables to choose and the one's that need to get evaluated. This was one of the lessons that I learned from the project. I also found how important data visualization is in a data science project. There were difficulties while making changes to the columns and modifying them as well.

As to what Xian Jing mentioned earlier, I also thought there could be improvements in the project management part. The communication between the client and us could have been a lot better, and their delays in response to our emails were not at all favorable. This actually made our work more difficult. But considering the situation they were in, it is none to be blamed.

Overall, this postgraduate project has been an excellent opportunity for me. It made me understand the difficulties a data scientist faces while dealing with real-life datasets. I am sure that this learning experience will definitely help me develop my analytical and statistical skills and help me achieve my career goals, which is ultimately to become a data scientist.

Bibliography

- [1] Benefi, “Build Financial stability in your workforce..” <https://benefi.ca/for-employers-1>.
- [2] “mice function - rdocumentation.” <https://www.rdocumentation.org/packages/mice/versions/3.13.0/topics/mice>.
- [3] A. Chakure, “Random forest regression.” <https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f>, 2019.
- [4] J. Brownlee, “Xgboost for regression.” <https://machinelearningmastery.com/xgboost-for-regression/>, 2021.
- [5] P. Aznar, “What is the difference between extra trees and random forest?.” <https://quantdare.com/what-is-the-difference-between-extra-trees-and-random-forest/>, 2020.
- [6] O. Canada, “Unemployment - unemployment rate - oecd data.” <http://data.oecd.org/unemp/unemployment-rate.html>.
- [7] G. of Canada, “Job tenure by industry, annual.” <https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1410005501>.

- [8] B. Chen, “What is one-hot encoding and how to use pandas getdummies function.” <https://towardsdatascience.com/what-is-one-hot-encoding-and-how-to-use-pandas-get-dummies>, 2020.
- [9] J. Brownlee, “How to develop ridge regression models in python.” <https://machinelearningmastery.com/ridge-regression-with-python/>, 2020.
- [10] J. Brownlee, “How to develop lasso regression models in python.” <https://machinelearningmastery.com/lasso-regression-with-python/>, 2021.
- [11] J. Brownlee, “How to develop elastic net regression models in python.” <https://machinelearningmastery.com/elastic-net-regression-in-python/>, 2020.
- [12] J. Brownlee, “How to develop a bagging ensemble with python.” <https://machinelearningmastery.com/bagging-ensemble-with-python/>, 2020.
- [13] J. Brownlee, “Robust regression for machine learning in python.” <https://machinelearningmastery.com/robust-regression-for-machine-learning-in-python/>, 2020.
- [14] DataTechNotes, “Bayesian ridge regression example in python.” <https://www.datatechnotes.com/2019/11/bayesian-ridge-regression-example-in.html>, 2019.

- [15] S. Girgin, “Decision tree regression in 6 steps with python.” <https://medium.com/pursuitnotes/decision-tree-regression-in-6-steps-with-python-1a1c5aa2ee16>, 2019.
- [16] I. Muhajir, “K-neighbors regression analysis in python.” <https://medium.com/analytics-vidhya/k-neighbors-regression-analysis-in-python-61532d56d8e4>, 2019.
- [17] J. Brownlee, “How to develop a gradient boosting machine ensemble in python.” <https://machinelearningmastery.com/gradient-boosting-machine-ensemble-in-python/>, 2020.
- [18] L. ARORA, “Build your first machine learning pipeline using scikit-learn!” <https://www.analyticsvidhya.com/blog/2020/01/build-your-first-machine-learning-pipeline-using-scikit-learn/>, 2020.
- [19] J. Bergstra and Y. Bengio, “Random Search for Hyper-Parameter Optimization,” *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, 2012.
- [20] D. S. P. Management, “What is crisp dm?.” <https://www.datascience-pm.com/crisp-dm-2/>.