Analysis on The Prediction of Homeless using Machine Learning Algorithm

by

Nur Izza Natilia binti Mohd Burhan
17003815

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Information Systems (Hons)

FYP II
September 2021

Universiti Teknologi PETRONAS
32610 Seri Iskandar
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Analysis on The Prediction of Homeless using Machine Learning Algorithm

by

Nur Izza Natilia binti Mohd Burhan

17003815

A project dissertation submitted to the

Information Systems Programme

Universiti Teknologi PETRONAS

In partial fulfilment of the requirement for the

BACHELOR OF INFORMATION SYSTEMS (Hons)

Approved by,

_____

(Ts. Ahmad Izuddin bin Zainal Abidin)

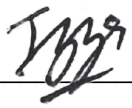
UNIVERSITI TEKNOLOGI PETRONAS

SERI ISKANDAR, PERAK

SEPTEMBER 2021

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



Nur Izza Natilia binti Mohd Burhan

ABSTRACT

The analysis shows that the categories of homeless are including individual, people in families, chronically homeless individual and veterans. Public are only aware about homeless people's welfare but not their different background. Previous research has relied on the traditional method where they collect qualitative data and lead to time consuming. We used data from the Housing and Urban Development Department (HUD) of US to analyse the number of homeless people in different urbanicity category. The increase of homeless people and their background of homelessness are clearly identified using Random Forest algorithm. Our findings indicate that the homeless came from two main environment which are sheltered and unsheltered. In this project, the Python software is used to develop the machine learning algorithm. The goal of this project is to predict the numbers of homeless people in different area including major city, suburban and rural and apply the machine learning algorithm to classify whether they are homeless or not.

ACKNOWLEDGEMENT

First of all, praise be to Allah for the opportunity in giving me time and ideas to complete this Final Year Project (FYP) report and further my analysis on the topic I have chosen.

I would like to take this time to thank everyone who contributed to and was involved in the accomplishment of my FYP. I have learned a lot of new things over the two semesters, and it has been an incredible experience for me. Throughout the trip, there have been ups and downs, but all of these experiences will aid me in becoming a better person in the future.

Furthermore, I would want to convey my heartfelt gratitude and appreciation to my supervisor, Mr. Ahmad Izuddin bin Zainal Abidin, for his unwavering assistance and support throughout my FYP. He has been assisting me in various ways to guarantee that I do not have too many commitments or issues when completing my project. He has been one of the most important factors in my completion of this FYP. He never stops providing me advise, thoughts, and suggestions, which has really aided me in generating ideas for the application of machine learning.

In addition, I would like to express my gratitude to my family, particularly my parents. They have been an incredible source of encouragement and motivation for me during the session. Their unwavering support will be remembered forever.

Last but not least, I would like to take this occasion to express my gratitude to all of my colleagues who assisted in the application of machine learning. I learned a lot from them, and their willingness to assist me made it much easier for me to finish this assignment.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

FYP: Final Year Project

HUD: Housing and Urban Development Department of US

CRISP-DM: Cross Industry Standard Process for Data Mining

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Background of Study

i. Homelessness

Homelessness is a person, family, or community's lack of secure, safe, permanent, adequate home or the near possibility of means and capacity to acquire it. It is crucial to remember that this definition does not include all forms of homelessness. Different groups of individuals are impacted differently, and each person's experience is unique. Homelessness is not just a result of housing insecurity. These distinctions are significant when evaluating strategies of treating homelessness, because one technique does not apply to every community. There are several myths and misunderstandings about homelessness. Some feel it is a choice where they believe that those who are suffering homelessness can simply pull themselves up "by the bootstraps" if they want to, and that they are homeless merely because they are lazy.

ii. Machine learning

Machine learning is an artificial intelligence discipline that focuses on building systems that can learn from data and improve their accuracy over time without being trained to do so. Machine learning today differs from machine learning in the past due to advancements in computer technology. It arose from pattern recognition and the notion that computers can learn without being taught specific tasks. Instead, artificial intelligence researchers wanted to see if computers could learn from data. Because models can adapt independently as exposed to new data, machine learning's iterative component is critical. They rely on previous computations to make consistent, repeatable decisions and outcomes. It is not a new science, but it has gained popularity recently.

iii. Classification in machine learning

Classification is a predictive modelling task that predicts a class label for a given input data sample. From the standpoint of modelling, classification necessitates a training dataset with many instances of inputs and outputs from which to learn. A model will utilize the training dataset to determine the optimal way to map instances of input data to particular class labels. As a result, the training dataset must be sufficiently representative of the issue and contain a large number of samples of each class label.

1.2 Problem Statement

Based on non-government organization (NGO) which is UBUNTU Malaysia, they stated that the rate of homelessness is increasing year by year (Kay Li, 2018). The main explanation for this might be the stagnant economic condition in recent years. Homelessness arises when the general population and government are indifferent to the plight of the homeless and prefer to overlook the problem. Moreover, there are lack of quality data on the background of homeless that lead to inaccuracy, irrelevancy, incompleteness, untimeliness, and inconsistency. Also, previous method used by some researchers was not advanced enough since they only used traditional method where they collect qualitative data by interviewing some homeless people.

1.3 Objectives

This project's major goal is to utilize machine learning algorithm that can create a classification model for homelessness. This project will help to reduce the time to collect data of homeless people. Moreover, other goals that must be met in this project are as follows:

- To predict the numbers of homeless people by using machine learning technique.
- To classify the categories of homelessness in major city, suburban and rural area.

1.4 Scope of study

This research will process on the dataset of different type of homelessness. The machine will train on 3 urbanicity category of homelessness which are major city, suburban and rural. A sample of dataset from Department of Housing and Urban Development is used for this project. The method will then predict which type of homelessness is the highest. To develop this classification algorithm, a Python software will be used to train the dataset.

1.5 Significance of Study

This project is important because we want to learn from previous study to produce reliable results. Some of previous study indicate that they only use some classification method that was hard to understand. So, we enhanced the machine learning technique more easily. Also, we want to highlight the importance of humanity issues which is homeless by not just being aware of their welfare but also recognize their various situation of homelessness.

CHAPTER 2

LITERATURE REVIEW

2.1 Homelessness

Homelessness is formally defined by the United States government as “lack of a fixed, regular, and adequate nighttime residence, and if they sleep in a shelter designated for temporary living accommodations or in places not designated for human habitation,” according to the Oxford Encyclopedia of Social Work. There are two types of homeless which are sheltered and unsheltered. Homeless individuals who are sheltered spend the night in emergency shelters or transitional, or temporary, accommodation. Homeless individuals who are not sheltered sleep on the streets, in automobiles, in abandoned houses, or in other areas not designed for human habitation.

A lesser-known fact among Malaysians is that 90 percent of the homeless population are Malaysian natives, not immigrants (Lia, 2020). Unemployment, low income, and domestic violence are the leading causes of homelessness in Malaysia. Contrary to popular belief, most of Malaysia's homeless are working-class people who cannot afford a place to live due to a lack of a consistent source of income. According to the findings of this study, the homeless underused healthcare facilities, which might be attributed to their inability to pay as well as their physical limitations (Aizuddin et al., 2019). However, this study discovered that their salary was very low, making it impossible for them to even rent a room in Kuala Lumpur.

It is not easy to obtain accurate data about homelessness. The most recent figures are from a study conducted in February 2016 by the Kuala Lumpur City Council (DBKL), which estimates the city's homeless population to be between 1,500 and 2,000 people (Irsyad, 2016). In actuality, previous Social Welfare Department (JKM) findings from a study of 1,387 homeless people in Kuala Lumpur indicated that just 4.8 percent of them took narcotics (Yani et al., 2016). Thus, individual and systemic factors are the two types of variables that influence homelessness. Individual factors are personal experiences in a person's life that may have led to their ending up

homeless. Mental health difficulties, a lack of familial support, substance misuse, a history with the criminal justice system, traumatic events and poverty are examples of these.

Furthermore, according to the findings, being homeless is an action identified by a gradual deterioration of one's resilience due to a series of unpleasant experiences in someone's life (Mabhala et al., 2016). For example, losing a significant person in someone's life and being in an abusive environment were two of the most commonly mentioned occurrences. The findings also reveal that the last stage of homelessness is a total collapse of relationships with the individuals with whom they live. The following are the most prevalent behaviours identified by participants as a primary cause of breakdown: drug addiction, drinking, self-harm, and disruptive behaviour. So it is also in legal problems, including burglary, criminal offences, arson, convictions and theft.

2.2 Existing Application of Machine Learning Method

Application of machine learning is important in making decision, improve accuracy and efficiency while eliminating the risk of human mistake. Most administrative data sets do not adequately represent homelessness, making it difficult to determine how, when, and where this group may be effectively supported (Byrne et al., 2020). A connected database was utilised in this study to collect data on over five million people in Massachusetts. The model performed well in terms of specificity, sensitivity, and classification properties. Byrne et al. (2020) stated that they also looked at the connection between model anticipated homeless status and fatal opioid overdoses and found that model predicted homeless status was associated with an approximately 23-fold increase in the likelihood of fatal opioid overdose.

Moreover, given demographic and service consumption data, the Municipal Government of London, Canada, launched an effort to develop a prediction model that properly forecasts the probability that an individual would become chronically homeless 6 months in the future (VanBerlo et al., 2020). London is one of several Canadian communities that utilise the Homeless Individuals and Families Information System (HIFIS) programme to coordinate service delivery for homeless people. In reality, it was the HIFIS database administrator from the Information Technology Services division who first questioned if the data in the HIFIS database might be used

to forecast chronic homelessness. Furthermore, the possibility of using this data in the first place was dependent on a solid data governance basis. In our research, we explored a number of different modelling approaches (VanBerlo et al., 2020). The following are the possible modelling techniques: Logistic regression, Random Forest, XGBoost, Multilayer perceptron and Hybrid recurrent neural network + multilayer perceptron. To meet the transparency requirements, an explainability method known as Local Interpretable Model-Agnostic Explanations was used (Ribeiro et al., 2016). LIME is model-independent, which means it may be used to generate explanations for any sort of black box model. LIME finds the aspects of an example that most significantly contribute to the prediction of a model. LIME returns a list of weights for each feature value. Positive weights indicate that the characteristic led to a positive prediction that is chronic homelessness, whereas negative weights indicate that the feature contributed to a negative prediction that is not chronically homeless.

Other than that, with almost 60,000 individuals now living in public shelters, New York City is dealing with an ever-increasing homeless population (Hong et al., 2018). In 2015, about 25% of families lived in a shelter for more than nine months, and 17% of families with children who left a homeless shelter returned to the system within 30 days of leaving. This means that “long-term” shelter inhabitants and those who re-enter shelters contribute significantly to the increase in the homeless population staying in municipal shelters, as well as systemic challenges in securing suitable permanent housing. This article focuses on our early work with Win (Women in Need) shelters to better understand the factors that impact homeless family readmission and stay duration. They provide a centralised database of the homeless population served by Win shelters, which includes over 6,000 homeless families. They utilise an unsupervised clustering approach and logistic regression models to discover long-term length-of-stay and factors of re-entry. Age, citizenship, medical problems, occupation, and foster care history or shelter stays as a child have all been identified as important indicators. The K-means clustering results show three primary groups corresponding to previous typologies of episodically homeless, chronically homeless and transitionally homeless. Controlling for other characteristics, the probability of younger clients being readmitted is less. However, this connection flips when age is combined with medical conditions or race (Hong et al., 2018).

2.3 Advantages of Random Forest Algorithm

A random forest is a machine learning technique for solving classification and regression problems. It makes use of ensemble learning, which is a technique for solving complicated problems by combining several classifiers. Many decision trees make up a random forest algorithm. Bagging or bootstrap aggregation are used to train the 'forest' formed by the random forest method. Bagging is a meta-algorithm that increases the accuracy of machine learning methods by grouping them together.

The random forest algorithm determines the outcome based on decision tree predictions. It forecasts by averaging or averaging the output of various trees. The precision of the result improves as the number of trees grows. A random forest method overcomes the drawbacks of a decision tree algorithm. It reduces dataset overfitting and improves precision. It generates forecasts without requiring a large number of package setups like scikit-learn (Mbaabu, 2020).

There are some advantages of being using Random Forest classifier in our project which is there is no need to scale the features. Random Forest does not require feature scaling which are standardisation and normalisation because it uses a rule-based method rather than distance calculation (Kumar, 2019).

Furthermore, Random Forest is effectively handling non-linear parameters. Unlike curve-based algorithms, non-linear parameters have no effect on the performance of a Random Forest. As a result, if the independent variables are highly nonlinear, Random Forest may outperform conventional curve-based methods.

CHAPTER 3

METHODOLOGY

3.1 CRISP-DM Methodology

Figure 1: CRISP-DM Methodology

In the year 1996, a technique used to develop data mining projects is CRISP-DM. It stands for Cross Industry Standard Process for Data Mining. A Data Mining project is conceptualised in six stages, with cycle iterations dependent on developer requests. The steps involved include Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment.

Firstly, the business understanding stage. Its objective is to offer context for the goals and data so that the engineer or developer understands the value of data in

that particular business model. It comprises document reading, specific field learning, and many ways they help the development team, such as making queries about critical context. As a result of this phase, the development team now understands the project's background. The project's goals should be established before the project begins. For example, the development team should be aware that the aim is to increase sales, and once that phase is complete, the team should understand what the client sells and how they sell it.

Secondly, the data understanding stage where it means to determines what can be expected and accomplished using the data. It considers data governance compliance, data completeness and value distributions when determining data quality. This stage is a crucial part of the project because it determines how feasible and reliable the final results will be. In this stage, we look for ways to make the information more valuable. The development team should first learn about the business and how that information benefits it when they are unsure about the data significance or usage. Because of this stage, data scientists now understand how, in terms of data, the result should fulfil the project's goals, what algorithm and process deliver that result, how the data is current, and how it should be helpful to the algorithm and process involved.

Data preparation is the third stage, including the Extract, Transform and Load (ETLs) or Extract, Load and Transform (ELTs) process, which employs algorithms and methods to transform data into something usable. Thus, data engineers and data scientists are responsible for standardising information when data governance standards are not followed or implemented in an organisation. Similarly, some algorithms perform better with specific parameters, while others refuse to accept non-numerical values and do not work well with a wide range of values. The development team, on the other hand, is responsible for standardising data.

Next, the centre of the machine learning project is modelling, which is at the fourth stage. The satisfied outcome or the project’s objective achievement contribution are performing in this stage. Although this is the most glamorous phase of the project, it is also the quickest because there is nothing to change if everything else has been done well. The technique is designed to go back to data preparation and enhance the current data if the results can be improved. This is where we choose the right modelling whether it is supervised learning, unsupervised learning and reinforcement learning.

The user must ensure that the findings are legitimate and correct in the fifth stage, evaluation. If the results are incorrect, the method allows a return to the first stage to determine the false results. On most data science projects, testing and training data were divided by the data scientist. This stage uses the testing data to ensure that the model that emerges from the modelling process is accurate. The confusion matrix is one approach to validating findings in the context of supervised learning, for example, categorising things.

In the end, the deployment stage which entails presenting the findings in a usable and understandable format to achieve the project's objectives. It is the only stage that is not part of a cycle. Depending on the ultimate user, a practical and understandable approach may differ.

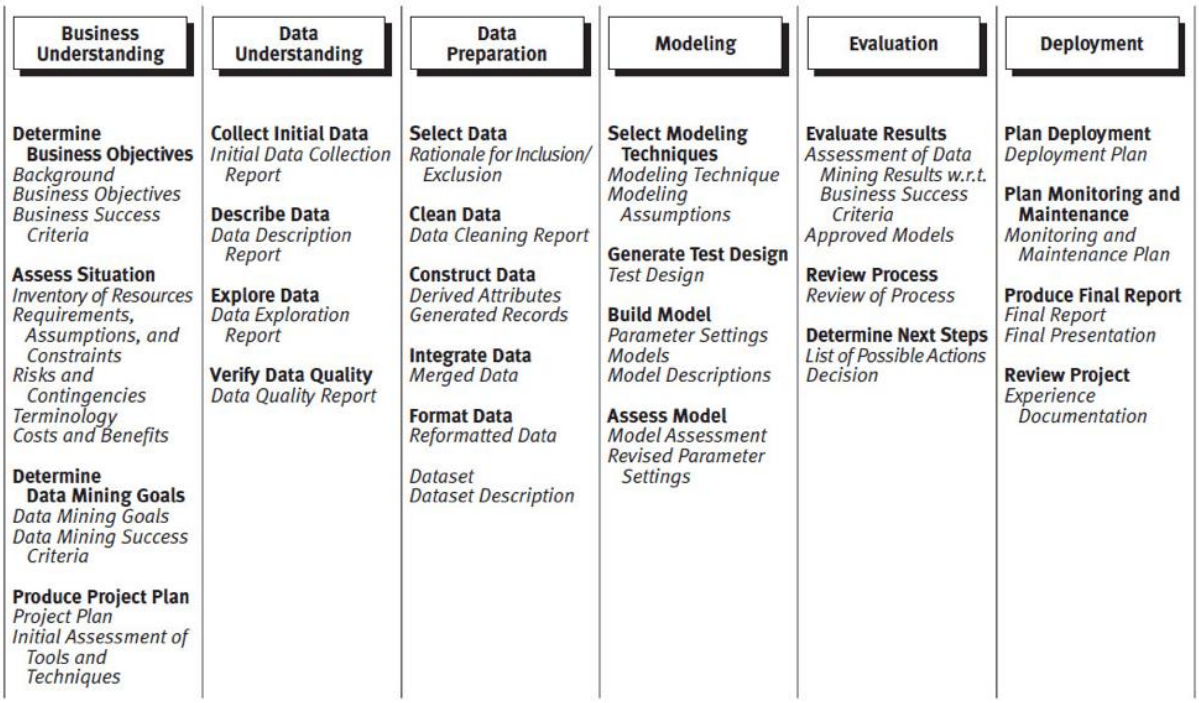


Figure 2: CRISP-DM Tasks and Outcomes

3.2 Machine Learning Workflow

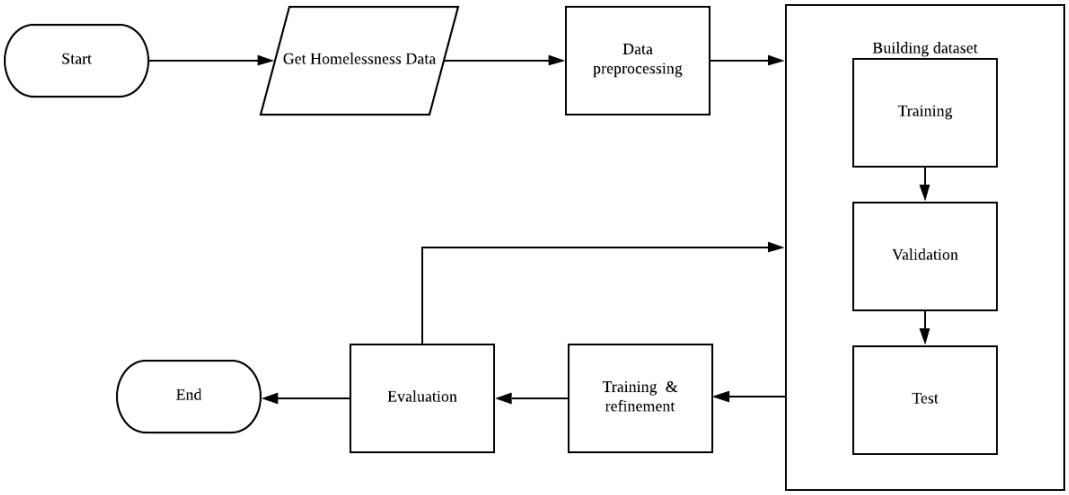


Figure 3: Project Flowchart

Machine learning workflows specify the actions that must be taken throughout a certain machine learning implementation. Machine learning workflows differ depending on the project, but four fundamental steps are usually covered.

One of the most essential steps in machine learning workflows is data collection. With the quality of the data we acquire during data collection, we define the potential usefulness and accuracy of this project. To gather data, we must first identify our sources and then combine data from those sources into a single dataset. This might include obtaining homelessness data sets from Department of Housing and Urban Development.

After we have collected our data, we will need to pre-process it. Cleaning, validating, and converting data into a useable dataset is what pre-processing entails. This may be a very simple procedure if we collect data from a single source. However, if we are aggregating data from many sources, we must ensure that the data formats match, that the data is similarly credible, and that any potential duplicates are removed.

The next step is to build a dataset. This step divides the processed data into three datasets: training, validating, and testing. For the training set, it is used to teach the algorithm how to handle data and train it. The parameters in this set define classifications of model. Next, the accuracy of the model is evaluated using the

validation set and the parameters of the model are adjusted using this dataset. In addition, the test set is utilized to check the accuracy and performance of the models and is meant to draw attention to any flaws or inconsistencies in the model.

Once we have our datasets, we train our model. This entails giving our training data to our algorithm so that it may learn proper classification parameters and features. After training, we use our validation dataset to refine the model. This may require changing or removing variables, as well as fine-tuning model-specific settings which is hyperparameters until an acceptable level of accuracy is achieved.

Finally, when we have identified an appropriate collection of hyperparameters and optimized our model's accuracy, we can test our model. Testing makes use of our test dataset and is intended to ensure that our models employ accurate features. We may return to training the model to increase accuracy, modify output parameters, or deploy the model as needed based on the input we get.

3.3 Steps in Machine Learning

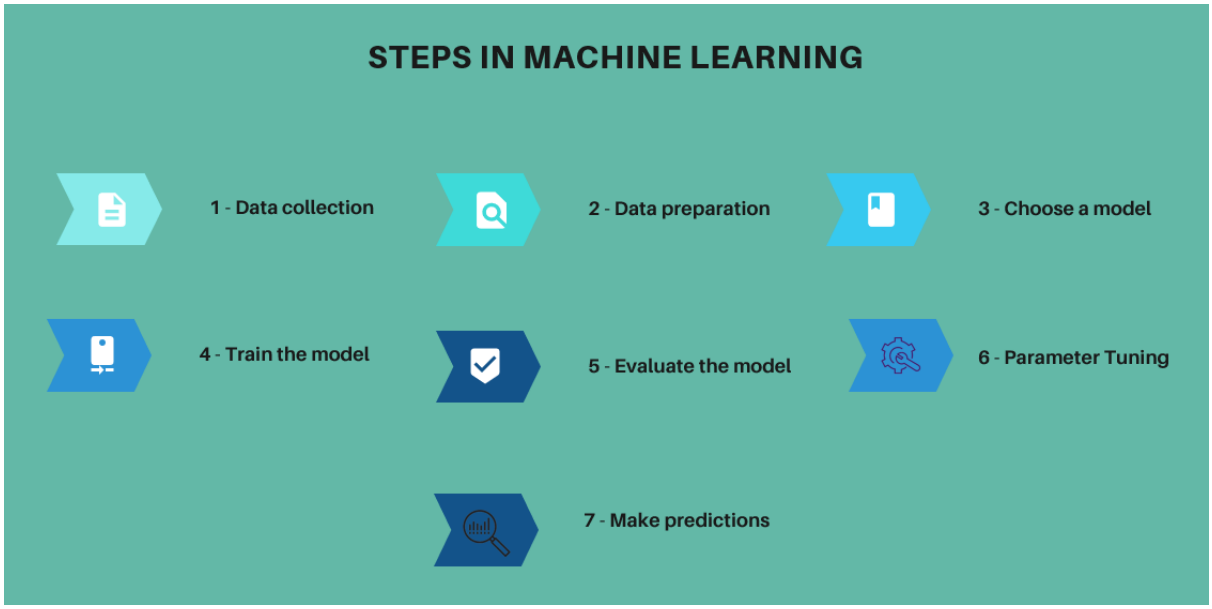


Figure 4: 7 Steps in Machine Learning

1. **Data collection**

   This phase is crucial because the quality and quantity of data we collect will directly impact the success of our prediction model. The information we gather for this project will be the various backgrounds of sheltered and unsheltered homeless people in different areas. The dataset that we collect is from HUD.

2. **Data preparation**

   The process of loading our data into a suitable location and preparing it for machine learning training is known as data preparation. We will begin by gathering all of our information and then randomize the order. We do not want the demand of our facts to affect what we learn because it has nothing to do with determining whether a homeless person is sheltered or unsheltered. This is also an excellent opportunity to perform any relevant data visualizations to see any important correlations between various factors to see any data imbalances. So, we will need to split the data into two sections as well. The majority of the dataset will be in the first section, which will train our model. The performance of our trained model will be evaluated in the second section.

- **Removing null values from dataset**

  In data preparation step, we do data cleansing method. After we have examined our data, we should remove some of the rows that have null values. One of the most crucial phases in data preparation is removing null values from the dataset. Any machine learning algorithm's performance and accuracy suffer as a result of these null values. As a result, it is critical to eliminate null values from a dataset before using a machine learning algorithm on it. Although some algorithms, such as XGBoost, have built-in support for null values, we should still do it manually as a best practice when preparing the data.

  In this HUD dataset, the initial dataset has 3008 rows and 332 columns. We removed rows that have null values by using 'dropna' method from the columns:

  i. total sheltered - HUD PIT
  ii. total unsheltered - HUD PIT
  iii. total homeless - HUD PIT
  iv. individuals sheltered - HUD PIT
  v. individuals unsheltered - HUD PIT
  vi. total individuals - HUD PIT
  vii. persons in families sheltered - HUD PIT
  viii. persons in Families unsheltered - HUD PIT
  ix. total persons in families - HUD PIT
  x. total chronically homeless individuals - HUD PIT
  xi. total chronically homeless persons in families - HUD PIT
  xii. total veterans - HUD PIT

  After we have done removing rows from the mentioned columns, there are 1871 rows remaining.

- **Removing unwanted columns from dataset**

  From the dataset, we also realized that we do not want the columns that have object datatype. So, we remove them by using drop method and select the subset continuum of care number (cocnumber) and state

abbreviation (state_abr). The results shows that there are 330 remaining columns.

3. Choose a model

The next stage in our process is to select a model. Over the years, academics and data scientists have developed a variety of models. Some are better suited to visual data, while others are better suited to sequences like music data, numerical data, and text-based data. In this project, since we divide them into two categories which are sheltered and unsheltered homeless, we use Random Forest model which has a significantly reduced time complexity.

4. Training

Move on to the widely considered the most important machine learning aspect which is training. In this case, we use the portion of the data set designated for training to teach our model to determine whether they come from different areas urbanicity category of homeless. In mathematical words, the inputs which are our two features, would have coefficients. These coefficients are known as feature weights. A constant or y-intercept would also be included. This is referred regarded as the model's bias. Their values are determined by trial and error. We begin by assigning them random values and providing inputs. The obtained output is compared to the actual output, and the disparity is reduced by experimenting with different weights and biases values. Iterations are performed using different entries from our training data set until the model achieves the required degree of accuracy.

5. Evaluation

It is time to put the model to the evaluation after being trained to see if it is any good. This is where the previously saved dataset comes in handy. By evaluating our model, we can test it against data that has never been used for training. In addition, this statistic allows us to forecast how the model will perform when faced with data it hasn't seen before. This is meant to show how the model might work in real life.

6. Parameter tuning

After we have completed our evaluation, we might want to see if there is any way we can improve our training. By fine-tuning our parameters, we can accomplish this. When we trained, we made some implicit assumptions about a few parameters, and now is a good time to double-check those assumptions and experiment with different values. One example would be the number of times we go over the training dataset during training. It means that rather than just once, we can "display" the model our entire dataset multiple times. This can sometimes lead to increased precision. The "learning rate" is another variable to consider. It means that the distance of the shift line during each step according to the previous training phase. All of these variables impact the accuracy of our model and the time taken to train the model.

7. Prediction

Data is used by machine learning to respond to queries. The questions' answers we get are in the inference or prediction stage. This is where all of our efforts come together and where the value of machine learning is recognized. Finally, given the different types of homeless, we can use our algorithm to predict the number of homeless people in different urbanicity category.

3.4 Gantt Chart

Table 1: Project Gantt Chart

Project Elements	Weeks											
	1	2	3	4	5	6	7	8	9	10	11	12
Phase 1: Business Understanding												
Determine business objectives												
Assess the situation												
Determine data mining goals												
Produce project plan												
Phase 2: Data Understanding												
Collect initial data												
Describe data												
Explore data												
Verify data quality												
Phase 3: Data Preparation												
Select data												
Data Cleansing												
Construct data												
Integrate data												
Format data												
Phase 4: Modeling												
Select modeling techniques												
Generate test design												
Build model												
Assess model												
Phase 5: Evaluation												
Evaluate results												
Review process												
Determine next steps												
Phase 6: Deployment												
Plan deployment												
Plan monitoring and maintenance												
Product final report												
Review project												

For phase 1 business understanding which is during the first week, we determine the business objectives where the objectives of this project are to analyze the background of homeless and do the predictive analytics for homeless in different urbanicity categories. Then we will assess the situation where we will find the risk for this project where sometimes it does not follow the timeline. Next, we determine the data mining goals where we want to build a model using available homeless data to predict the likelihood of individual being homeless.

Next, phase 2 is data understanding during week 2. We collect initial data from HUD. We also describe the data referring to their data dictionary, explore the data to enhance our understanding for future data preparation. Lastly, we verify the data quality to help us in identifying data errors that need to be solved.

Onto the data preparation phase during week 3 and 4, where we select data that are relevant to be insert in our prediction method later. Then, we do data cleansing to remove certain data from initial data such as removing null

values, remove attributes that have the datatype object and so on. We also do data reformatting by changing the datatype from float to integer.

In phase 4 which is modeling, it takes time to select the right modelling technique which is random forest and compare with linear regression algorithm as well. We build the model by using certain machine learning libraries like Scikit-learn.

In the evaluation phase during week 11, we evaluate the results where we do training and testing the dataset. We get the accuracy of all the model we select to see their performance. Then we review the process again to make sure the model is meeting the accuracy.

Finally, the deployment which is we produce the final report and review the project to our supervisor to meet the current objective of our project.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Exploratory Data Analysis

In this exploratory data analysis, we analyze few descriptive analytics for the dataset and create data visualization to enhance our understanding based on the different background of homeless. We run a few codes and create some charts such as bar graph and pie chart to conclude the overall study of homelessness.

```
#display the dataframe dataypes
df.dtypes
```

```
year                          int64
cocnumber                    object
pit_tot_shelt_pit_hud       float64
pit_tot_unshelt_pit_hud     float64
pit_tot_hless_pit_hud       float64
                             ...
sub_west_coast_all_urb        int64
sub_west_census               int64
major_city                    int64
suburban                      int64
rural                         int64
Length: 332, dtype: object
```

Figure 5: Dataframe datatypes

Figure above shows that we run the code by using df.dtypes where we want to display the dataframe datatypes. The output shows that it includes integer, object and float datatypes.

```
df.head(5)
```

	year	cocnumber	pit_tot_shelt_pit_hud	pit_tot_unshelt_pit_hud	pit_tot_hless_pit_hud	pit_ind_shelt_pit_hud	pit_ind_unshelt_pit_hud	pit_ind_hless_pit_hud	p
0	2010	AK-500	1113.0	118.0	1231.0	633.0	107.0	740.0	
1	2011	AK-500	1082.0	141.0	1223.0	677.0	117.0	794.0	
2	2012	AK-500	1097.0	50.0	1147.0	756.0	35.0	791.0	
3	2013	AK-500	1070.0	52.0	1122.0	792.0	52.0	844.0	
4	2014	AK-500	970.0	53.0	1023.0	688.0	48.0	736.0	

5 rows × 332 columns

Figure 6: Dataframe head

Figure above shows when we want to display only first 5 rows of the dataset. The dataframe only display the data from the year 2010 to 2014 from cocnumber 'AK-500'.

```
# Check for null values
df.isnull().sum()
```

```
year                      0
cocnumber                 0
pit_tot_shelt_pit_hud    14
pit_tot_unshelt_pit_hud  14
pit_tot_hless_pit_hud    14
                          ..
sub_west_coast_all_urb    0
sub_west_census           0
major_city                0
suburban                  0
rural                     0
Length: 332, dtype: int64
```

Figure 7: Null Values Checking

Moreover, we want to check whether there are null values or not in the dataset. We use df.isnull().sum() to visualize the columns that have null values. For example, figure above shows that there are 14 rows that have null values came from column "pit_tot_shelt_pit_hud", "pit_tot_unshelt_pit_hud", "pit_tot_hless_pit_hud" which is the total sheltered, total unsheltered, total homeless that came from HUD.

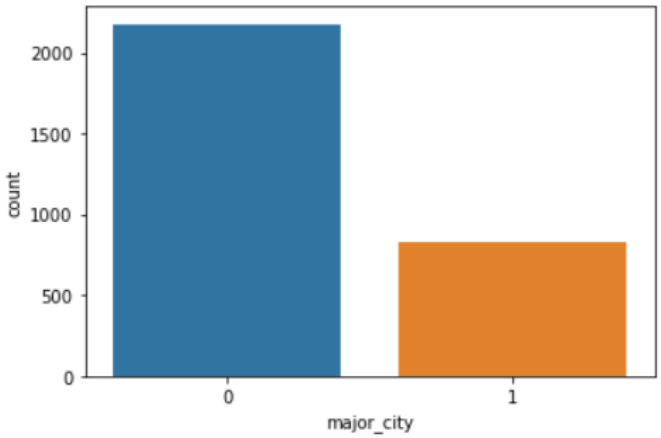


Figure 8: Bar Graph of Major City Homeless

Figure above shows that for 0 number, it indicates that it is not from major city homeless, while 1 is homeless from major city. The number of people experience homelessness from 2010 until 2017 at major city is only 832 people. The people who were not experiencing homelessness in major city is higher which is 2176 people.

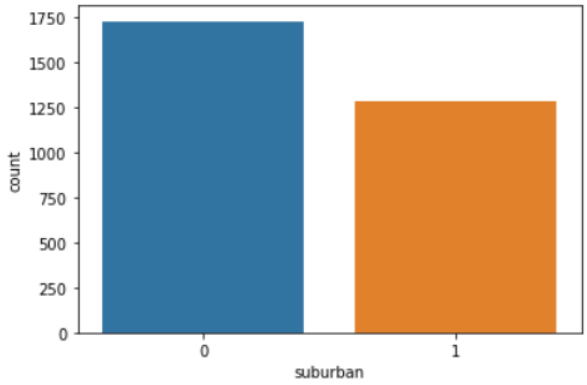


Figure 9: Bar Graph of Suburban Homeless

Onto the suburban area, it shows that the number of people experience homelessness from 2010 until 2017 at suburban is only 1280 people. The people who were not experiencing homelessness in suburban is higher which is 2176 people.

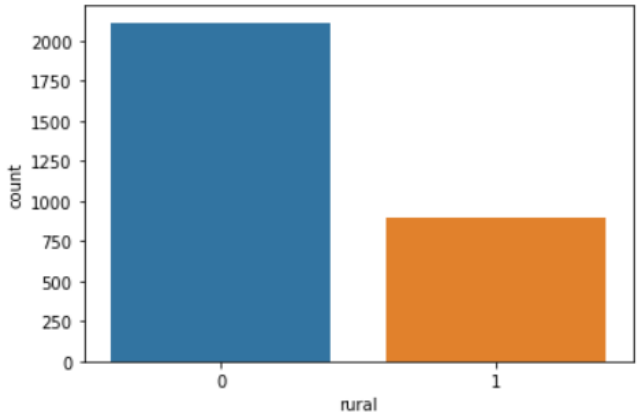


Figure 10: Bar Graph of Rural Homeless

For rural area, the number of people experience homelessness from 2010 until 2017 is lower than people are not coming from rural which is only 896 people. The people who were not experiencing homelessness in rural is higher which is 2112 people.

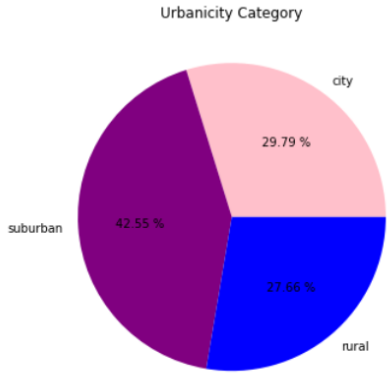


Figure 11: Pie Chart of Urbanicity Category

From the pie chart above, we can see that there are 3 different areas people undergo homelessness which are city, suburban and rural. Homeless that came from suburban area has the highest percentage where 42.55% is a collection of mostly residential properties that are not tightly packed but are close to a densely packed metropolitan area. People that are undergo city homelessness is higher than in rural area which is 29.79%. The difference between city and rural area is only 2.13%. Homeless from rural area experience the least which is only 27.66% because their poverty rate is high, their income is insufficient and mostly unemployed.

4.2 Comparative Study between Random Forest and Linear Regression Algorithm

In this comparative study, we discover different type of algorithm and differentiate the results from Random Forest and Linear Regression algorithm. There are a few steps we conduct during this study.

Random Forest

a) Choose the main column for major city, suburban and rural

In this step, we create a dataframe where we only select important columns that we want to classify. There are only 10 columns that we have chosen which are:

i. total sheltered - HUD PIT
ii. total unsheltered - HUD PIT
iii. individuals sheltered - HUD PIT
iv. individuals unsheltered - HUD PIT
v. persons in families sheltered - HUD PIT
vi. persons in Families unsheltered - HUD PIT
vii. total chronically homeless individuals - HUD PIT
viii. total chronically homeless persons in families - HUD PIT
ix. total veterans - HUD PIT
x. major city / suburban / rural

b) Fill in the null values with mean

This is the statistical procedure for dealing with null values. When compared to deleting null values, this approach produces good results. When the data is regularly distributed, the mean of the numerical column data is utilized to replace null values. For example in this dataframe, we run the code majorcity.fillna(majorcity.mean(), inplace=True) to fill the null values with mean.

c) Split the data into train and test (X and Y)

- Firstly, we must remove major_city column.

```
X=majorcity.drop('major_city',axis=1)
Y= majorcity['major_city']
```

Figure 12: X and Y Data Splitting

- Then we import the sklearn library and partition the major city data into train and test sets. We have used the 'train_test_split' function to split the data in an 80:20 ratio, which means that 80% of the data will be used to train the model and 20% will be used to test the model.

```
# Import the Sklearn library and partition the major city data into train and test sets.
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=86)
```

Figure 13: Import Library

- Next, we train the major city dataset.

```
X_train
```

	pit_tot_shelt_pit_hud	pit_tot_unshelt_pit_hud	pit_ind_shelt_pit_hud	pit_ind_unshelt_pit_hud	pit_perfam_shelt_pit_hud	pit_perfam_unshelt_pit_hud	pit_ind_
122	17.0	40.0	14.0	20.0	3.0	20.0	
107	122.0	132.0	111.0	61.0	11.0	71.0	
2153	1495.0	42.0	641.0	42.0	854.0	0.0	
1791	657.0	110.0	342.0	82.0	315.0	28.0	
2755	697.0	45.0	456.0	41.0	241.0	4.0	
...	...	...	...	...	...	...	
1612	1702.0	229.0	1263.0	208.0	439.0	21.0	
1001	1473.0	114.0	1079.0	114.0	394.0	0.0	
2075	133.0	7.0	84.0	7.0	49.0	0.0	
1123	321.0	193.0	171.0	186.0	150.0	7.0	
1888	146.0	0.0	83.0	0.0	63.0	0.0	

2406 rows × 9 columns

Figure 14: X training

- Test the major city dataset.

```
X_test
```

	pit_tot_shelt_pit_hud	pit_tot_unshelt_pit_hud	pit_ind_shelt_pit_hud	pit_ind_unshelt_pit_hud	pit_perfam_shelt_pit_hud	pit_perfam_unshelt_pit_hud	pit_ind_
225	142.0	1314.0	98.0	1254.0	44.0	60.0	
1146	658.0	59.0	425.0	59.0	233.0	0.0	
2148	3143.0	64.0	855.0	64.0	2288.0	0.0	
1657	476.0	51.0	348.0	51.0	128.0	0.0	
1153	1450.0	18.0	382.0	18.0	1068.0	0.0	
...	...	...	...	...	...	...	
86	469.0	339.0	408.0	334.0	61.0	5.0	
2378	5280.0	500.0	2476.0	500.0	2804.0	0.0	
2445	1994.0	416.0	847.0	354.0	1147.0	62.0	
1437	166.0	13.0	112.0	13.0	54.0	0.0	
2354	2035.0	4070.0	1232.0	2110.0	803.0	1960.0	

602 rows × 9 columns

Figure 15: X testing

- Train the major_city column.

```
Y_train
```

```
122     0
107     0
2153    0
1791    0
2755    0
       ..
1612    1
1001    1
2075    0
1123    0
1888    0
Name: major_city, Length: 2406, dtype: int64
```

Figure 16: Y training

- Test the major_city column.

```
Y_test
```

```
225     0
1146    0
2148    0
1657    1
1153    0
       ..
86      1
2378    1
2445    0
1437    0
2354    0
Name: major_city, Length: 602, dtype: int64
```

Figure 17: Y testing

d) Modeling

- Use Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier # classifier
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

# Generate dt
# Build dt from the training set
dt=RandomForestClassifier(n_estimators = 1400,
 min_samples_split= 5,
 min_samples_leaf= 1,
 max_features = 'sqrt',
 max_depth = None,
 bootstrap =True)
dt.fit(X_train,Y_train)
```

```
RandomForestClassifier(max_features='sqrt', min_samples_split=5,
                       n_estimators=1400)
```

Figure 18: Random Forest Classifier Code

e) Evaluation

- Calculate the accuracy score, display the confusion matrix and classification report.

```
# Create predict method for X inputs
# Display accuracy score, confusion matrix and classification report
prediction=dt.predict(X_test)
print(f"Accuracy Score = {accuracy_score(Y_test,prediction)*100}")
print(f"Confusion Matrix =\n {confusion_matrix(Y_test,prediction)}")
print(f"Classification Report =\n {classification_report(Y_test,prediction)}")
```

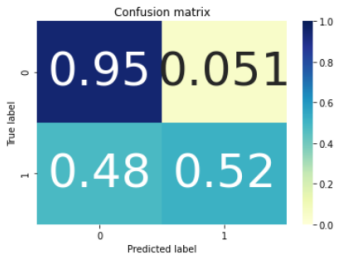
```
Accuracy Score = 82.55813953488372
Confusion Matrix =
 [[408  22]
 [ 83  89]]
Classification Report =
              precision    recall  f1-score   support

           0       0.83      0.95      0.89       430
           1       0.80      0.52      0.63       172

    accuracy                           0.83       602
   macro avg       0.82      0.73      0.76       602
weighted avg       0.82      0.83      0.81       602
```

Figure 19: Accuracy Score, Confusion Matrix, Classification Report

Figure above shows that the accuracy score is 83% which is suitable for modeling.

```
# Confusion Matrix function

def plot_confusion_matrix(cm, classes=None, title='Confusion matrix'):
    """Plots a confusion matrix."""
    if classes is not None:
        sns.heatmap(cm, cmap="YlGnBu", xticklabels=classes, yticklabels=classes, vmin=0., vmax=1., annot=True, annot_kws={'size':
    else:
        sns.heatmap(cm, vmin=0., vmax=1.)
    plt.title(title)
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

    #A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on
```

```
# Visualize and plot cm
import seaborn as sns

cm = confusion_matrix(Y_test,prediction)
cm_norm = cm / cm.sum(axis=1).reshape(-1,1)

plot_confusion_matrix(cm_norm, classes = dt.classes_, title='Confusion matrix')
```




```
# Calculate False Positives (FP), False Negatives (FN), True Positives (TP) & True Negatives (TN)
import numpy as np

FP = cm.sum(axis=0) - np.diag(cm)
FN = cm.sum(axis=1) - np.diag(cm)
TP = np.diag(cm)
TN = cm.sum() - (FP + FN + TP)


# Sensitivity, hit rate, recall, or true positive rate
TPR = TP / (TP + FN)
print("The True Positive Rate is:", TPR)

# Precision or positive predictive value
PPV = TP / (TP + FP)
print("The Precision is:", PPV)

# False positive rate or False alarm rate
FPR = FP / (FP + TN)
print("The False positive rate is:", FPR)


# False negative rate or Miss Rate
FNR = FN / (FN + TP)
print("The False Negative Rate is: ", FNR)


# Total averages :
print("")
print("The average TPR is:", TPR.sum()/2)
print("The average Precision is:", PPV.sum()/2)
print("The average False positive rate is:", FPR.sum()/2)
print("The average False Negative Rate is:", FNR.sum()/2)
```

```
The True Positive Rate is: [0.94883721 0.51744186]
The Precision is: [0.83095723 0.8018018 ]
The False positive rate is: [0.48255814 0.05116279]
The False Negative Rate is:  [0.05116279 0.48255814]

The average TPR is: 0.733139534883721
The average Precision is: 0.8163795159721841
The average False positive rate is: 0.2668604651162791
The average False Negative Rate is: 0.2668604651162791
```

Figure 20: True Positive, True Negative, False Positive, False Negative

We can use accuracy, which will provide us correctly classified results, when our classes are about equal in size. For classification problems, accuracy is a popular evaluation parameter. It's the number of right guesses divided by the total number of predictions.

Accuracy can become an unreliable criterion for judging our performance when there is a class imbalance (Chauhan, 2020). For example, if we had a 99/1 split between two classes, A and B, with B being our positive class and A being the unusual event, we could develop a model that was 99 percent accurate by simply declaring everything belonged to class A. Obviously, we shouldn't bother creating a model if it doesn't help us identify class B; as a result, we'll need different metrics to discourage this behaviour. Instead of accuracy, precision and recall are used.

The true positive rate (TPR), which is the ratio of genuine positives to everything positive, is determined by recall. The model that classifies everything as A would have a recall of 0% for the positive class, B (precision would be undefined — 0/0) in the case of the 99/1 split between classes A and B. In the presence of a class imbalance, precision and recall provide a superior approach of measuring model performance. They'll inform us that the model isn't really useful in our situation.

The F1 score is the harmonic mean of precision and memory, with 1 (perfect precision and recall) being the highest and 0 being the worst. Extreme values are punished more severely in an F1 score. In the following classification circumstances, an F1 Score could be an effective evaluation metric: Whether FP and FN are both similarly expensive that is, when they miss real positives or uncover false positives, they have nearly identical effects on the model. Adding new data won't impact the outcome because the TN is so high.

f) Hyperparameter tuning (random grid)

```
#hyperparameter tuning
from sklearn.model_selection import RandomizedSearchCV

# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]

# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}

print(random_grid)
```

```
{'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000], 'max_features': ['auto', 'sqrt'], 'max_depth': [10,
20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4], 'bootstrap':
[True, False]}
```

Figure 21: Hyperparameter Tuning

g) Make prediction

```
Homelessness_in_MajorCity = dt.predict([[970.0,53.0,688.0,48.0,282.0,5.0,94.0,7.0,138.0]])[0]
print('Homelessness indication = '+
      str(Homelessness_in_MajorCity))
```

```
Homelessness indication = 1
```

```
Homelessness_in_MajorCity = dt.predict([[501.0,452.0,306.0,371.0,195.0,81.0,221.0,23.0,137.0]])[0]
print('Homelessness indication = '+
      str(Homelessness_in_MajorCity))
```

```
Homelessness indication = 0
```

Figure 22: Homeless Prediction in Major City

The final results shows that if the output is 1, it means that the homeless experience the homeless in major city, while 0 is not experiencing homeless from major city.

Linear Regression

For comparing the result between linear regression and random forest, the differences are from modeling and evaluation part. Other steps remain the same. Figure below shows the accuracy score, confusion matrix and classification report for Linear Regression.

```
# Create predict method for X inputs
# Display accuracy score, confusion matrix and classification report

prediction=dt.predict(X3_test)
print(f"Accuracy Score = {accuracy_score(Y3_test,prediction)*100}")
print(f"Confusion Matrix =\n {confusion_matrix(Y3_test,prediction)}")
print(f"Classification Report =\n {classification_report(Y3_test,prediction)}")
```

```
Accuracy Score = 56.644518272425245
Confusion Matrix =
 [[332  98]
 [163   9]]
Classification Report =
              precision    recall  f1-score   support

           0       0.67      0.77      0.72       430
           1       0.08      0.05      0.06       172

    accuracy                           0.57       602
   macro avg       0.38      0.41      0.39       602
weighted avg       0.50      0.57      0.53       602
```

Figure 23: Accuracy Score for Linear Regression

Based on the figure above, it shows that the accuracy score for linear regression is low which is only 57%. There are a few reasons why this algorithm is not suitable for classification. The first is that Linear Regression is concerned with continuous values, whereas classification issues require discrete values. The second issue is that when new data points are added, the threshold value shifts (Narasimhan, 2021).

```
sheltered_homeless_majorcity=sm.OLS(Y3_train,X3_train).fit()
```

```
print(sheltered_homeless_majorcity.params)
```

```
pit_tot_shelt_pit_hud                0.000064
pit_tot_unshelt_pit_hud             42.799150
pit_ind_shelt_pit_hud                0.000385
pit_ind_unshelt_pit_hud            -42.799326
pit_perfam_shelt_pit_hud            -0.000321
pit_perfam_unshelt_pit_hud         -42.799258
pit_ind_chronic_hless_pit_hud        0.000283
pit_perfam_chronic_hless_pit_hud     0.000430
pit_vet_hless_pit_hud                0.000252
dtype: float64
```

Figure 24: Model Fitting and Print Parameter

```
sheltered_homeless_majorcity.summary2()
```

Model:	OLS	Adj. R-squared (uncentered):	0.297
Dependent Variable:	major_city	AIC:	2875.7456
Date:	2021-11-28 13:53	BIC:	2922.0314
No. Observations:	2406	Log-Likelihood:	-1429.9
Df Model:	8	F-statistic:	128.1
Df Residuals:	2398	Prob (F-statistic):	4.13e-179
R-squared (uncentered):	0.299	Scale:	0.19283

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
pit_tot_shelt_pit_hud	0.0001	0.0000	13.9993	0.0000	0.0001	0.0001
pit_tot_unshelt_pit_hud	42.7991	39.6683	1.0789	0.2807	-34.9885	120.5868
pit_ind_shelt_pit_hud	0.0004	0.0000	18.8119	0.0000	0.0003	0.0004
pit_ind_unshelt_pit_hud	-42.7993	39.6683	-1.0789	0.2807	-120.5870	34.9883
pit_perfam_shelt_pit_hud	-0.0003	0.0000	-18.9904	0.0000	-0.0004	-0.0003
pit_perfam_unshelt_pit_hud	-42.7993	39.6683	-1.0789	0.2807	-120.5869	34.9884
pit_ind_chronic_hless_pit_hud	0.0003	0.0001	5.5525	0.0000	0.0002	0.0004
pit_perfam_chronic_hless_pit_hud	0.0004	0.0001	3.2288	0.0013	0.0002	0.0007
pit_vet_hless_pit_hud	0.0003	0.0001	5.0076	0.0000	0.0002	0.0004

Omnibus:	185.621	Durbin-Watson:	1.867
Prob(Omnibus):	0.000	Jarque-Bera (JB):	269.797
Skew:	0.621	Prob(JB):	0.000
Kurtosis:	4.071	Condition No.:	5031357456479047

Figure 25: Summary of Model

```
Y3_pred_test=sheltered_homeless_majorcity.predict(X3_test)
Y3_pred_train=sheltered_homeless_majorcity.predict(X3_train)
```

```
np.abs(r2_score(Y3_test,Y3_pred_test))
```

```
1495.9867854086756
```

```
np.sqrt(mean_squared_error(Y3_test,Y3_pred_test))
```

```
17.47877304693497
```

Figure 26: Predict values test dataset, R square for train data and calculating root mean squared error

```
plt.figure(figsize=(15,10))
plt.scatter(Y3_train,Y3_pred_train,c='green')
plt.plot([Y3_train.min(),Y3_train.max()],[Y3_train.min(),Y3_train.max()],'k--',c='blue',lw=3)
plt.xlabel('Actual')
plt.ylabel('Predicted')
```

```
Text(0, 0.5, 'Predicted')
```

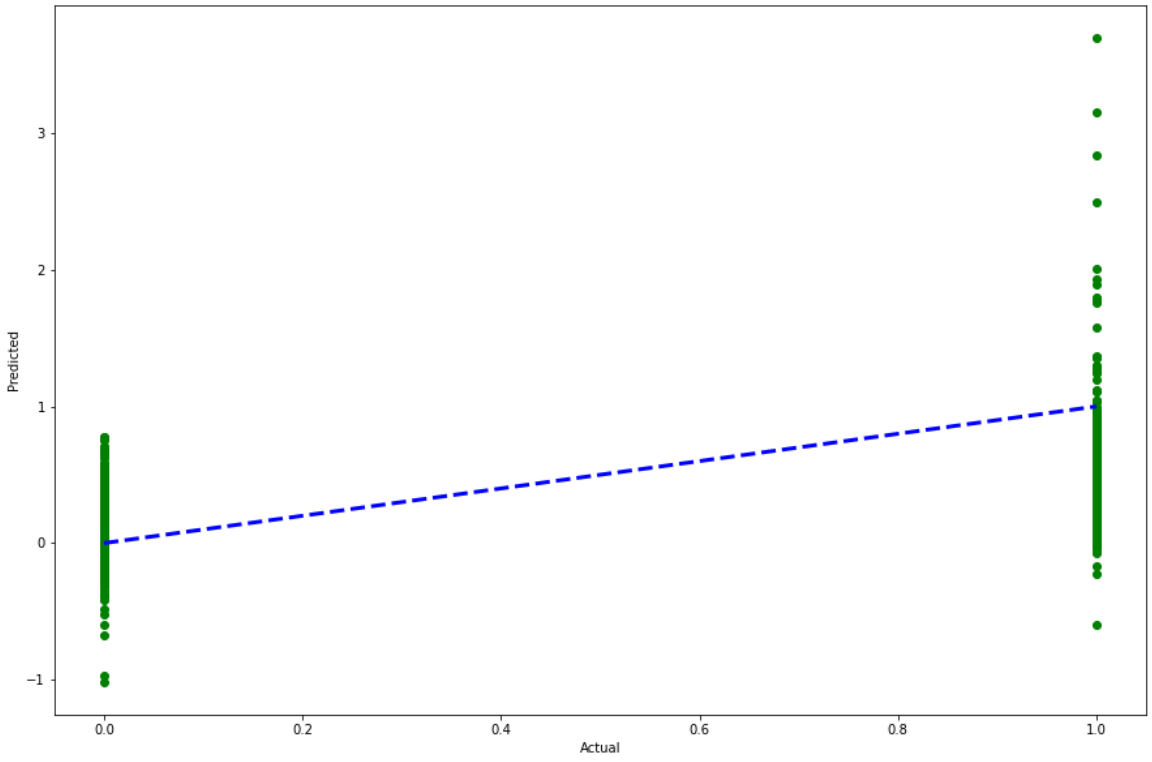


Figure 27: Scatter Plot for Train Data

The range of "homeless" values seen in the training dataset is clearly outside the range of predicted homeless. A Linear Regression model constructed a linear model on the data, as the name implies. The formula y = mx+C is a simple way to think about it. As a result, because it fits a linear model, it can predict values from outside the training set. It has the ability to extrapolate from the data.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In conclusion, the machine learning method used for this project is classification. Machine learning includes the key processes involved in the transformation of raw data into training data sets capable of allowing a system's decision making. This project used Random Forest algorithm as it best at determining which area category a homeless belongs to and predict the number of homeless. When presented with never-before-seen data, the model is trained to detect the underlying patterns and relationships between the input data and the output labels, allowing it to produce accurate labelling results.

5.2 Future Work

CRISP-DM is a future-proof option for anyone looking to solve data science problems because of its flexible and iterative approach. While it is critical to develop a unique method, it is also important to remember that using CRISP-DM adds professionalism and consistency to operational procedures. In the future, we might develop more enhance existing methods on classifying the homeless background and collect more data about them.

REFERENCES

Aizuddin, A. N., Abdul Jabar, S. W., & Idris, I. B. (2019). Factors associated with health services financier among temporary sheltered homeless in urban Malaysia. *BMC Public Health*, *19*(S4). https://doi.org/10.1186/s12889-019-6871-5

Byrne T, Baggett T, Land T, Bernson D, Hood M-E, Kennedy-Perez C, et al. (2020) A classification model of homelessness using integrated administrative data: Implications for targeting interventions to improve the housing status, health and well-being of a highly vulnerable population. PLoS ONE 15(8): e0237905. https://doi.org/10.1371/journal.pone.0237905

Chauhan, N. S. (2020). *Model Evaluation Metrics in Machine Learning*. KDnuggets. https://www.kdnuggets.com/2020/05/model-evaluation-metrics-machine-learning.html

Hong, B., Malik, A., Lundquist, J., Bellach, I., & Kontokosta, C. E. (2018). Applications of Machine Learning Methods to Predict Readmission and Length-of-Stay for Homeless Families: The Case of Win Shelters in New York City. *Journal of Technology in Human Services, 36:1, 89-104.* https://www.tandfonline.com/doi/abs/10.1080/15228835.2017.1418703?journalCode=wths20

Irsyad, A. (2016) Number of Homeless People Increased by Three Fold in Kuala Lumpur, What Are We Doing to Curb The Problem. Retrieved from Malaysian Digest: malaysiandigest.com

Kay Li, W. (2018). *The Homeless in Malaysia: Issues and Policy Solutions*. Wong Chen. https://www.wongchen.com/wp-content/uploads/2014/03/KayLi-The-Homeless-in-Malaysia-1-1.pdf

Lia, M. (2020) *Homelessness in Malaysia: NGO and Government Collaboration*. The Borgen Project. https://borgenproject.org/homelessness-in-malaysia/

Mabhala, M.A., Yohannes, A. & Griffith, M. (2017) Social conditions of becoming homelessness: qualitative analysis of life stories of homeless peoples. *Int J Equity Health* **16,** 150. https://doi.org/10.1186/s12939-017-0646-3

Narasimhan, A. K. (2021, February 5). *Why Linear Regression is not suitable for classification?* Medium. https://medium.com/analytics-vidhya/why-linear-regression-is-not-suitable-for-classification-cd724dd61cb8

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016) *"Why Should I Trust You?": Explaining the Predictions of Any Classifier.* ArXiv.Org. https://arxiv.org/abs/1602.04938

VanBerlo, B., Ross, M. A., Rivard, J., & Booker, R. (2020). Interpretable machine learning approaches to prediction of chronic homelessness. *Engineering Applications of Artificial Intelligence, 102*, 104243. https://doi.org/10.1016/j.engappai.2021.104243

Yani et al., (2016) Factors Associated with Homelessness and its Medical Issues among Urban Malaysians: A Qualitative Research, Journal of Clinical and Health Sciences, Vol 1 (1), pp 47. Retrieved from http://jchsmedicine.uitm.edu.my/images/article/original/Factors-and-Medical-Issues-Associated-With-HomelessUrban-Malaysians-A-Qualitative-Research.pdf

APPENDICES

Some codes run from Jupyter Notebook.

```
#import libraries
from sklearn.datasets import make_classification
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import pandas as pd
import statsmodels.api as sm
from sklearn.metrics import r2_score,mean_squared_error
```

```
#import csv file to the dataframe
df = pd.read_csv("05b_analysis_file_update.csv")
df.head()
```

	year	cocnumber	pit_tot_shelt_pit_hud	pit_tot_unshelt_pit_hud	pit_tot_hless_pit_hud	pit_ind_shelt_pit_hud	pit_ind_unshelt_pit_hud	pit_ind_hless_pit_hud	p
0	2010	AK-500	1113.0	118.0	1231.0	633.0	107.0	740.0	
1	2011	AK-500	1082.0	141.0	1223.0	677.0	117.0	794.0	
2	2012	AK-500	1097.0	50.0	1147.0	756.0	35.0	791.0	
3	2013	AK-500	1070.0	52.0	1122.0	792.0	52.0	844.0	
4	2014	AK-500	970.0	53.0	1023.0	688.0	48.0	736.0	

5 rows × 332 columns

```
# Run the last 5 rows
df.tail(5)
```

	year	cocnumber	pit_tot_shelt_pit_hud	pit_tot_unshelt_pit_hud	pit_tot_hless_pit_hud	pit_ind_shelt_pit_hud	pit_ind_unshelt_pit_hud	pit_ind_hless_pit_hud
3003	2013	WY-500	501.0	452.0	953.0	306.0	371.0	677.0
3004	2014	WY-500	563.0	194.0	757.0	327.0	136.0	463.0
3005	2015	WY-500	507.0	291.0	798.0	292.0	208.0	500.0
3006	2016	WY-500	491.0	366.0	857.0	277.0	240.0	517.0
3007	2017	WY-500	510.0	363.0	873.0	383.0	239.0	622.0

5 rows × 332 columns

```
# Check to see if the dataset contains any NaN values.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3008 entries, 0 to 3007
Columns: 332 entries, year to rural
dtypes: float64(257), int64(73), object(2)
memory usage: 7.6+ MB
```

```
# Analyze and list the descriptive statistics for the dataframe.
df.describe()
```

	year	pit_tot_shelt_pit_hud	pit_tot_unshelt_pit_hud	pit_tot_hless_pit_hud	pit_ind_shelt_pit_hud	pit_ind_unshelt_pit_hud	pit_ind_hless_pit_hud	pit
count	3008.000000	2994.000000	2994.000000	2994.000000	2994.000000	2994.000000	2994.000000	
mean	2013.500000	1033.852037	526.581830	1560.433868	540.342351	442.891784	983.230795	
std	2.291669	3445.807044	1742.313472	4313.387045	1367.938173	1569.110148	2497.924191	
min	2010.000000	3.000000	0.000000	7.000000	0.000000	0.000000	2.000000	
25%	2011.750000	224.000000	36.250000	320.250000	120.250000	31.000000	192.000000	
50%	2013.500000	445.500000	114.000000	679.000000	253.000000	98.000000	428.000000	
75%	2015.250000	961.500000	418.250000	1468.000000	560.000000	337.000000	930.000000	
max	2017.000000	72565.000000	42828.000000	76501.000000	27188.000000	41241.000000	49265.000000	

8 rows × 330 columns

```
sub_urban = df[[ 'pit_tot_shelt_pit_hud', 'pit_tot_unshelt_pit_hud', 'pit_ind_shelt_pit_hud','pit_ind_unshelt_pit_hud', 'pit_perf
```

```
pd.set_option('display.max_columns', None)
```

```
sub_urban
```

	pit_tot_shelt_pit_hud	pit_tot_unshelt_pit_hud	pit_ind_shelt_pit_hud	pit_ind_unshelt_pit_hud	pit_perfam_shelt_pit_hud	pit_perfam_unshelt_pit_hud	pit_ind_
0	1113.0	118.0	633.0	107.0	480.0	11.0	
1	1082.0	141.0	677.0	117.0	405.0	24.0	
2	1097.0	50.0	756.0	35.0	341.0	15.0	
3	1070.0	52.0	792.0	52.0	278.0	0.0	
4	970.0	53.0	688.0	48.0	282.0	5.0	
...	...	...	...	...	...	...	
3003	501.0	452.0	306.0	371.0	195.0	81.0	
3004	563.0	194.0	327.0	136.0	236.0	58.0	
3005	507.0	291.0	292.0	208.0	215.0	83.0	
3006	491.0	366.0	277.0	240.0	214.0	126.0	
3007	510.0	363.0	383.0	239.0	127.0	124.0	

3008 rows × 10 columns

```
Rural = df[[ 'pit_tot_shelt_pit_hud', 'pit_tot_unshelt_pit_hud', 'pit_ind_shelt_pit_hud','pit_ind_unshelt_pit_hud', 'pit_perfam_s
```

```
Rural
```

	pit_tot_shelt_pit_hud	pit_tot_unshelt_pit_hud	pit_ind_shelt_pit_hud	pit_ind_unshelt_pit_hud	pit_perfam_shelt_pit_hud	pit_perfam_unshelt_pit_hud	pit_ind_
0	1113.0	118.0	633.0	107.0	480.0	11.0	
1	1082.0	141.0	677.0	117.0	405.0	24.0	
2	1097.0	50.0	756.0	35.0	341.0	15.0	
3	1070.0	52.0	792.0	52.0	278.0	0.0	
4	970.0	53.0	688.0	48.0	282.0	5.0	
...	...	...	...	...	...	...	
3003	501.0	452.0	306.0	371.0	195.0	81.0	
3004	563.0	194.0	327.0	136.0	236.0	58.0	
3005	507.0	291.0	292.0	208.0	215.0	83.0	
3006	491.0	366.0	277.0	240.0	214.0	126.0	
3007	510.0	363.0	383.0	239.0	127.0	124.0	

3008 rows × 10 columns