University of Waterloo

Faculty of Engineering  
Department of Electrical and Computer Engineering

Machine Learning for Income Predictor

Integrate.ai

55 Bloor St E, Toronto, ON M4W 1A9

Prepared by  
YuCheng Liu  
20558944  
y698liu@uwaterloo.ca  
3B Computer Engineering  
1s Jan 2018

YuCheng Liu

330 Phillip Street

Waterloo, Ontario, Canada

N2L 3W9

10 January 2018

Vincent Gaudet, Chair  
Electrical and Computer Engineering  
University of Waterloo  
Waterloo, Ontario  
N2L 3G1

Dear Sir,

This report, entitled “Accuracy Optimization of Income Predictor”, was prepared as my 3B Work Term Report for the University of Waterloo. This report is in fulfillment of the course WKRPT 301. The purpose of this report is to discuss the performance of the income predictor that use different machine learning algorithms.

Integrate.ai is a leading platform for large enterprises to train AI-enabled solutions that drive customer engagement and revenue growth. Integrate.ai's mission is to build a future in which AI enriches people’s lives while creating better, more valuable businesses. To do this the company is building an AI powered platform for B2C enterprises that integrates with business processes to make customer interactions more natural and valuable. Founded by Steve Irvine, former Facebook executive, we are proud to be based in Toronto, Canada at the centre of an exciting AI ecosystem. I was a machine learning engineer in the team. I focused on developing and enhancing machine learning models for different business cases.

I would like to thank David Findlay for guiding me during the development phase of the project. I would also like to thank Professor Douglas Harder for the reference of formatting the report. I hereby confirm that I have received no further help other than what has mentioned above in writing this report. I also confirm this report has not been previously submitted for academic credit at this or any other academic institution.

Sincerely,

YuCheng Liu

ID 20558944

# Contributions

This is a work-term report based on the experience gained at my previous co-op job. I worked in the machine learning team at Integrate.ai. It was a relatively large team in a startup company. My team consisted of roughly 12 employees. The team is in charge of analyzing business use case and developing accurate machine learning models. We were using Spark and PySpark as our big data development tool and Python and Scala as our data analyzing tool. The number of people working on this project was limited to myself and one other fulltime machine learning scientist and our supervisor, but we work closely with other people in the team to get requirements and technical help.

Our development team had many goals. Primarily, the group was helping the customer’s business in building accurate machine learning models that will help them to make their customer interactions more natural and valuable. For example, one of our customers is a technology company that is known for its widely advertised freemium mobile MMO strategy games. By analyzing their internal data and talking with their product managers, we have found a great business use case use machine learning to make the players feel more engaged in the game, which will result in the players spending more time and money on the game. We decided to build a recommendation system using demographic and in-game data to provide players that have never joined an alliance with a set of alliances with headcount for users of their classification. As an engineer, I am also a part of the process of data integration, data ingestion, and data cleaning. For example, in order to build and train a machine learning model, we need to have enough data points from the past to predict any unknown traits or any future events. Thus, we need to build a data pipeline that allows our customer to send their private data to us safely. After consuming the data, it is usually in a raw format like excel sheets or rows of data points, so they cannot be used directly and feed into a machine learning model. It would be my job to clean the data and found out the columns that are most effective to a machine learning model, which is known as feature selection. This is the first step of building a machine learning model, one of the strategies that I used is using the unsupervised learning algorithm K-Means on the raw data points to finding out which column might have an insight.

Working as a machine learning engineer, my tasks are mainly focused on building machine learning model, which is preparing the data, feature selection, model training and model accuracy tuning. I developed and trained machine learning models that are written and deployed in Pyspark and Python or Spark and Scala. Most of my work is completing requests from different product manager and project managers, where they work directly with the customers and define the business use case. However, the business development team is also a source that provides us requirements and tasks that is asked by the customer. For example, in order to make the machine learning model more accurate, sometimes we need to use third party data like regional data or weather data, but according to some enterprises’ data security policy, any third-party data that is used in the model must be encrypted and generalized so that no personal information is involved. Thus, I developed a utility function and it will be in our company’s tool library. Therefore, it will be a library that we use for any client. On the other hand, the algorithm that I used is a cryptography function that is one way only. As the result, even if the client sees the encrypted data, all the data that was original strings are all encrypted into an array of numbers, which no one can revert or understand. One of another utility functions that I have created is a tool that is implemented base on the Levenshtein distance methodology.

Throughout the work term, my goal was to find out the value between deploying machine learning models to a real-time production and building and training a machine learning model from nothing. For my personal development, it is very important for me to be able to handle project independently and be a part of the team instead of just been a student. I want to create models from scratch instead of being given a well-developed and doing the fill in the blank work. From the beginning of the work term, I was mainly focused on using my software development skill to automated machine learning tasks that were done manually in the team. At the same time, my team also include me in all kinds of machine learning design and model training meetings. I started my first machine learning task with data analyzing, because data is the heart of machine learning, and personally I believe that data can tell you more than you can ever imagine. Throughout the data analyze task, I build my first machine learning model, it is an unsupervised learning with the K-Means algorithms. I was trying to find out the relationship between the amount of money a customer spends, and the customer’s credit card limit. The interesting insight was that the people with the medium credit card limit spend the most money. It is not a linear relationship where the higher a customer’s credit card limit, the more many they spend each month. On the other hand, it is acutally the people with a medium credit card limit spending more money than people with a high credit card limit. Thus, the purpose of this report is to analyze the different machine learning algorithms’ approach and effect to the same business use case. Then, the report will give recommendation on how to approach a machine learning use case and what algorithms to use.

# Summary

The main purpose of the report is to document the research and analysis of the two-different supervised machine learning algorithms used in this business use case. Namely, Linear Regression, and Random Forest Regression. In the end, the report will provide recommendations for companies on which algorithm is most suitable for this use case depending on the accuracy, cost and response time.

The report will first introduce the general concept of machine learning, and the types of machine learning. Then, there will be an introduction to the business use case that we are trying to use machine learning to solve. The third section will analyze the raw data that is going to be used to train the machine learning model. In the fourth section, the report will formally provide the readers the engineering problem that the team is trying to solve. This report will present two different solutions that have been implemented on a smaller scale with different machine learning algorithms but trained on the same data. In the end, from a different perspective, we will analyze those two solutions and provide a recommendation for the production.

The major conclusions in this report will discuss the two different solutions that team can use to approach a business use case and process the customer data. It will also compare the two-different solution’s performance, cost, and response time. In the end, the report will also make recommendations on the type of machine learning algorithm to use for solving a business use case.

The main purpose of this report is to analyze and document all the learning and decisions from working with different machine learning algorithms for solving the same problem. From a different perspective, it will show how each machine learning algorithm performs. After analyzing and comparing the two solutions, the report will make a recommendations on what kind of how should a machine learning team approach business use case.

Table of Contents

[Contributions iii](#_Toc503221285)

[Summary v](#_Toc503221286)

[List of Figures vii](#_Toc503221287)

[List of Table viii](#_Toc503221288)

[1 Introduction 1](#_Toc503221289)

[1.1 What is Machine Learning 1](#_Toc503221290)

[1.1.1 Concept introduction 1](#_Toc503221291)

[1.2 Types of Machine Learning 2](#_Toc503221292)

[1.2.1 Supervised Learning 2](#_Toc503221293)

[1.2.2 Unsupervised Learning 3](#_Toc503221294)

[1.2.3 Semi-supervised learning 4](#_Toc503221295)

[1.2.4 Reinforcement learning 4](#_Toc503221296)

[2 Project Scope 5](#_Toc503221297)

[2.1 Project background 5](#_Toc503221298)

[2.2 Development plan 5](#_Toc503221299)

[2.3 Analysis of the available data 5](#_Toc503221300)

[3 The engineering problem 6](#_Toc503221301)

[4 Different Machine Learning Algorithms Solutions 7](#_Toc503221302)

[4.1 Solution 1: Linear regression 7](#_Toc503221303)

[4.2 Solution 2: Random forest 9](#_Toc503221304)

[5 Engineering Analysis 11](#_Toc503221305)

[6 Conclusions 13](#_Toc503221306)

[7 Recommendations 14](#_Toc503221307)

[Glossary 15](#_Toc503221308)

[References 16](#_Toc503221309)

# List of Figures

[Figure 1. Supervised Learning 1 3](#_Toc503318120)

[Figure 2. Unsupervised Learning 1 4](#_Toc503318125)

[Figure 3. ERD of available datasets 1 6](#_Toc503319072)

[Figure 4. Distribution of prediction 1 8](#_Toc503319078)

[Figure 5. Distribution of RF prediction 1 10](#_Toc503319085)

# List of Table

[Table 1 Types of Machine Learning 2 2](#_Toc503319151)

[Table 2. Decision matrix weight 1 7](#_Toc503319159)

[Table 3. Summarized result for LR 1 9](#_Toc503319162)

[Table 4. Scoring matrix for LR 1 9](#_Toc503319166)

[Table 5. Summarized results for RF 1 11](#_Toc503319170)

[Table 6. Scoring matrix for RF 1 11](#_Toc503319173)

[Table 7. Final Decision Matrix 1 11](#_Toc503319177)

1. Introduction

Machine Learning (ML) is a multidisciplinary interdisciplinary research involving many disciplines such as probability theory, statistics, approximation theory, convex analysis and algorithm complexity theory [1]. It specializes in how computers simulate or realize human learning behaviors in order to acquire new knowledge or skills and to reorganize existing knowledge structures to continuously improve the Machine Learning model’s performance.

It is the core of artificial intelligence. In order to make the computer has the fundamental way of intelligence and to make its application spread throughout all areas of artificial intelligence, it mainly use induction, synthesis rather than deduction.

What is Machine Learning

In 1950’s, Arthur Samuel defined machine learning as the field of study that gives computers the ability to learn without being explicitly programmed. For a more modern definition, Tom Mitchell defined machine learning as “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P. if its performance at tasks in T, as measured by P, improves with experience E.”. Basically, machine learning refers to letting the machine learn from a large number of examples, rather than telling him what to do in the form of conventional rules of writing. In simple terms, in traditional program development, you need to write a lot of rules to tell the computer how to solve a specific problem. For machine learning, what you write is a specific set of algorithms that allow computers to discover these rules for the engineer, and then solve them based on these rules.

Concept introduction

In order to understand Tom Mitchell’s definition of machine learning, the most common used example is the checkers playing program. The checkers playing program learns over time what are good board positions and what are bad board positions. And eventually learn to play checkers better than a professional player was able to. For the checkers playing example [1] the experience e, will be the experience of having the program play 10's of 1000's of games against itself. The task t, will be the task of playing checkers. And the performance measure p, will be the probability that it wins the next game of checkers against some new opponent. Thus, for each additional game, which is a task, the checkers playing program plays, the checkers playing program has a higher chance of winning, which is an increase in performance P, so the checkers playing program is improved by the experience of playing many games, which is E. In the checkers playing program example, the computer is learning for the past experience and improving the accuracy with a clear goal. This is known as supervised learning, and the idea is that the engineer is going to teach the computer how to do something.

Types of Machine Learning

There are often some variations on how to define and spate the different types of machine learning algorithms. Most commonly, they are divided into four different categories according to their purpose and the main categories are listed in Table 1:

*Table 1 Types of Machine Learning 2*



Supervised Learning

In supervised learning, the data set is given to the engineer and the engineer already know what the correct output should look like, with that idea in mind the algorithm is able to establish a relationship between the input and the output.

Supervised learning problems are often problems related to “regression” and “classification”. For a regression problem, the output of the machine learning algorithms that the engineer are using to predict the result of problem is within a continuous output. In a mathematical point of view, the engineer is trying to map input variables to some continuous function. On the other hand, in a classification problem, the predicted result is often binary and must be a discrete output. In other words, the engineers are trying to map input variables of the classification problem into discrete categories. In Figure 1, it is a flowchart of how supervised learning is being done.

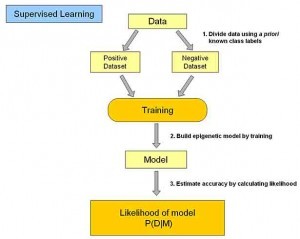


Figure 1. Supervised Learning

Unsupervised Learning

In unsupervised learning, the main difference is that given a data set to the engineer but the engineer does not have any information about our purpose and correct output. Unsupervised learning allows the engineer to approach problems with little or no idea what our results should look like, and provide the engineers many interesting points about the datasets as shown in Figure 2. The engineer can derive structure from data where the engineer doesn’t necessarily know the effect of the variables. In order to derive this structure, the engineer often clusters the data based on relationships among variables in the data.

Basically, the learning models are designed to infer some of the internal structure of the data. Common application scenarios include association rules learning and clustering. Common algorithms include: Apriority algorithm and k-Means algorithm.

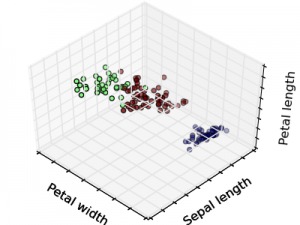


Figure 2. Unsupervised Learning

Semi-supervised learning

In semi-supervised learning, the input data part is identified, some are not identified, this learning model can be used to make predictions, but the model first needs to learn the inherent structure of the data in order to reasonably organize the data to be predicted. The application scenarios include classification and regression. The algorithm includes some extensions to common supervised learning algorithms that attempt to model unlabeled data first and then predict the identified data. Such as Graph Inference or Laplacian SVM.

Reinforcement learning

In Reinforcement learning, the input data as a feedback to the model, unlike the supervisory model, the input data is only used as a way to check the if model is right and wrong, under reinforcement learning, the input data is fed back directly to the model, the model must make adjustments immediately. Common application scenarios include dynamic systems and robotics control. Common algorithms include Q-Learning and Temporal difference learning.

1. Project Scope

Project background

The customer is one of the big banks in Canada, and the team was working with their data science team in building a machine learning model that will find out the customers with a higher potential in signing up for a credit card from a list of customers. This is a supervised learning problem, the team was given data on the customers that signed up for a credit card and was contacted by the bank in the past 90 days. Thus, the team want to build a classifier that can identify if given a new customer’s personal data, does that customer worth the bank spending time to contact.

Development plan

After deciding the business use case, the team decide to separate this big project into smaller pieces. Due to the limited data available, about 50% of the labeled data are missing some important data, so instead of building a simple classifier that is trained on only the data that are complete, the team decided to build a few smaller regression models that will fill the missing data.

The feature that I was responsible for building a regression model is the yearly income of the customers. Income is an very important feature that must be included in the training data, because depending on the customer’s yearly income, the model will have a better idea of whether if this customer can afford using a credit card. Income data also give the engineer information about the amount of credit that customer can handle, on the other hand it also allows the engineer to have a better guess of whether that customer already have a credit card with another bank.

Analysis of the available data

After understanding the problem that the team are trying to solve, the next step is to understand the available data and make the most out of it. There are 5 different data tables in total, and they are linked together by the region id as shown in Figure 3.

Since the region table is the center of all the datasets, it is important to use as many features from that table as possible. First of all, the table tells the engineer the total population within the region, at the same time, it also tells the team the amount of men and women, so this gives a good ratio of the gender’s effect for the region based average salary table.

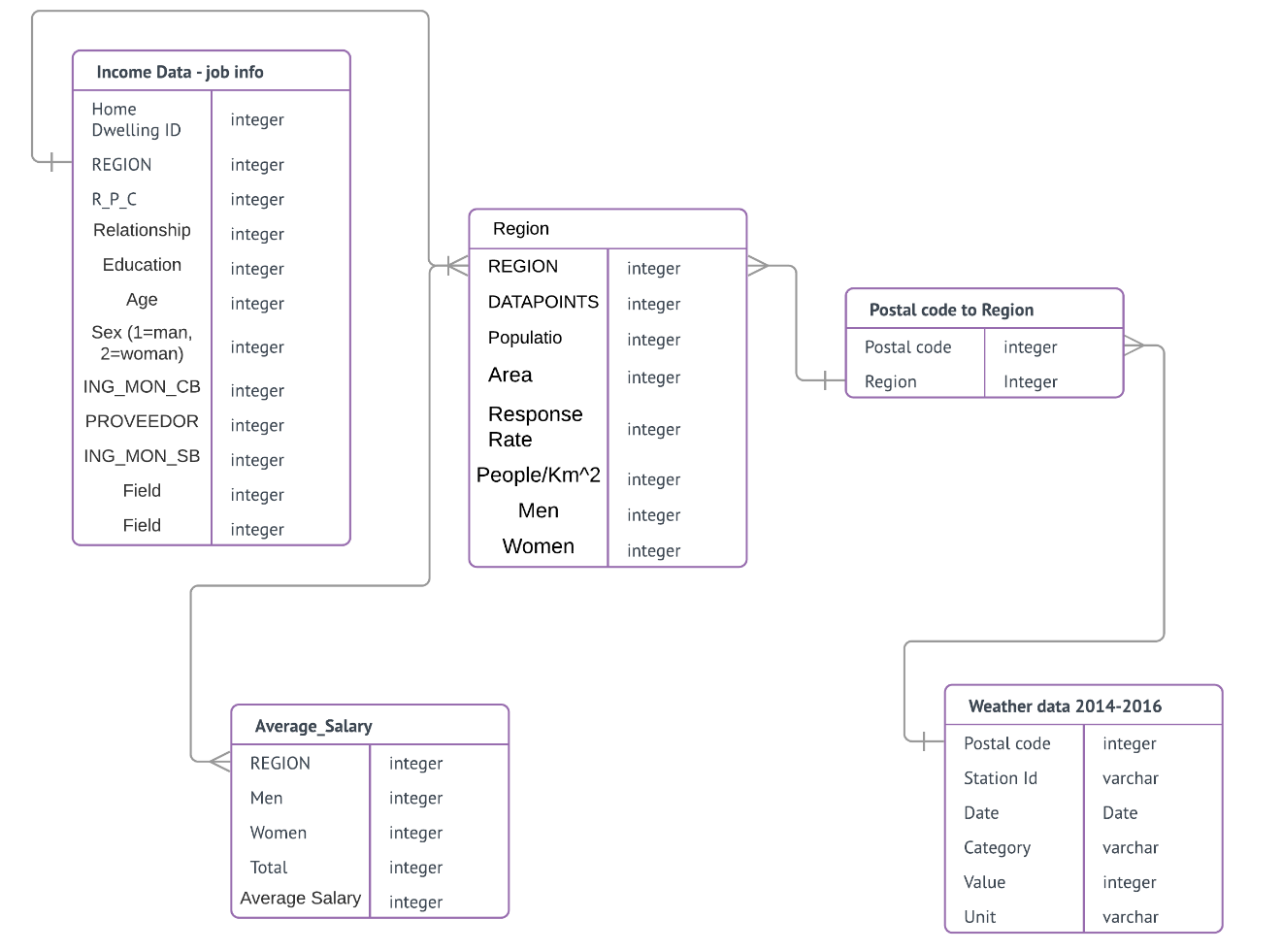


Figure 3. ERD of available datasets

1. The Engineering Problem

The project intended to apply different supervised machine learning algorithms to the same business use case to found out the most accurate and efficient solution for this machine learning use case. Most importantly, since each machine learning algorithms provide a different approach in solving the same problem, spending the time to implement each solution at a small scale is actually providing the team more knowledge about the customer’s data and the business use case.

This machine learning model is trying to predict a person’s yearly income given one’s age, sex, region and many other features as input, and a finite integer as output. The purpose of the model is to fill the missing income data for the customers in the whole data universe that do not have income information. The two machine learning algorithms that is going to be implemented and tested are linear regression, and Random forest regression. The final decision will be based on comparing the performance, the cost of implementation and the response time for each of the machine learning algorithms following the weight in Table 2.

Table 2. Decision matrix weight

|  |  |
| --- | --- |
| Category | Weight |
| Cost | 20% |
| Response Time | 20% |
| Performance | 60% |

1. Different Machine Learning Algorithms Solutions

Supervised learning is a machine learning task that is trying to infer a function from the labeled training data. Different machine learning algorithms are different ways to process the same data. For example, linear regression is trying to fit a line to all the data points, where the random forest is based on decision tress. In the following section, the report will present two solutions that used different machine learning algorithms, and the report will provide statics on the performance of each algorithms.

Solution 1: Linear regression

Regression models are used to test if a relationship exists between variables; that is, to use one variable to predict another. However, there is some random error that cannot be predicted. There is a simple formula to represent the basic relationship between the input and output in the linear regression model [2].

The Simple Linear Regression formula: Y = β0+ β1X + error

Where,

Y = dependent variable (response)

X = independent variable (predictor / explanatory)

β0= intercept (value of Y when X = 0)

β1= slope of the regression line

Error = random error

Sample data or labeled data are used to estimated the true values for the intercept and slope of the formula. Using the sample data, Y = b0 + b1X, and Y = predicted value of Y. The difference between the actual value of Y and the predicted value (using sample data) is known as the error.

So, Error = (actual value) – (predicted value)

The first solution has used linear regression to build the machine learning model. The total count of all the available data is 101050. During the data preparation process, the whole dataset has been randomly split with a 70% and 30% ratio, which means that the model is trained on 70% of the available data, and being evaluated on the rest 30%. Training the linear regression model took around 20s and evaluating the model took around 10s.

The resultant coefficients of the model were: [5134.983665365605, -196738.[8587523364](tel:(858)%20752-3364),

-5375.06711782812, -4161.871545497067,0.0, -32675.9606318151]

Intercept: 1153608.7303823817

From the cross-validation result, the percentage that the linear regression model was able to successfully predict the income of a person is 40%. The result means that out of 30000 people there were 15000 people’s income got successfully predicted. In Figure 4, it is clear that the result is better than random, because excluding all the outliers, the model was able to successfully predict most of the middle-class people’s income. After analyzing all the important perspectives, the summarized results are show in Table 3.

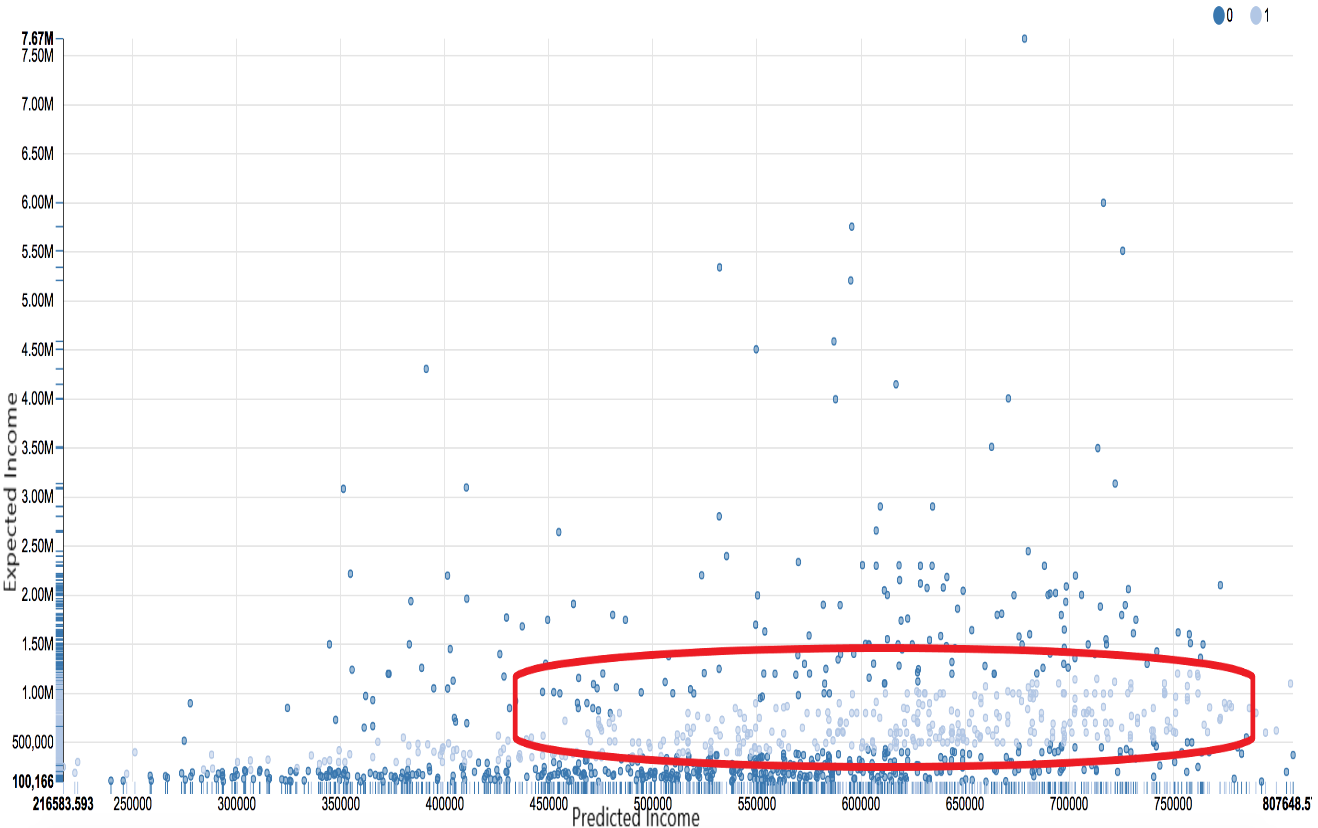


Figure 4. Distribution of prediction

Table 3. Summarized result for LR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Performance: Precision | Model training time | Model evaluation time | Cost: Algorithm difficulty | Cost: Data preparation |
| 40% | 15s | 10s | 4 | 6 |

From the summarized table, the linear regression model received a score of 4.0 for performance a score of (10-4) \*(10-6)/10 = 2.4 for cost, and a score of (10 – 1.5) \*(10-1)/10 =7.7 for response time. Thus, the final score for random forest model is show in Table 4.

Table 4. Scoring matrix for LR

|  |  |
| --- | --- |
| Linear Regression | Score |
| Performance | 4.0 |
| Cost | 2.4 |
| Response Time | 7.7 |

Solution 2: Random Forest Regression

One of the most fundamental and popular method for various machine learning task is decision trees. Tree learning “comes closest to meeting the requirements for serving as an off-the-shelf procedure for data mining”, say Hastie et al. [5]. Decision tree is invariant under scaling and various other transformations of feature values, and it is robust to inclusion of irrelevant features, and produces inspectable models. Random Forest is based on having multiple decision trees, and ensemble those tree’s result to get a regression value.

Given a standard training set D of size n, bagging generates m new training sets D\_i, each of size n′, by sampling from D uniformly and with replacement. This kind of sample is known as a bootstrap sample. The m models are fitted using the above m bootstrap samples and combined by averaging the output (for regression) or voting (for classification).

Random Forest Algorithm [3]:

Given a training set with n samples {X，Y}

for b=1, …, B:

1.From X there are n samples returned to form a collection {Xb, Yb};

2. In {Xb, Yb} train the decision tree (or regression trees);

End

After the training is done, take the average of all models as output (or vote with the majority vote).

The second solution has used random forest regression to build the machine learning model. The total count of all the available data is still the same, which is 101050. During the data preparation process, the team decided keep the 70% and 30% ratio of training data, and testing data. Training the linear regression model took around 40s and evaluating the model took around 10s. The model has set a limit of 10 trees in the random forest model.

From the cross-validation result, the percentage that the linear regression model was able to successfully predict the income of a person is 75%. The result means that out of 30000 people there were 21500 people’s income got successfully predicted. In Figure 5, it is clear that the result is better than linear regression, because excluding all the outliers, the model was able to successfully predict most of the middle-class people’s income and the lower-class people’s income. After analyzing all the important perspectives, the summarized results are show in Table 5.

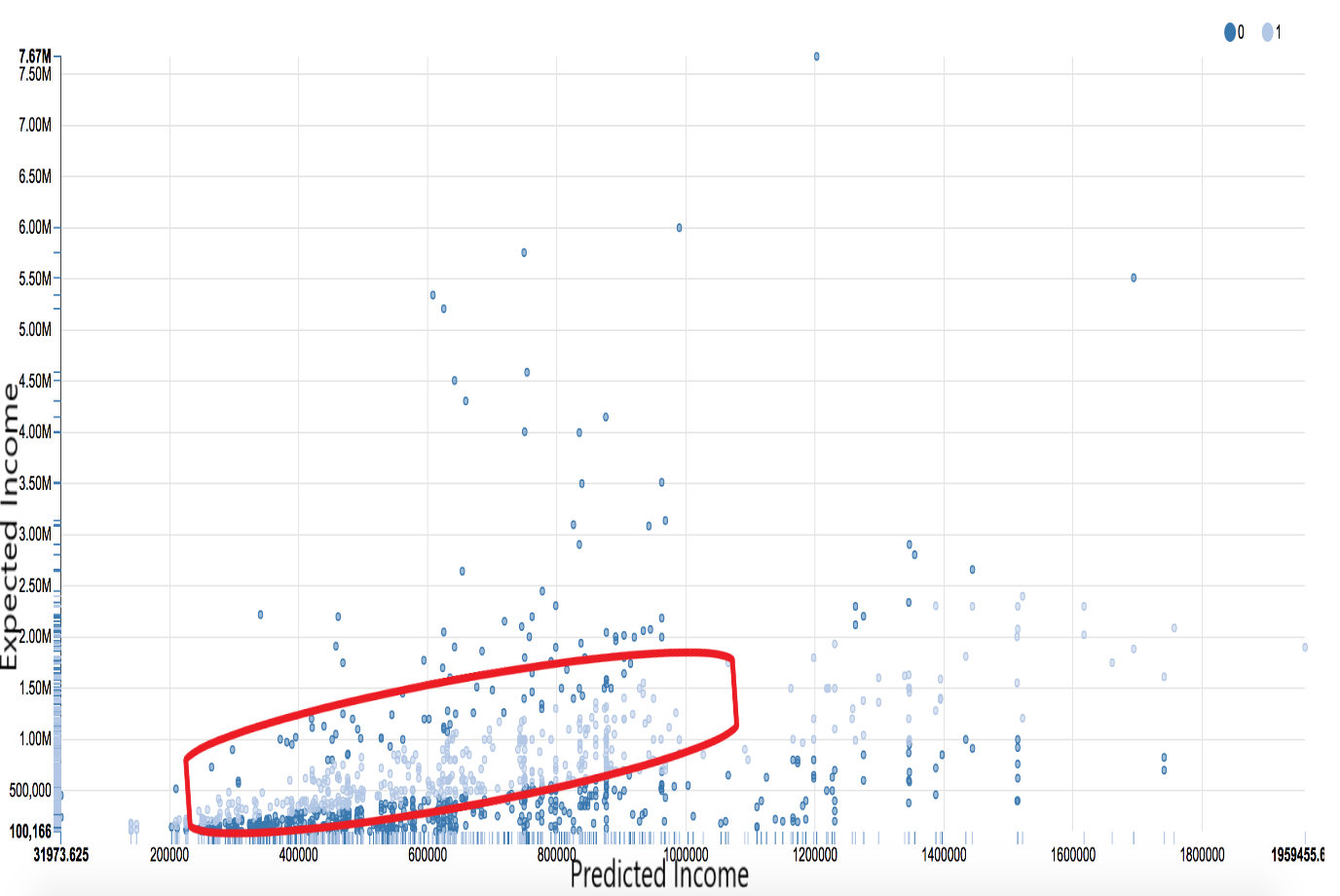


Figure 5. Distribution of RF prediction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Performance: Precision | Model training time | Model evaluation time | Cost: Algorithm difficulty | Cost: Data preparation |
| 75% | 40s | 10s | 8 | 6 |

Table 5. Summarized results for RF

From the summarized table, the random forest regression model received a score of 7.5 for performance a score of (10-8) \*(10-6)/10 = 0.8 for cost, and a score of (10 – 4)\*(10-1)/10 =5.4 for response time. Thus, the final score for random forest model is show in Table 6.

Table 6. Scoring matrix for RF

|  |  |
| --- | --- |
| Random Forest | Score |
| Performance | 7.5 |
| Cost | 0.8 |
| Response Time | 5.4 |

1. Engineering Analysis

The analysis of the solutions based on the cost, performance and response time. For building a machine learning model, the model should have error as low as possible, and its accuracy is what matter the most. Thus, from the final decision matrix in Table 7, because random forest model had a significant better result and a higher accuracy [4], it is a better a solution to process the data for the income predictor. Although the linear regression had a lower cost and better response time, in this situation it did not affect the final decision due to the company’s main focus and the basic guideline of machine learning model.

Table 7. Final Decision Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cost | Performance | Response Time | Total |
| Solution1: Linear Regression | 2.4-> 4.8 | 4-> 24 | 7.7-> 15.4 | 44.2 |
| Solution2: Random Forest Regression | 1.2-> 2.4 | 7.5-> 45 | 5.4-> 10.8 | 58.2 |

On the other hand, if the team were to evaluate the two machine learning models in general, both algorithms are good solutions to regression problems. Solution 1, linear regression, is a good place to start for any machine learning problem. It is very easy to understand and implement, and it often works just as well for most of the purposes. Through the process of developing a linear regression model, the machine learning engineer will have a better understand of the raw data. On the other hand, the sample linear regression model will present many problems that the engineer has not thought about, which allows the engineer to choose a more suitable algorithm. Solution 2, random forest regression, is a well-developed and refined algorithm that the industry have been using for a long period of time. It is harder to understand comparing to the linear regression model, because it is based on decision trees with many other optimization methods. However, all the work pays off and with a higher cost and a higher response time [5], the performance of the model is much better than the linear regression.

1. Conclusions

According to the analysis, solution 1 is clearly easy to understand and implement. At the same time, it also has a faster response time in both training and testing because of its simple approach to process the data. However, the result of solution 1 was not acceptable by the industry standard. With a 40% accuracy, it is worse than random in a general sense, and only slightly better than random when the outliers are removed. On the other hand, solution 2 has a difficult math content to understand and the implementation require special engineers to handle all the parameter tuning. However, the result of solution 2 is higher than the industry standard and profitable for the customers.

From the analysis in the report, it was concluded that solution 2 implementing the income predictor with a random forest regression is the better choice. From the final decision matrix (Section 5, Table 7), the solution with random forest regression is a more sophisticated solution, because it had an impressing performance and an acceptable cost and response time. It satisfies all design key points and requirements. The solution works well with qualitative (categorical) features, while also handles multiple features that may be correlated.

# 7 Recommendations

Based on the analysis and conclusions in this report, it is recommended that the machine learning team should use random forest regression for the income predictor. The team can start solving the business use case with a simple machine learning algorithm like simple linear regression. From the cost point of view, starting with a simple machine learning algorithm is very easy for an engineer. After understanding the uniqueness of the raw data, and researched over the problems that occurred in linear regression, the machine learning engineer should choose a more sophisticated like random forest regression to proceed to the actual model training and testing. This saves the team time in debugging, any data bug will be easier to realize and solve in a linear regression model comparing to the random forest regression.

However, performance and accuracy is always the most important factor for any machine learning algorithms, so despite the cost and the response time of linear regression, it is not a good algorithm for a production model. Thus, the team should use random forest regression to build a machine learning model that is invariant under scaling and various other transformations of feature values, and robust to inclusion of irrelevant features.

# Glossary

**Labeled data**: A group of samples that have been tagged with one or more labels. Labeling typically takes a set of unlabeled data and augments each piece of that unlabeled data with meaningful tags that are informative

**Decision Tree**: A decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

**Algorithm**: A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

**Spark:** an open-source cluster-computing framework. Originally developed at the University of California, Berkeley's AMPLab, the Spark codebase was later donated to the Apache Software Foundation, which has maintained it since.

**Scala:** a general-purpose programming language providing support for functional programming and a strong static type system. Designed to be concise, many of Scala's design decisions aimed to address criticisms of Java.

**Feature：**a distinctive attribute or aspect of something.

# Bibliography

|  |  |
| --- | --- |
| [1] | A. Ng, "Machine learning," Coursera, [Online]. Available: https://www.coursera.org/learn/machine-learning/supplement/aAgxl/what-is-machine-learning. |
| [2] | D. X, "Machine Learning Methods," Blog CSDN, [Online]. Available: Machine Learning Methods (6): Random Forest Random Forest, bagging. |
| [3] | A. Olteanu, "Learning Curves for Machine Learning," Dataquest, [Online]. Available: https://www.dataquest.io/blog/learning-curves-machine-learning/. |
| [4] | T. Hastie, R. Tibshirani and Friedman, The Elements of Statistical Learning (2nd ed.).. |
| [5] | ctufts, "Classification Model Pros and Cons," Github, [Online]. Available: https://github.com/ctufts/Cheat\_Sheets/wiki/Classification-Model-Pros-and-Cons. |