11/27/2021

Predicting Rental Bike Rides with the Bike Sharing Dataset

Final Report

Amanda Rorabaugh | Nicholas Aryel Ferreira

Georgia Institute of Technology

Master of Science in Analytics

Contents

[Abstract 2](#_Toc88925005)

[Introduction 2](#_Toc88925006)

[Data Source and Problem Statement 2](#_Toc88925007)

[Proposed Methodology 3](#_Toc88925008)

[Exploratory Data Analysis 4](#_Toc88925009)

[Variable Selection 6](#_Toc88925010)

[Analysis and Results 7](#_Toc88925011)

[Multiple Linear Regression 7](#_Toc88925012)

[Poisson Regression 7](#_Toc88925013)

[LASSO 8](#_Toc88925014)

[Ridge Regression 8](#_Toc88925015)

[Elastic Net 9](#_Toc88925016)

[Decision Tree 10](#_Toc88925017)

[Random Forest 10](#_Toc88925018)

[AdaBoost 11](#_Toc88925019)

[Bagging 11](#_Toc88925020)

[Neural Networks 12](#_Toc88925021)

[Insights and Conclusions 12](#_Toc88925022)

[Lessons we have learned 14](#_Toc88925023)

Abstract *— Machine learning can be used for a variety of things. One of them is to predict information that we don’t know based on existing patterns on data that is available. More specifically, we will be looking on how machine learning can be used to predict the number of bike sharing rides based on several factors. This problem can help bike sharing companies in the complex task of balancing the number of bikes available with the most probable demand and help plan the availability of rental bikes according to different factors, such as season of the year, month, humidity, holidays, etc.*

# Introduction

The main purpose of this project is to evaluate the use of statistical methods that learn through experience (machine learning) to predict the number of bike sharing rides based on historical data from a bike-sharing company called Capital Bikeshare.

Graphical user interface

Description automatically generated with medium confidence

Figure 1: Capital Bikeshare

The project aims to compare such methods in different aspects such as predictability (R-Squared, Mean Squared Errors), efficiency (time to train the model), and best parameters obtained through cross-validation, as well as to make an in-depth analysis of the dataset to understand the predictive power of each explanatory variable, multicollinearity between variables and analysis of outliers.

The next sections of the report will cover the problem statement and the data source utilized in the analysis, methodology, an exploratory data analysis of the dataset, variable selection, the main analyses and results for each method utilized, final conclusions and insights from the analyses.

We hope that the results of this project will serve as a building block for bike-sharing companies in the understanding of the main factors that determine the demand for bike rides, as well as the best machine learning methods to predict this demand, so that companies can better allocate their resources and increase customer satisfaction.

# Data Source and Problem Statement

The dataset was obtained through the UC Irvine Machine Learning Repository on 9/15/2021 and is available at the website: [https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset#](https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset)

This dataset contains 17,379 datapoints with 17 attributes aggregated by hour from January 1st, 2011, to December 31st, 2012. The data is provided by Capital Bikeshare, a Washington DC based bike-share company that was opened in September 2010, and contains information such as:

• Time, day, season, year, month, hour, weekday

• Whether it is a holiday, whether it is a working day

• Temperature, humidity, windspeed

• Total number of rides during that hour broken into registered rides and casual rides

Table

Description automatically generated

Figure 2: dataset

The project will focus on the total rides (‘cnt’) as the response but will also present a comparison for casual and registered rides.

The analysis of this dataset is aimed to answer the following problem statement: *How do different machine learning algorithms perform and compare with one another in terms of predictability and efficiency in the task of predicting the number of bike rides for Capital Bikeshare?*

# Proposed Methodology

The following machine learning algorithms will be evaluated in this study:

* Multiple linear regression
* Elastic Net
* Ridge regression
* Poisson regression
* LASSO
* Decision Trees
* Random Forest
* Ada Boost
* Bagging
* Neural Networks.

Each method will be briefly described and evaluated in terms of the mean squared training and testing errors, R-Squared and training time. All methods, with the exception of Multiple Linear Regression will also contain a cross-validation analysis to find one or more best parameters for the model. Insights will then be derived from the analysis and discussed in the Insights and Conclusions section.

# Exploratory Data Analysis

An initial analysis of the distribution of daily bike rides reveals that both the total number of bike rides and registered bike rides present a **symmetric distribution** that resembles a normal curve, while the distribution of the number of casual rides is **heavily skewed** to the right because of the irregularity in the frequency of casual rides over time:

Chart, histogram

Description automatically generated

Figure 3: bike rides distribution

The analysis of the number of bike rides over time reveals that the demand is usually higher in the **middle of the year**, corresponding to the seasons of summer and fall, for both registered and casual riders:

Chart

Description automatically generated

Figure 4: distribution of the bike rides by season and by month

Then, Cook’s distance was utilized to find the outliers in the hourly dataset, which estimates the influence of each data point, against the ‘instant’ attribute which serves as an identifier field for the data.

There were 4 outliers identified using Cook’s distance for the dataset aggregated by hour. This is a low number considering a dataset of 17,380 observations and thus do not have a significant effect on the analysis.

In the dataset aggregated by day, there was an outlier associated to the day when Washington DC was in a state of emergency due to Hurricane Sandy. On this day, there were only 22 bike rides.

Chart

Description automatically generated

Figure 5: outlier analysis

Another analysis was performed to understand the correlation between variables. The correlation matrix reveals a strong correlation between the variables month (mnth) and season, as well as atemp and temp. The data also suggests a **moderate correlation** between the temperature (temp, atemp) and the number of rides (casual, registered, cnt).

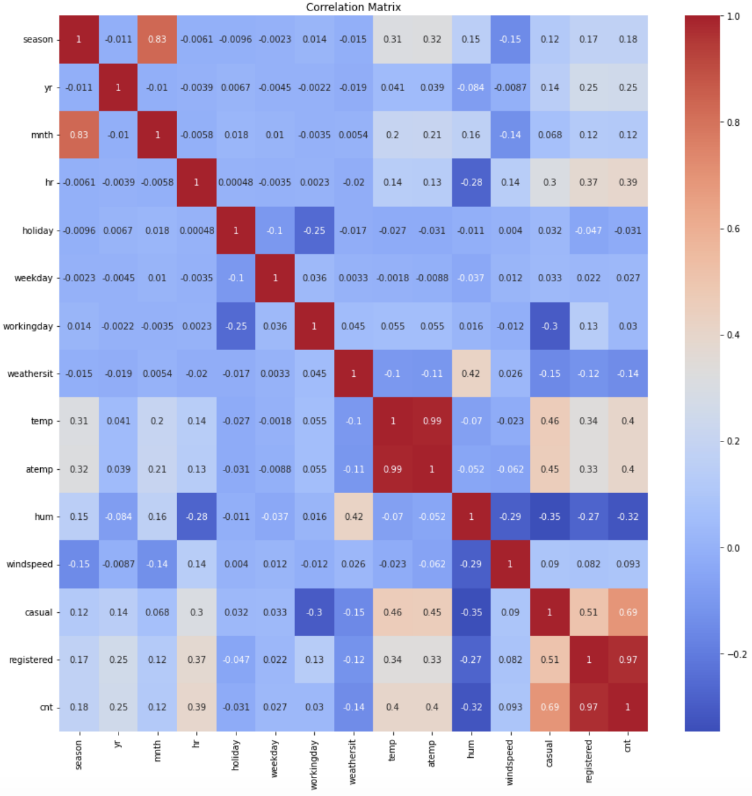


Figure 6: correlation analysis

This relationship between temperature and number of rides was further investigated through a scatter plot for each of the response variables (total, registered, casual):

Chart, scatter chart

Description automatically generated

Figure 7: temperature x rides

As the scatter plots show, the relationship between temperature and the response variable is not perfectly linear, but it is directly proportional as shown by the right upward direction of the datapoints. The casual rides present an interesting behavior in which the relationship seems to be exponential across the temperature values.

# Variable Selection

Five methods of variable selection were utilized to determine which variables significantly contribute to the prediction of the total number of bike rides in each hour:

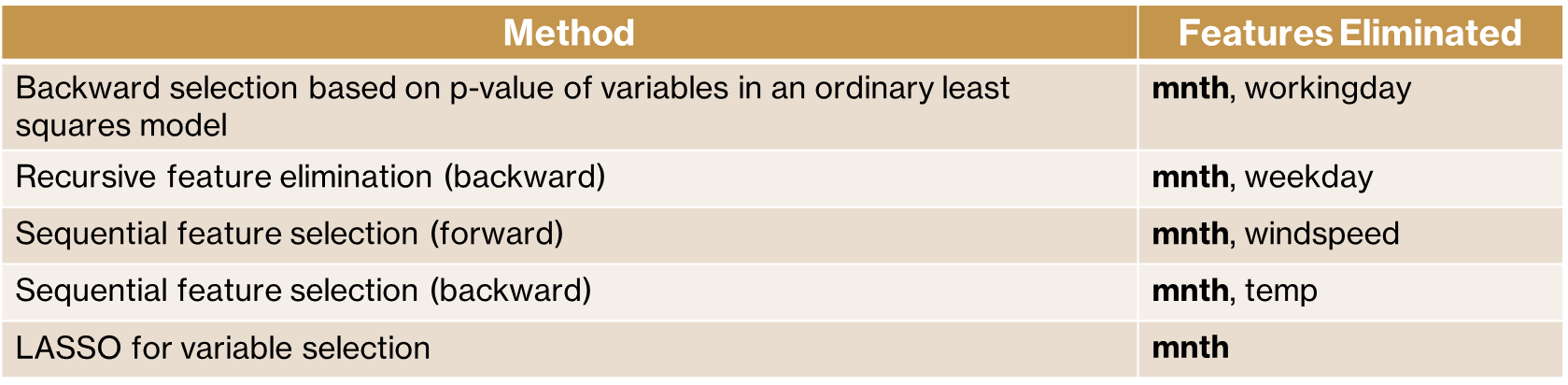


Table 1: variable selection methods

Having considered the results from all methods, the variable "mnth" (month) is the only variable eliminated in all of them, therefore, it will not be used in the models.

As an example of the implementation of a variable selection method, the LASSO path shows that the **variable ‘month’ should be the first to be eliminated** from all other variables as seen in the chart below:

Chart

Description automatically generated

Figure 8: LASSO path

The LASSO path shows the most significant variables according to the regularization parameter. Notice that for higher values of the parameter, the variable “month” is the last to be selected (in red).

Another example is the backward selection based on p-value. Fitting an ordinary least squares model and using manual backward selection based on p-value eliminated “mnth” (0.999 p-value) and “workingday” (0.368 p-value). Having eliminated these two variables, the resulting model presents only statistically significant explanatory variables (using an alpha of 0.05):

Table

Description automatically generated

Table 2: OLS variable selection final model

# Analysis and Results

Having selected the explanatory variables, the dataset was transformed into a 2D array and randomly divided into a training set (80%) and testing set (20%) for the explanatory variables and response (total bike rides).

## Multiple Linear Regression

The multiple linear regression model fits a linear model through the method of least squares to minimize the residual sum of squares.

An initial analysis was made to test the assumptions for the regression model:

Chart, histogram

Description automatically generated

Figure 9: Regression model initial analysis

The normal probability plot and the histogram of residuals show that the residuals from the model are approximately normal. The residuals vs observations chart also shows that the variance is constant across the observations.

The model obtained a mean squared training error of 19978.397, a mean squared testing error of 20709.979, a R-Squared of 0.708 and the overall training time was 0.0087s.

The linear regression model presented a medium performance on R-Squared, and a shorter training time compared to other models. Its testing error, however, is relatively high compared to the ensemble methods that will be shown in the next sections.

## Poisson Regression

Poisson Regression is a generalized linear model which assumes that the response variable has a Poisson distribution.

A 5-fold cross-validation was performed to assess the best parameter alpha according to the CV score from the GridSearchCV function:

Chart, scatter chart

Description automatically generated

Figure 10: Poisson Regression cross-validation

The chart shows that the score peaks when alpha is 0.21. The model obtained a mean squared training error of 20373.849, a mean squared testing error of 20752.959, a R-Squared of 0.386 and the overall training time was 0.0764s.

Poisson Regression performed worse than most models in terms of testing error and R-Squared. Its training time is relatively short. One possible explanation is the fact that the number of bike rides per hour does not properly follow a Poisson distribution.

## LASSO

In this model, some variables have their coefficient set to zero while others are preserved through the Least Absolute Selection and Shrinkage Operator (LASSO). It works well for when the true model is sparse.

A 5-fold cross-validation was performed to assess the best regularizer parameter according to the CV score from the GridSearchCV function:

Chart, scatter chart

Description automatically generated

Figure 11: LASSO cross-validation

The chart shows that the score peaks when the regularizer parameter is 0.0034. The model obtained a mean squared training error of 19963.211, a mean squared testing error of 20693.294, a R-Squared of 0.387 and the overall training time was 0.0235s.

The LASSO model presented a relatively low R-Squared and high testing error compared to other models. Its training time, like the other regression models, is relatively short.

## Ridge Regression

Ridge Regression is a type of regression that uses regularization to reduce variance, shrinking the regression coefficients towards 0, which also adds some bias to the model.

A 5-fold cross-validation was performed to assess the best regularizer parameter according to the CV score from the GridSearchCV function. The chart below shows the Ridge coefficients of the model as a function of the regularization parameter alpha:

Chart

Description automatically generated

Figure 12: Ridge Regression coefficients

The model obtained the best alpha of 1.743 in the cross-validation. It also obtained a mean squared training error of 19963.715, a mean squared testing error of 20691.323, a R-Squared of 0.387 and the overall training time was 0.0123s.

Like LASSO, Ridge Regression presented a relatively low R-Squared and high testing error compared to other models. Its training time, like the other regression models, is relatively short.

## Elastic Net

Elastic Net combines the variable selection benefits of LASSO with the prediction benefits of Ridge Regression. The alpha parameter for Elastic Net determines how much weight is given to two types of penalties: L1 and L2. An alpha value of 0 would give all weight to the L2 penalty, and alpha of 0.5 would split the weight evenly, and an alpha of 1 would give all weight to the L1 penalty.

A 5-fold cross-validation was performed to assess the best alpha parameter according to the CV score from the GridSearchCV function:

Chart, scatter chart

Description automatically generated

Figure 13: Elastic Net cross-validation

The chart shows that the score peaks when the alpha is 0.0003. The model obtained a mean squared training error of 19963.876, a mean squared testing error of 20691.132, a R-Squared of 0.387 and the overall training time was 0.0098s.

Like LASSO and Ridge Regression, Elastic Net presented a relatively low R-Squared and high testing error compared to other models. Its training time, like the other regression models, is relatively short.

## Decision Tree

The Regression Decision Tree was utilized, which is an intuitive model that predicts a response based upon the partition of the feature space into several regions, which can be thought as a special case of additive model.

A 5-fold cross-validation was performed to assess the best complexity parameter according to the CV score from the GridSearchCV function:

Chart, scatter chart

Description automatically generated

Figure 14: Decision Tree cross-validation

The chart shows that the score peaks when the complexity parameter is 3.4. The model obtained a mean squared training error of 1826.529, a mean squared testing error of 3262.878, a R-Squared of 0.903 and the overall training time was 0.5365s.

The Decision Tree model performed better than the previous models in terms of R-Squared and testing error. Although the training time is higher than the previous models, it is still much shorter than the training time for the ensemble methods and neural networks which will be seen in the next sections.

## Random Forest

The Random Forest for regression was utilized, which is an ensemble of regression decision trees that predicts based on the average results from different trees. This is also considered a “black box” method, where the individual trees are not visible, however, it reduces the variance to provide better predictions.

A 5-fold cross-validation was performed to assess the best number of trees according to the CV score from the GridSearchCV function:

Chart, scatter chart

Description automatically generated

Figure 15: Random Forest cross-validation

The chart shows that the score peaks when the number of trees is 160. The model obtained a mean squared training error of 288.226, a mean squared testing error of 1774.569, a R-Squared of 0.947 and the overall training time was 5.0749s.

The Random Forest was the best model in terms of R-Squared and testing error, along with Bagging which will be seen in the next sections. Its training time, however, is relatively high compared to the other models.

## AdaBoost

AdaBoost is an ensemble algorithm that uses several weak learners to predict. Each weak learner performs slightly better than random guessing.

The AdaBoost Algorithm re-weighs training data after adding each weak learner in order to focus more on the data that previous weak learners have misclassified. Unlike random forest, AdaBoost creates a “forest of stumps” comprised of a node and two leaves. AdaBoost has a learning rate parameter that determines the contribution of each of these stumps to the weights.

A 5-fold cross-validation was performed to assess the best learning rate according to the CV score from the GridSearchCV function. The number of estimators (trees) was set as 160 from the CV results with Random Forest:

Chart, scatter chart

Description automatically generated

Figure 16: AdaBoost cross-validation

The chart shows that the score peaks when the learning rate is 3.1. The model obtained a mean squared training error of 10503.712, a mean squared testing error of 10449.236, a R-Squared of 0.691 and the overall training time was 1.3349s.

The AdaBoost algorithm obtained a medium performance in terms of R-Squared, testing error and training time. The predictability was better than other models such as Ridge Regression and LASSO, but lower compared to the other ensemble methods, decision tree, and neural networks.

## Bagging

Bootstrap aggregating (Bagging) is an ensemble method that generates many bootstrapping samples reusing the data, and for each sample it builds a distinct tree, providing a final prediction as the average across the predictions of all trees.

It attempts to decrease variance that is prevalent in tree-based methods by using one set of training data to create many bootstrapping samples, in which each sample will be utilized to build a distinct tree.

A 5-fold cross-validation was performed to assess the best number of base estimators according to the CV score from the GridSearchCV function:

Chart, scatter chart

Description automatically generated

Figure 17: Bagging cross-validation

The chart shows that the score peaks when the number of base estimators is 90. The model obtained a mean squared training error of 294.090, a mean squared testing error of 1797.511, a R-Squared of 0.947 and the overall training time was 2.9447s.

The bagging algorithm had the best performance in terms of R-Squared along with the Random Forest model. It also provided a low testing error, but a high training time compared to other methods.

## Neural Networks

A multi-layer perceptron regressor (MLP Regressor) was utilized, which learns a non-linear function for regression using layers of perceptrons (neurons), where each layer receives input values from the previous layer and transforms them into output values.

Since there are many parameters to assess in cross-validation for Neural Networks, a different approach was taken. Cross-validation was performed across different types of parameters in a total of 48 combinations:

Graphical user interface, text

Description automatically generated

Figure 18: Neural networks cross-validation parameters

The cross-validation obtained the following best parameters: activation: relu, alpha: 0.0001, hidden layer sizes: (50,50,50), learning rate: constant, solver: adam. The model also obtained a mean squared training error of 2456.847, a mean squared testing error of 2443.998, a R-Squared of 0.928 and the overall training time was 11.5224s.

Neural Networks obtained a high performance in terms of R-Squared and testing error, although its training time is the highest compared to all other models.

# Insights and Conclusions

The results of all models predicting the total number of bike rides (“cnt”) is summarized in the table below:

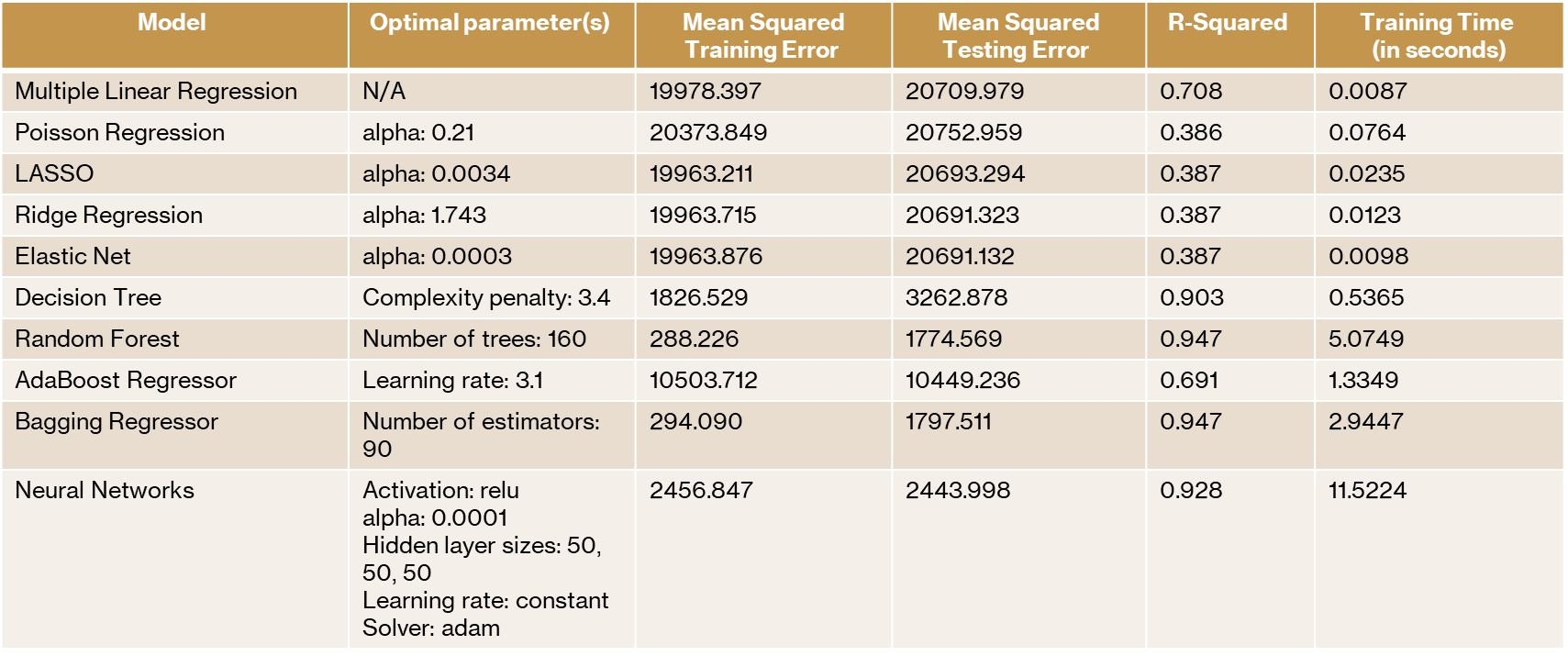


Table 3: main results – total rides

The same models were trained using the registered bike rides (“registered”) instead of total rides and obtained the following results:

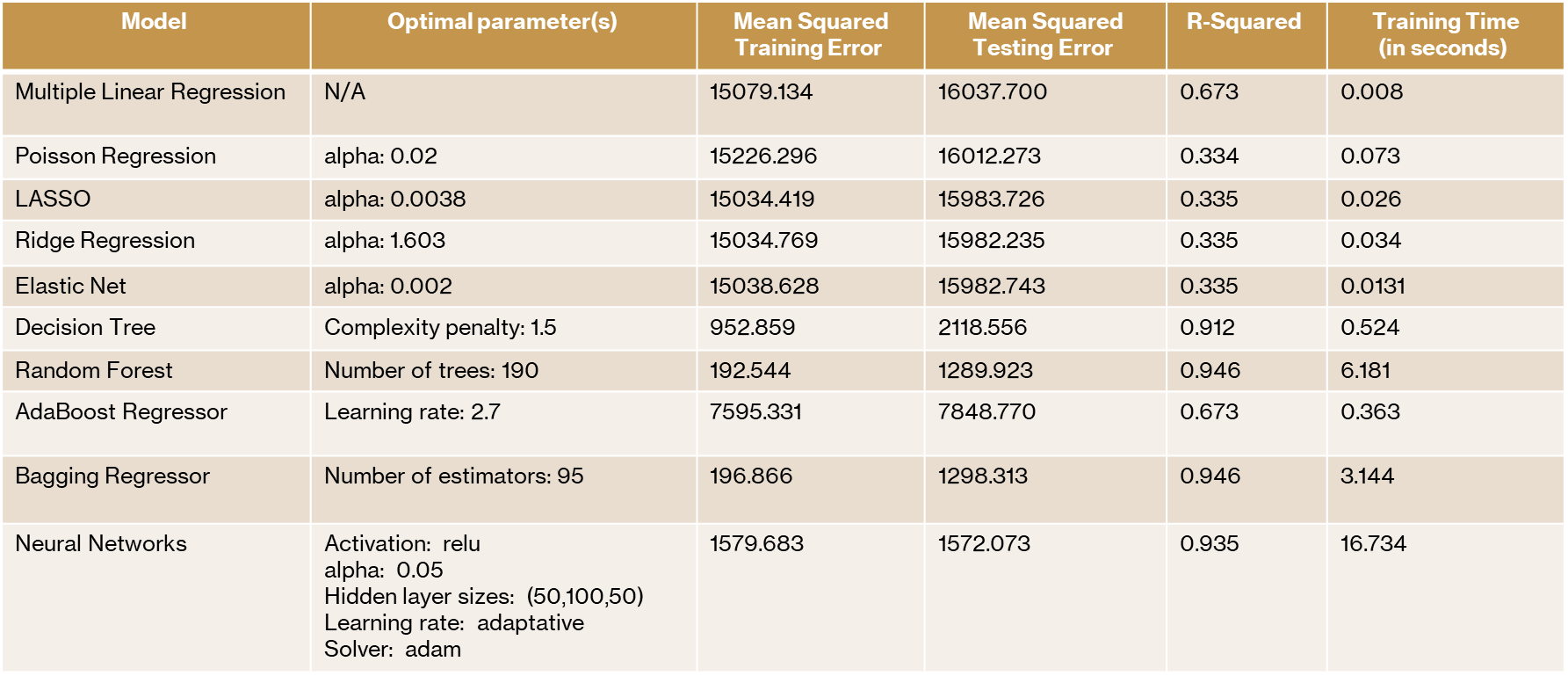


Table 4: main results – registered rides

Finally, the models were trained using the casual bike rides (“casual”). The results are shown below:

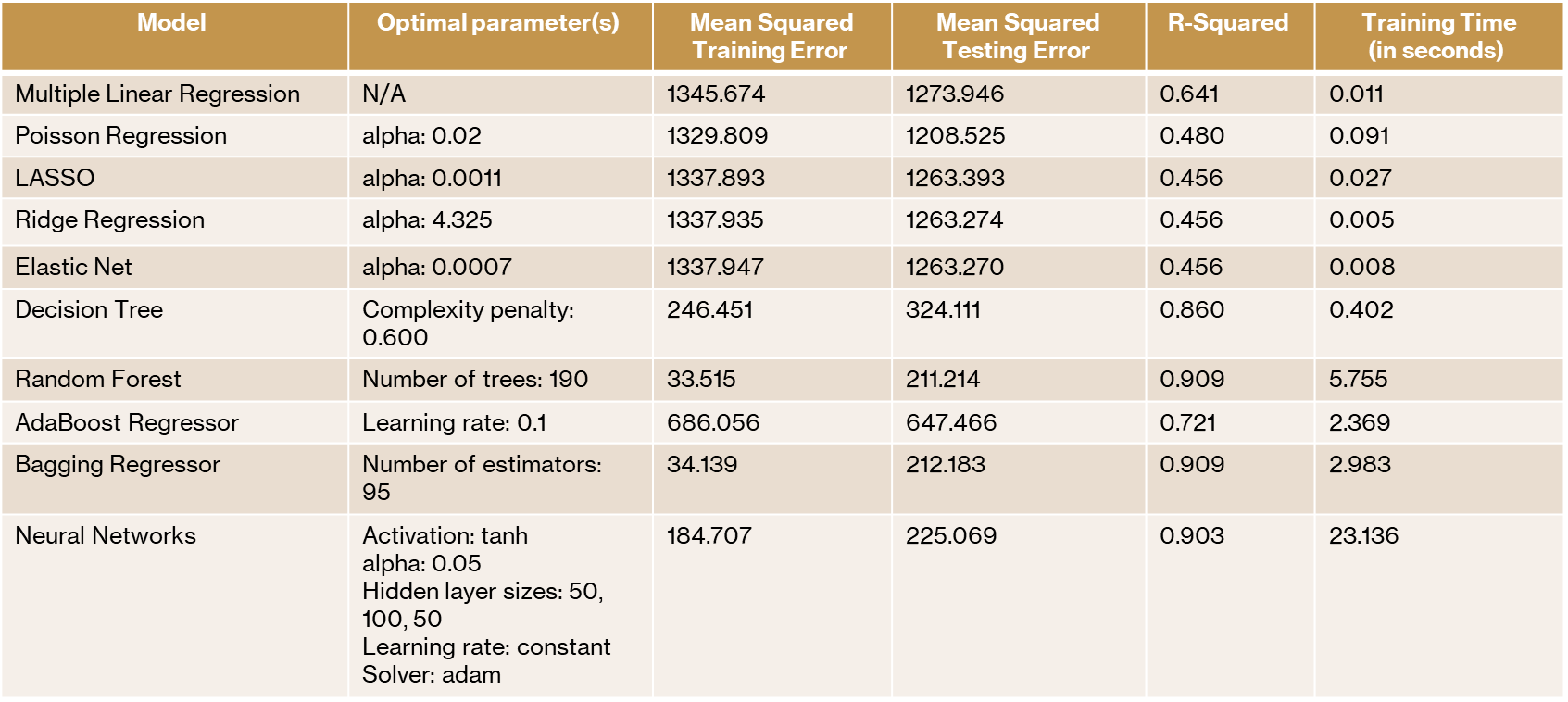


Table 5: main results – casual rides

Based on the assessment of the results from each method utilized in the prediction of bike rides, the following insights were drawn:

1. The Random Forest and Bagging models proved to be the **most effective** models to predict the number of bike rides, as shown by their R-Squared (0.95), followed by Neural Networks (0.93) and Decision Tree (0.90). They also provided the lowest testing errors.
2. Random Forest and Bagging models have in common that they utilize an **ensemble of predictors (trees)** and take as the final prediction the average of predictions from each predictor, which **reduces variance** if compared to a single tree model.
3. This variance reduction proved to be very important in developing an effective model for this problem of predicting bike rides.
4. Other models such as Multiple Linear Regression and LASSO had a lower performance comparetively. However, they were **more efficient** in terms of training time.
5. One interesting fact is that while Random Forest and Bagging presented a significant difference between training and testing errors, the Neural Networks model presented a testing error that was slightly lower than the training error, while providing good prediction results.

This may suggest that neural networks was **more effective in avoiding overfitting** of the training data, even though neural networks presented the highest training time.

1. For the models with highest predictability, which obtained a R-Squared greater than 0.9, the performance was relatively lower when predicting casual rides compared to the registered rides.

This was expected since casual rides are **more irregular** as seen by its skewed distribution, compared to the normal distribution from the registered rides.

In summary, it can be concluded that models such as random forest, bagging, decision tree, and neural networks proved to be very effective in the prediction of bike rides. Future work could be done by assessing additional models, such as the use of splines, additive models, as well as exploring a wider range of parameters and types of parameters for the neural networks model.

## Lessons we have learned

The project was foundational to solidify our understanding of machine learning algorithms and how they can be used in a real-world application.

By solving the problem of predicting the number of bike rides, we encountered challenges that are common in the data mining field, such as the presence of outliers in the data, inefficiency in cross-validating best parameters for neural networks across many types of parameters, and how to interpret the results obtained from the models. Solving this challenges provided a unique experience which can be replicated in future problems that we may face in various industries.

The course Data Mining and Statistical Learning was also important in enabling hands-on experience with different algorithms and problems that are common in the data mining field. The diversity of topics and coding examples were valuable to show how machine learning algorithms can be used in ways that generate value to society.