STA5076Z Supervised Learning Assignment 2

Adam Mosam (MSMADA002)

02 May 2023

Contents

1 Introduction

In this assignment, supervised learning techniques will be explored and used to predict the following quantities:

- Question 1: Survival rate of patients with heart failure - using data set from Chicco and Jurman (2009),
- Question 2: Bike share rentals per hour - using data set from Sathishkumar, Par and Cho (2020).

The datasets will be analysed and tested with various models which will include:

1. Logistic regression and classification trees in Question 1,
2. Random forests, gradient boosted trees and extreme boosting in Question 2.

The aim of this exercise is to gain a deeper understanding of the different modelling techniques and how they can be applied to real-world problems.

2 Question 1

In this question, medical records of 299 patients will be analysed. The dataset includes the following features:

- age: age of the patient (years)
- anaemia: decrease of red blood cells or hemoglobin (boolean)
- high blood pressure: if the patient has hypertension (boolean)
- creatinine phosphokinase (CPK): level of the CPK enzyme in the blood (mcg/L)
- diabetes: if the patient has diabetes (boolean)
- ejection fraction: percentage of blood leaving the heart at each contraction (percentage)
- platelets: platelets in the blood (kiloplatelets/mL)
- sex: woman or man (binary)
- serum creatinine: level of serum creatinine in the blood (mg/dL)
- serum sodium: level of serum sodium in the blood (mEq/L)
- smoking: if the patient smokes or not (boolean)
- time: follow-up period (days)
- death event (target): if the patient deceased during the follow-up period (boolean)

The data includes 12 independent variables and a single response variable represented by death_event.

As “anaemia”, “diabetes”, “high_blood_pressure”, “sex”, “smoking”, “DEATH_EVENT” are categorical, they have been changed to factor variables.

2.1 Part (a & b)

The data will be split into training and test sets, using a 80/20 split, respectively. Two models will be evaluated in this question, namely, a logistic regression model and a classification tree model. Both models will be run with the default settings. The following indicators will be used to measure the effectiveness of the model; classification accuracy, recall, specificity, F1 score, ROC AUC, and Matthews Correlation Coefficient (MCC). The results are shown below in Table 1.

Table 1: Indicators from logistic regression model and decission trees with default settings

Model	Accuracy Rate	Recall	specificity	F1	ROC AUC	MCC
Logistic Regression (Def.)	0.8	0.77	0.82	0.74	0.84	0.58
Classification Tree (Def.)	0.8	0.82	0.79	0.75	0.81	0.59

The MCC, as shown above for the two models, measure the difference between the actual and predicted values, and is useful when assessing unbalanced data.

Table 2: Classes of response variable

Not Deceased	Deceased
203	96

Looking at the observation count for the predictor "death_event", the data appears heavily unbalanced. Hence, the MCC in this case is an important indicator. The equation for MCC is given below

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}$$

With TP and TN being the true positive rate (recall) and true negative rate (specificity), respectively. FP and FN represent the false positive rate and false negative rate, respectively. As with most correlation coefficients, the MCC ranges between -1 and 1, where

- 1 represents perfect agreement between the actual and predicted values
- 0 indicates complete randomness between results, and
- -1 indicates perfect disagreement of results.

In the results of the two models, the MCC appears close to 1, with values above 0.5. This indicates that the models have perfromed well on the dataset.

2.2 Part (c)

In this question, the two models discussed in part (a & b) will be run 100 times on randomly partitioned training and test sets, with the mean of the indicators then extracted. These results may be viewed in Table 3

Table 3: Mean indicators from logistic regression model and decission trees with default settings, from 100 samples

Model	Accuracy Rate	Recall	specificity	F1	ROC AUC	MCC
Logistic Regression (Def.)	0.82	0.68	0.88	0.70	0.87	0.57
Classification Tree (Def.)	0.79	0.67	0.85	0.66	0.82	0.52

Boxplots of each of the indicators for both models are shown in Figure 1. The box plots reveal that the majority of the performance indicators follow a normal distribution, with a slight skew observed in the F1

scores for both models. Notably, the recall and MCC exhibit larger variances compared to the other metrics, highlighting the significance of performing multiple simulations using random samples to obtain reliable estimates.

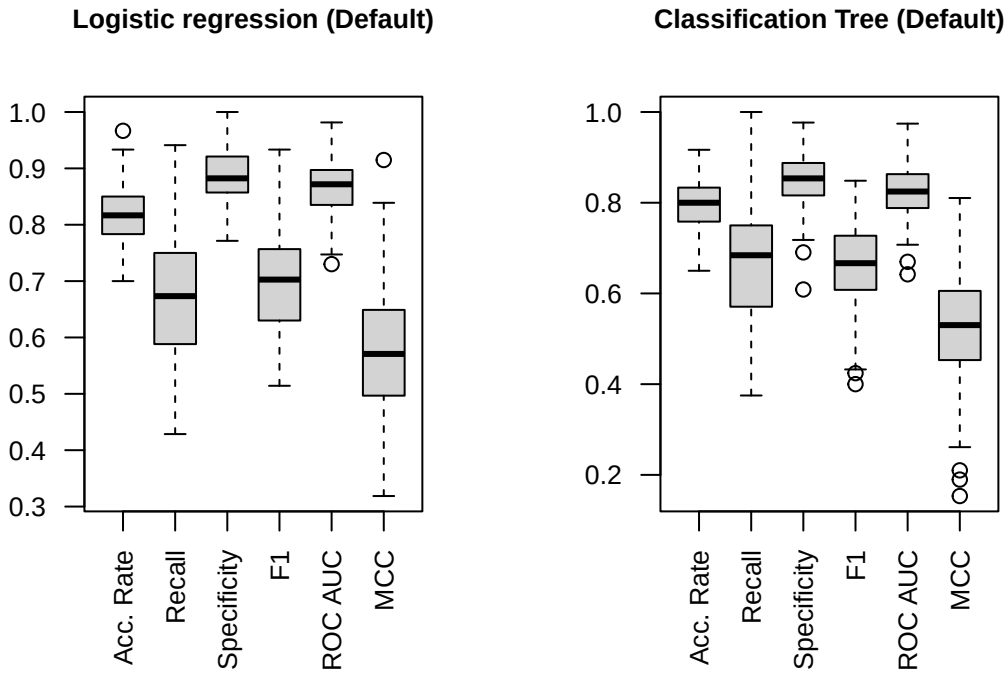

Figure 1: Boxplot of indicators results with 100 samples

2.3 Part (d)

In the context of classification trees, the ROC curve represents the relationship between the true positive rate (TPR or recall) and the false positive rate (FPR), by varying the classification threshold.

Given a set of predictor variables for a single observation, the classification tree algorithm uses the tree structure established from the training data to determine the outcome. The observation is assigned to a terminal node based on the conditions in the tree, and the probability of belonging to a class is estimated as the proportion of training samples that belong to that class within the terminal node.

To create an ROC curve, the classification threshold is varied, which adjusts the balance between the TPR and FPR. For each threshold, the tree is used to classify the observations and the resulting TPR and FPR values are used to plot a point on the ROC curve.

An example of ROC curves from Part (c) are shown in Figure 2 for the logistic regression model and the classification tree.

2.4 Part (e)

In this question, we will extend the models presented in Part (a & b) by applying L1 regularization (Lasso) to the logistic regression model, and by pruning and modifying the splitting criterion of the classification tree.

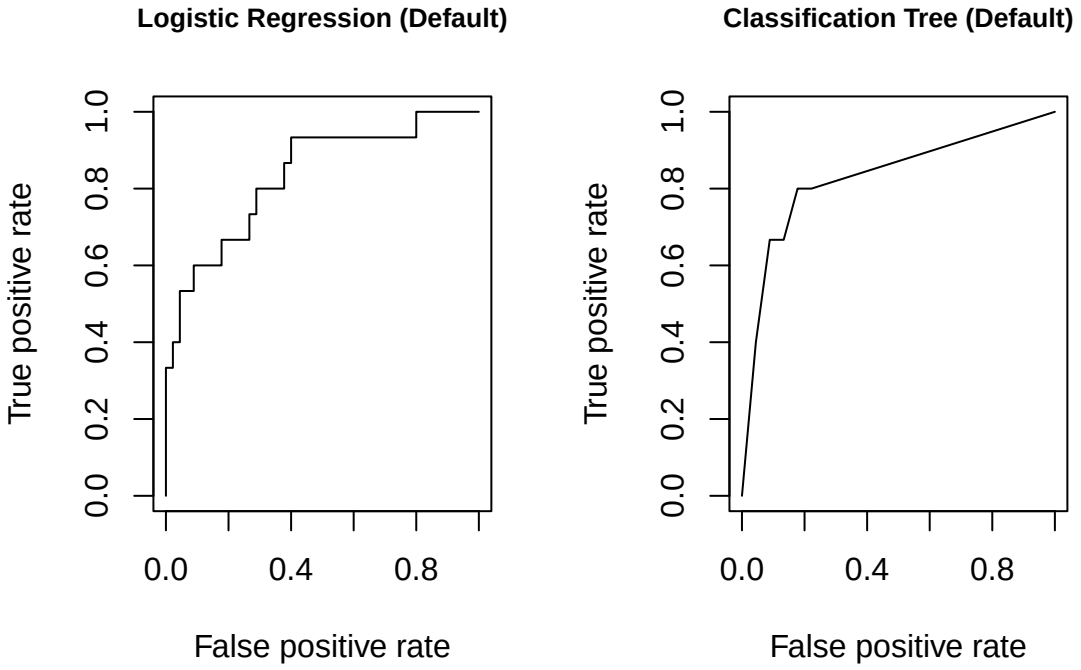

Figure 2: ROC curve using logistic regression model (default) and decission tree (default)

To determine the optimal hyperparameter λ for the logistic regression model with L1 regularization, we will use cross-validation with 10 folds to minimize the missclassification rate in the model presented. Similarly, we will use cross-validation to determine the optimal tree size for the classification tree through cost complexity pruning, by minimizing the deviance. Additionally, we will modify the splitting criterion used to grow the trees from the default option (deviance) to the gini index.

The results from all four models are shown below

Table 4: Indicators from logistic regression model (default and with L1 regularization) and desicssion tress (default and pruned tree)

Model	Accuracy Rate	Recall	specificity	F1	ROC AUC	MCC
Logistic Regression (Def.)	0.819	0.676	0.885	0.696	0.866	0.573
Classification Tree (Def.)	0.793	0.672	0.848	0.664	0.824	0.521
Logistic Regression (L1)	0.830	0.623	0.926	0.691	0.866	0.591
Classification Tree (Pruned)	0.826	0.655	0.905	0.698	0.801	0.590

As shown, the MCC from the adjusted models outperform the models run with the default settings, indicating an improved performance as a result of the parameter tuning. From all the results, the logistic regression model with L1 regularisation appeared to perfrom the best, based on the MCC score.

Chicco and Jurman (2020) investigated various machine learning models to predict the survival rate of patients with heart disease, and tested these models on the same dataset used in this study. The data in

the aforementioned study however, was subject to feature engineering, and involved two main datasets. The first dataset excluded the "Time" feature, which represents the time interval between the follow up. The second dataset included the original dataset, with the "Time" response variable.

The results presented below are based on the dataset without the response variable Time. As shown in Table 5, the results do not compare well with those shown earlier in Table 4. This is however due the models being simulated on different datasets.

Table 5: Table 4 results from Chicco and Jurman (2020), from logistic regression model (default and with L1 regularization) and desicssion tress (default and pruned tree)

Model	Accuracy Rate	Recall	specificity	F1	ROC AUC	MCC
Decision Tress	0.73	0.394	0.892	0.475	0.643	0.332
Logistic Regression	0.737	0.532	0.831	0.554	0.681	0.376

Chicco and Jurman (2020) discuss the impact of the Time response variable, and attribute greater model performance with its inclusion in the dataset. The results in Table 6 are extracted from Chicco and Jurman (2020), and include results using a logistic regression model only. To this table, the results from the logistic regression model with L1 regularization, computed in this study are added. Ina addition, the percent difference in results are also provided.

Table 6: Table 11 results from Chicco and Jurman (2020), and results from this study, using logistic regression model

Model	Accuracy Rate	Recall	specificity	F1	ROC AUC	MCC
Chicco & Jurman (2020)	0.833	0.78	0.856	0.714	0.818	0.607
This study	0.83	0.623	0.926	0.691	0.866	0.591
Percent Difference (%)	0	-25	8	-3	6	-3

As shown, the results appear to correspond well with those from Chicco and Jurman (2020), thus highlighting the significance of the Time response variable.

3 Question 2

In this question, the goal is to predict the number of bicycle share rentals per hour over using data collected over the course of one year in Seoul, South Korea, as provided in (Sathishkumar et al, 2020). The dataset includes the following features:

- Rented bike count (target): Count of bikes rented at each hour
- Date: year-month-day
- Hour: Hour of the day
- Temperature: Degrees Celsius
- Humidity: %
- Windspeed: m/s
- Visibility: 10m
- Dew point temperature: Degrees Celsius
- Solar radiation: MJ/m2

- Rainfall: mm
- Snowfall: cm
- Season: Winter, Spring, Summer, Autumn
- Holiday: Holiday/No holiday
- Functional Day: NoFunc(Non Functional Hours), Fun(Functional hours)

The data includes 13 independent variables and a single response variable represented by Rented Bike Count.

As "Seasons", "Holiday" and "Functioning Day" are categorical, they have been changed to factor variables.

3.1 Part (a)

The histogram of the response variable Rented Bike Count, is shown below

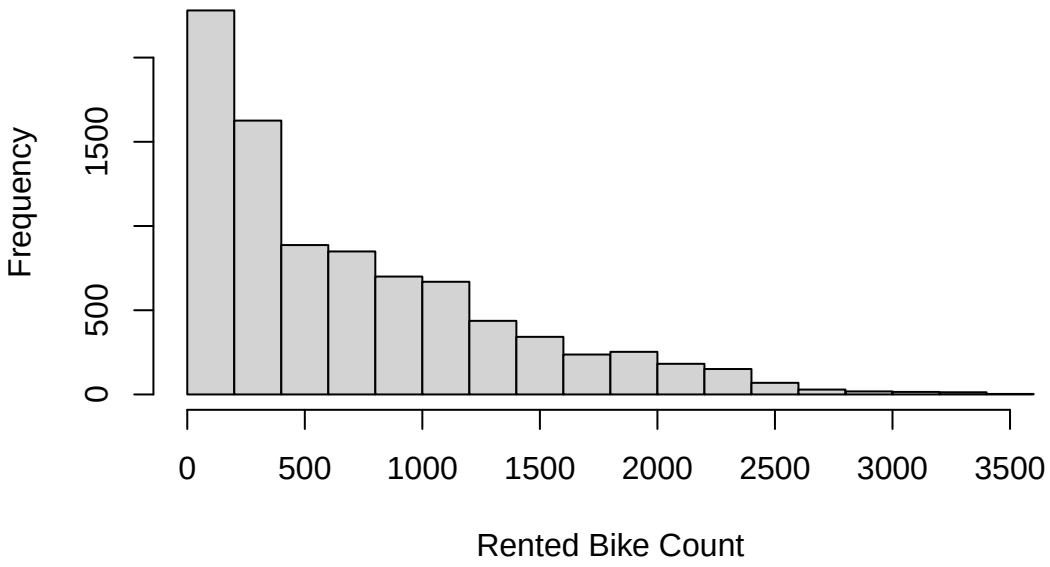

Figure 3: Histogram of response variable distribution

It is evident that the distribution of the plot is not normal and exhibits a right-skewed property. However, this may not be a major concern in the subsequent analysis as random trees and other ensemble methods that will be utilized in this section are known to be robust in handling skewed data. However, if a normal distribution is required, transformations of the response variable such as log or square root transformations may be applied.

Scatterplots of the response against various predictors are shown in the following sections. The first group of plots show the relationship of the predictor with the "Hour", "Holiday" and "Functioning day".

The scatter plot for the Hour feature displays two distinct spikes, as revealed by the best-fit curve. These spikes coincide with 8am and 6pm, indicating a high demand for bike rentals during these times. Furthermore, the scatter plots for the categorical features Holiday and Functioning day reveal that non-holidays and functioning days have the highest bike rental rates. There is a clear relationship between these findings,

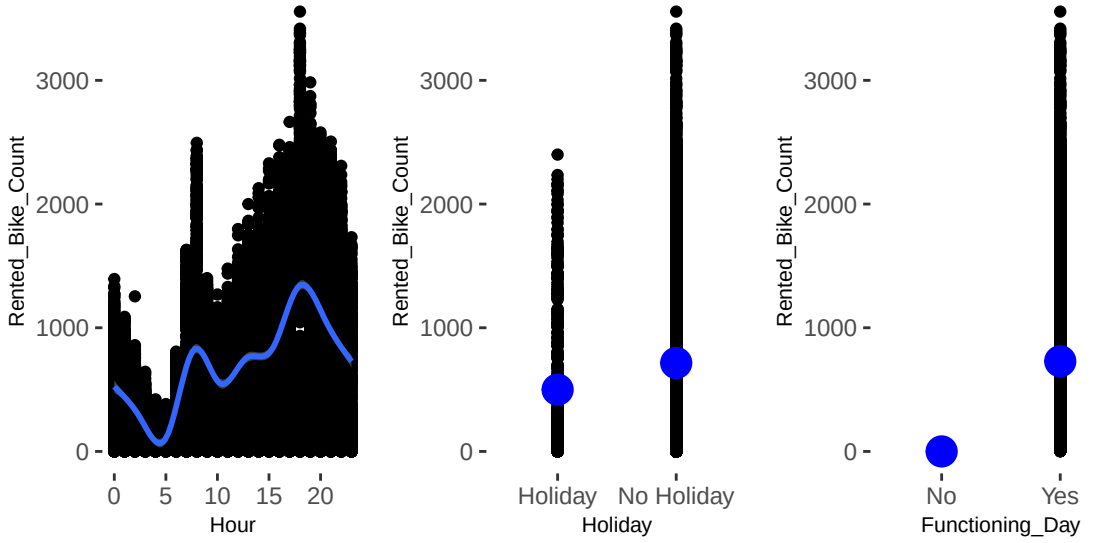

Figure 4: Scatter plot of Rented Bike Count against Hour, Holiday and Functioning day

suggesting that the majority of bike rental demand is likely from commuters traveling to and from work, which explains the concentration of rentals at 8am and 6pm.

The next group of scatter plots are intended on highlighting the affect of the weather on the bike rental demand. For this group, the features Snowfall, Rainfall, Wind speed and Visibility are shown.

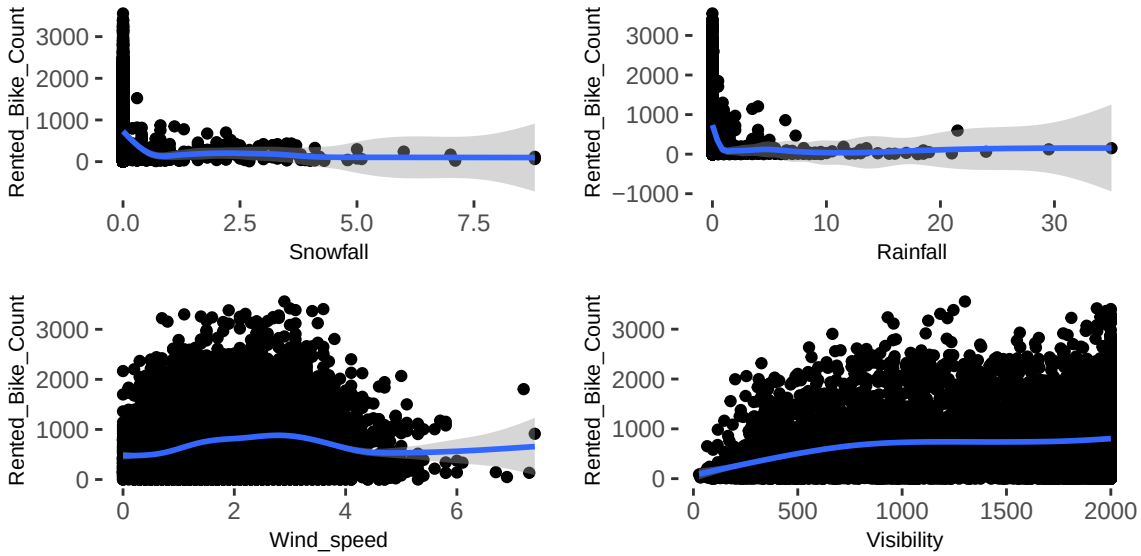

Figure 5: Scatter plot of Rented Bike Count against Snowfall, Rainfall and Visibility

The Snowfall and Rainfall plots indicate that there are numerous observations concentrated near zero with a high bike rental count. As precipitation levels increase, the number of bike rentals drops to a constant plateau. The high level of bike rentals when there is no precipitation could indicate bike rentals from leisure bike riders, with even a small amount of precipitation enough to deter them. The constant plateau of bike rentals could represent those who still need to ride regardless of the weather due to work obligations. However, the low plateau could also be attributed to laborers opting for alternative modes of transportation

such as the bus when the weather conditions are poor.

A linear relationship between visibility and the number of bike rentals can be observed, with the rental count increasing from zero at low visibility levels and then plateauing. While this suggests that bike rentals increase with higher visibility levels up to a certain point, it could also be influenced by the time of day, as the Hours plot indicates lower bike rentals during the night or early morning hours.

The wind speed is shown to have little influence on the bike rental counts, with a roughly constant bike count shown for varying wind speeds.

The impact of temperature and humidity on bike rental counts will be explored next using the features of Temperature, Humidity, Dew point temperature, and Solar Radiation. Temperature is found to have the strongest relationship with bike rentals, with higher bike counts observed on warmer days. This finding may indicate that people are more likely to rent bikes when the weather is pleasant. Additionally, the effect of humidity on bike rentals is positive, but with low bike counts recorded for very high humidity levels. This could be explained by the fact that high humidity may coincide with precipitation, making it less favorable for biking. It's worth noting that the combination of temperature and humidity can also influence bike rental counts, as extreme combinations of hot and humid conditions may discourage biking altogether. Lastly, Solar Radiation has a weak positive correlation with bike rental counts, meaning that bike rentals may be slightly higher on days with more sunshine.

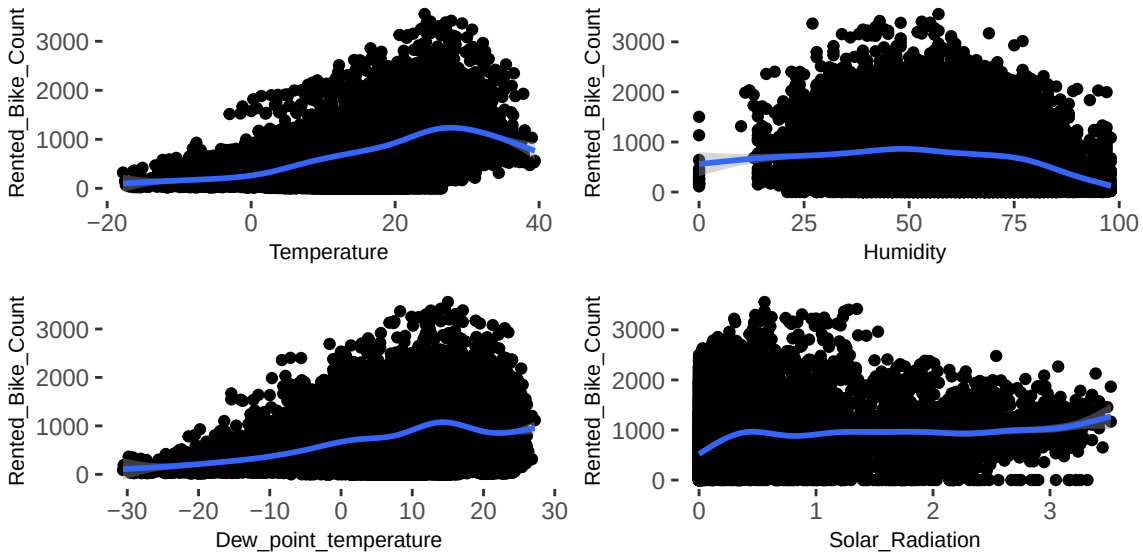

Figure 6: Scatter plot of Rented Bike Count against Temperature, Humidity, Dew Point Temperature and Solar Radiation

Finally, the plots for Seasons and Date are examined to investigate the variation in bike rentals across the four seasons and months of the year. A plot of temperature against the date is also shown. The earlier insights suggest that bike rentals are more in demand during the warmer months of spring and summer, while fewer rentals are observed during the colder months of winter. The Seasons plot confirms this observation, with the highest bike rentals in the summer, followed by spring and autumn. Winter has the lowest rental count.

Similarly, the Date plot shows a higher number of bike rentals from May to October, with a sharp drop in rental counts during December to March This could be due to the seasonal weather patterns and holidays, with winter being colder and the holiday season leading to a decrease in work commuting. Overall, the Seasons and Date plots provide additional evidence for the trend of higher bike rentals during warmer months and a seasonal variation in demand for bike rentals.

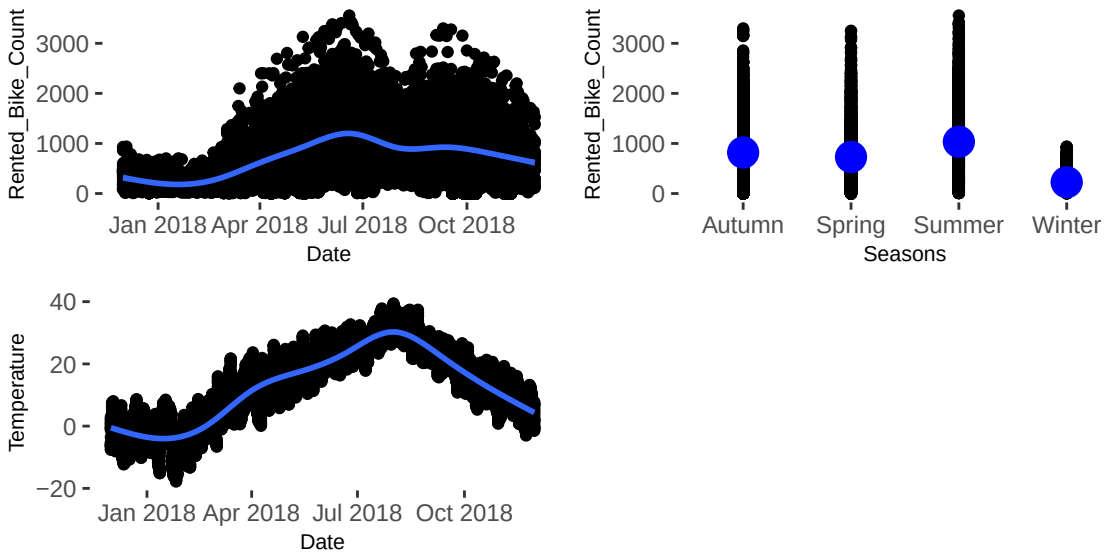

Figure 7: Scatter plot of Rented Bike Count against Date and Seasons, and Date against and Temperature

3.2 Part (b)

This question will demonstrate the process of building three models, namely:

- Random forest: using the *ranger* function.
- Gradient boosted trees: using *gbm* function.
- Extreme boosted trees: using *xgbTrees* function.

All three functions used are part of the *Caret* package. The tuning of hyperparameters and the assessment of performance on the respective dataset will be described in the sections to follow.

In addition, it should be noted that the Date variable, is converted to a numeric data type. This is done for compatibility reasons with respect to some of the functions used in this section. It should be noted that this change does not influence the results.

An 80/20 split is used for the training and test sets, respectively.

3.2.1 Random forests

The tuning parameters for the *ranger* function are as follows:

- *mtry* specifies the number of variables to consider at each split in the tree. A larger number of variables considered at a split will lead to more complex trees with less variables resulting in simpler trees. For this example, we consider the whole list of variables (2 to 13 variables).
- *min.node.size* specifies the minimum number of samples allowed at a node, with smaller values producing deeper trees. In order to determine a good range of values, we consider values of 1, 5 and 20.
- *splitrule* is the splitting criterion, with *variance* adopted as applicable for regression problems.

Additional parameters which will be tested are the number of trees to consider during training. To determine an optimal set of trees, the *randomForests* function is used. 250 trees are considered, with the variation in the out of bag RMSE produced as follows

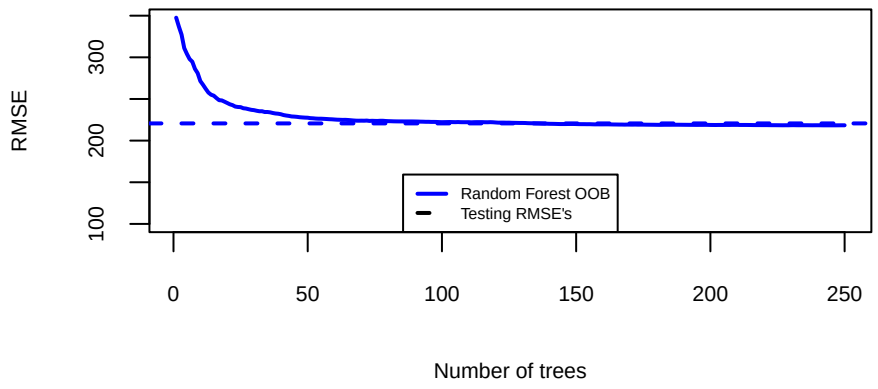

Figure 8: Training RMSE against number of trees using randomTrees

As shown, the RMSE appears to converge at the minimum at approximately 100 trees, which suggests that 250 trees is sufficient. To summarize, the hyperparameter inputs for the *ranger* grid search are as follows:

- *mtry*: 2 to 13
- *min.node.size*: 1, 5 and 20
- *splitrule*: variance

After running the training model, the optimal model was found to include the following hyperparameters:

- *mtry*: 8
- *min.node.size*: 5

3.2.2 Gradient boosted trees

The tuning parameters for the *gbm* function are as follows:

- *interaction.depth* Specifies the depth of each tree. This model will include a interaction depths ranging from 1, 5, 10 and 15.

- *shrinkage* is the shrinkage or learning rate and controls the contribution of each tree in the final model. A low shrinkage value improves the model and avoids overfitting but at the cost of computing time. In this study, shrinkage values of 0.01 and 0.1 are adopted and viewed as a good range of values.
- *n.trees* specifies the number of trees to consider. This model will consider 1000, 5000 and 10000 trees.
- *n.minobsinnode* is the minimum terminal node size, as defined in the previous section. A value of 1 chosen.

5 fold cross validation is used to determine the optimal hyperparameters.

After running the training model, the optimal model was found to include the following hyperparameters:

- *interaction.depth*: 15
- *shrinkage*: 0.01
- *n.trees*: 10^4

3.2.3 Extreme boosted trees

The tuning parameters for the *xgbTrees* function are as follows:

- *nrounds* is the number of boosting rounds or iterations to perform and controls the number of trees used in the model.
- *eta* is the learning or shrinkage rate.
- *max_depth* specifies the maximum depth of each tree.
- *gamma* is the the minimum loss reduction required to split a leaf node. A larger value of gamma will result in fewer and more conservative splits. This model considers a value of 0.001 for *gamma*.
- *colsample_bytree* is the fraction of columns to be randomly sampled for each tree. A value of 1 is chosen for this parameter.

- *min_child_weight* is the he minimum sum of instance weight needed in a child. A value of 1 is chosen for this parameter.
- *subsample* represents the fraction of instances to be randomly sampled for each tree. A value of 1 is chosen for this parameter.

For *nrounds*, *eta* and *max_depths*, the values used for gradient boosted trees are applied. In addition, 5 fold cross-validation is used to determine the optimal model.

After running the training model, the optimal model was found to include the following hyperparameters:

- *nrounds*: 10^4
- *eta*: 0.1
- *max_depth*: 5

3.2.4 RMSE results

The RMSE results from all three models are shown below

Table 7: Test set RMSE result, and the standard deviation of the test set target values

Sigma	Random.Forest	Gradiant.Boost.Tree	Extreme.Boosted.Tree
632.9	216.04	200.9	200.97

To evaluate the performance of the models used in this section, it's useful to compare the Root Mean Square Error (RMSE) metric to the standard deviation of the target values.

Comparing the RMSE values produced by the various models to the standard deviation reveals that the models are performing well. Specifically, the RMSE values are approximately one third of the standard deviation, which suggests that the models are accurately capturing the variability of the target values and making reliable predictions. The gradient boost tree appears to have performed the best, having reached the lowest RMSE value.

3.3 Part (c)

The variable importance plot is shown below.

From the variable importance plot, it is evident that similar results are produced across all three models. To sum up, the results from the variable importance plot and the previous analysis provide compelling evidence

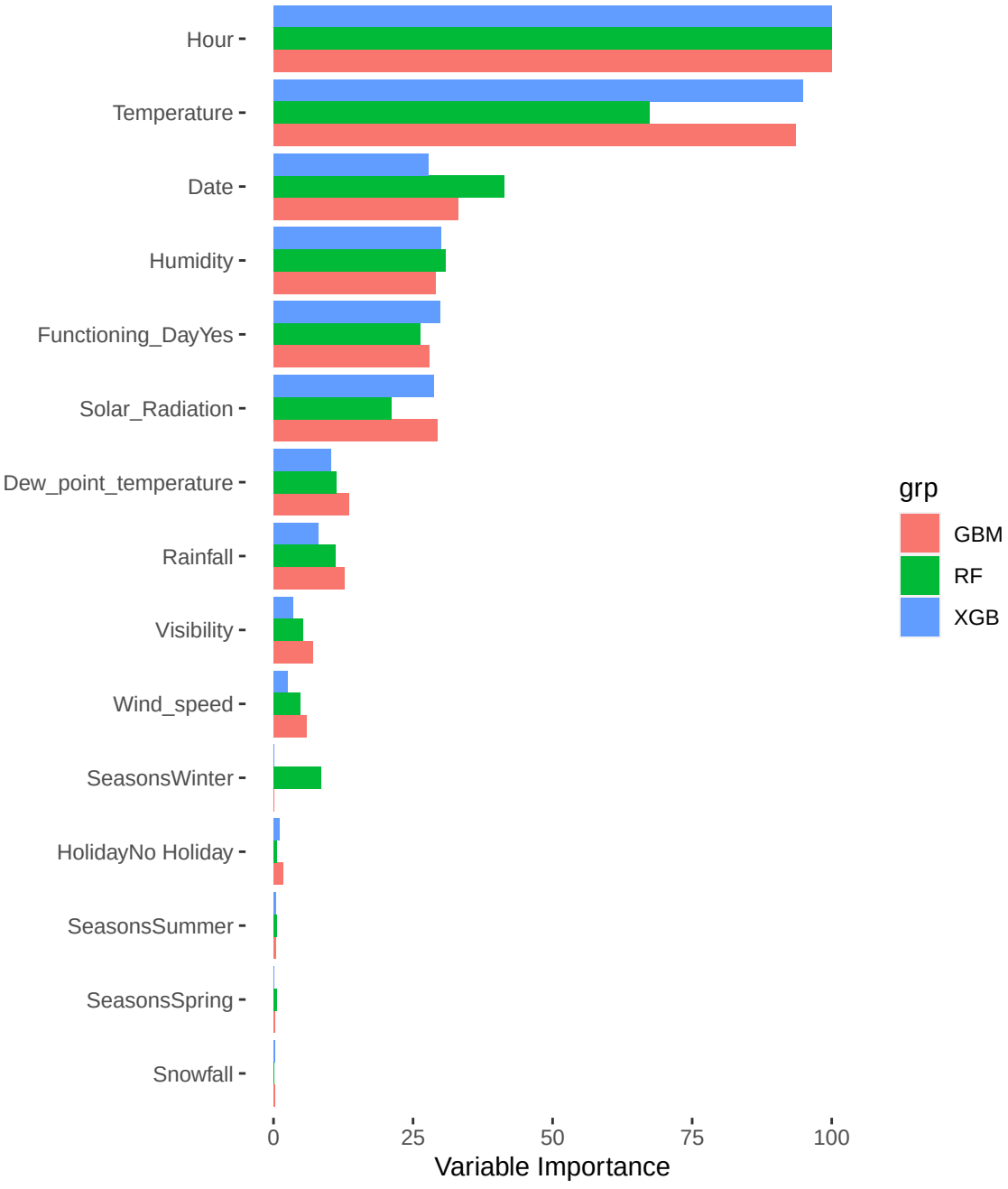

Figure 9: Variable importance barchart

that Hour and Temperature are the key features in forecasting bike rental demand. The Hour feature displays distinct spikes during peak commuting times, whereas Temperature has a strong positive association with rental counts, indicating that favorable weather conditions lead to higher rentals. Moreover, the importance of other variables, such as Humidity, Solar Radiation, and Functioning day, aligns with the earlier findings, supporting their impact on rental counts. In essence, the better the weather and working conditions, the higher the bike rentals, highlighting the importance of these features in predicting bike rental demand.

References

Chicco, D. and G. Jurman (2020). "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone". In: BMC Medical Informatics and Decision Making 20 (16). issn: 1472-6947. url: https://doi.org/10.1186/s12911-020-1023-5.

Sathishkumar, V. E., Jangwoo Park, and Yongyun Cho (2020). "Using data mining techniques for bike sharing demand prediction in metropolitan city". In: Computer Communications 153, pp. 353–366. issn: 0140-3664. doi: https://doi.org/10.1016/j.comcom.2020.02.007. url: https://www.sciencedirect.com/science/article/pii/S0140366419318997.