Final Project Report – Kaggle Titanic - Shih-Yuan Wang

Part A. Problem Statement

On April 15, 1912, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there were not enough lifeboats for everyone onboard, resulting in the death of 1,502 out of 2,224 passengers and crew. Even though there was some element of luck involved in surviving, not all passengers had an equal chance to escape onto lifeboats and survive. It seems some groups of people were more likely to survive through the shipwreck than others. This project is designed to explore what factors would influence a passenger's likelihood to survive using machine learning algorithms based on passenger data, including name, age, gender, and socio-economic class. The goal is to predict whether a passenger survived the sinking of the Titanic.

- Data source and problem setting: Kaggle Competition - Titanic - https://www.kaggle.com/c/titanic/overview

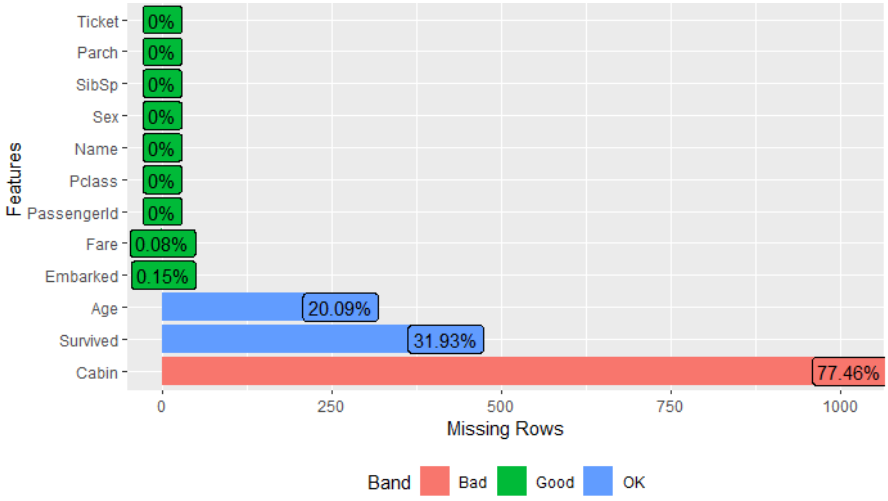
Part B. Machine Learning Approach

1. Access the Data Set

- There are 891 observations and 12 variables in the training data set, and the response variable is "Survived" (binary value). We need to predict other 418 passengers' survival for the test data set.

2. Data Overview and Clean the Data

- The data set consists of passenger identifier (PassengerID), numerical features (Age, Fare, SibSp, Parch), categorical features (Survived, Pclass, Sex, Embarked), and text (Name, Ticket, Cabin) features.
- There are missing values in column Fare, Embarked, Age, and Cabin for the combined training and test data set.



3. Explore Data and Impute Missing Values

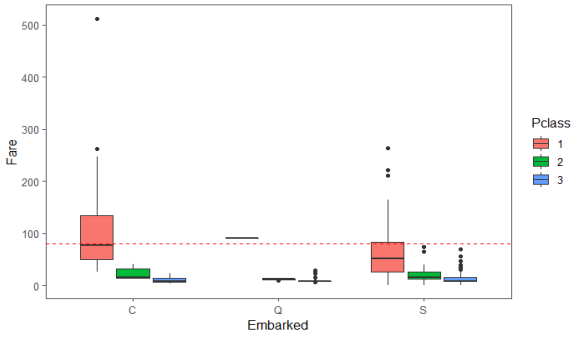
(3.1) Feature "Embarked" and "Fare"

I assume that Pclass, Fare, and Embarked might be related to each other since the upper class (1st) passengers are more likely to purchase more expensive tickets, and passengers in the same class or with the similar fare of tickets probably would embark at the same port. The following is the imputation method I used for missing values.

- **Embarked (2 missing values):** These two passengers are all female and Pclass 1, paid a fare of $80, and no family members were aboard the Titanic.

	PassengerId <int>	Survived <fctr>	Pclass <fctr>	Name <chr>	Sex <fctr>	A... <dbl>	SibSp <int>	Parch <int>	Ticket <chr>	Fare <dbl>	Cabin <chr>	Embarked <fctr>
62	62	1	1	Icard, Miss. Amelie	female	38	0	0	113572	80	B28	*NA*
830	830	1	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62	0	0	113572	80	B28	*NA*

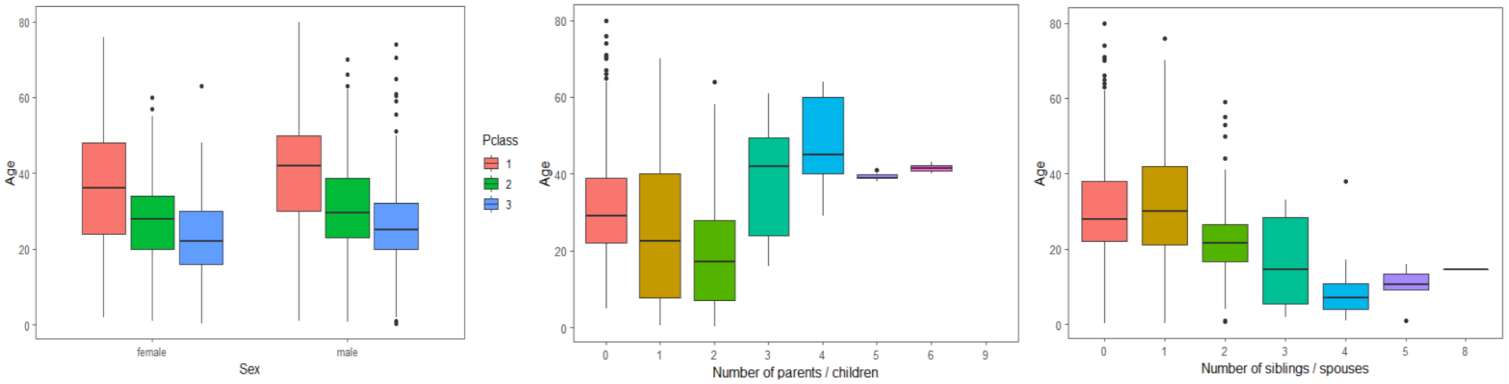
We can see how the Fare is distributed among Pclass and Embarked port in the right plot. Passengers with a fare of $80 and Pclass 1 were more likely to embark at port "C", so I replaced the Embarked missing values with "C".



- **Fare (1 missing value):** As this passenger is in Pclass 3 and embarked at port S, I replaced this missing Fare with the median fare for the class 3 and port S.

(3.2) Feature "Cabin": Most (77.5%) of the Cabin data is missing, and Pclass, Ticket, and Embarked features might contain relevant information for cabin, so I decided to exclude this feature.

(3.3) Feature "Age": There is around 20% of Age data is missing. I examined how age is distributed among the Sex, Pclass, Parch, and SibSp features to see which feature is more correlated with age.



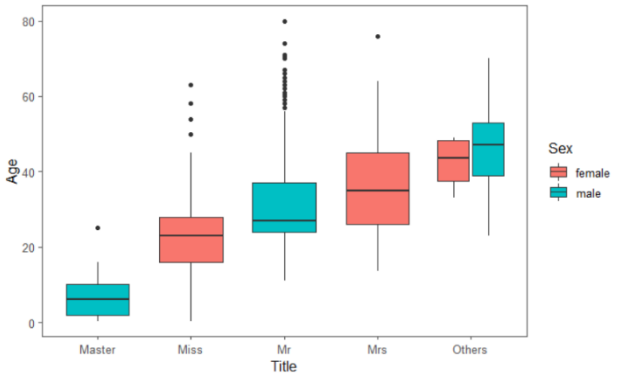
The plots show that the age distribution seems to be quite similar for male and female, but passengers in upper class are older than passengers in lower class (1>2>3). Moreover, it seems that the more parents/children and the less siblings/spouses passengers have, the older they are. As Pclass, Parch, and SibSp features are more correlated with age, I replaced the missing Age with the median age for the same Pclass, Parch and SibSp.

4. Feature Engineering

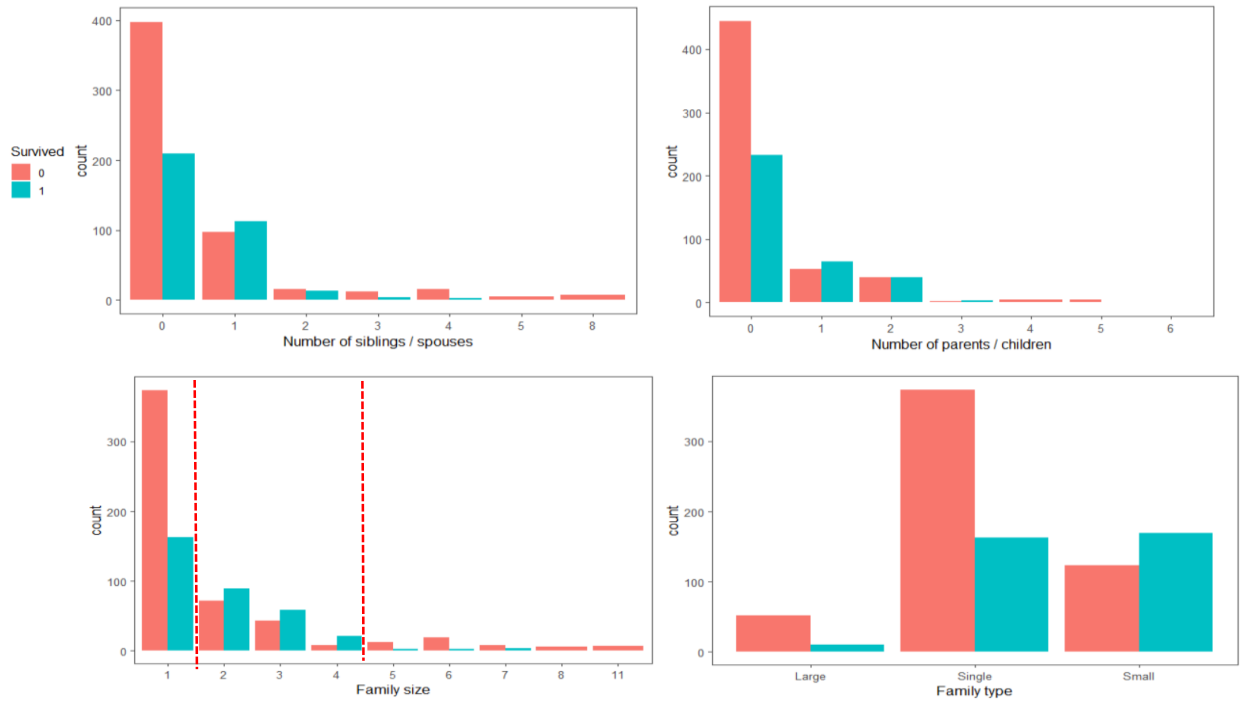
(4.1) "Ticket" number: The similar ticket number may stand for similar accommodation types or facilities they could use, so I decided to categorize the ticket by extracting the first character of the ticket number. If the ticket number only contains numeric value, it represents as "N". Count table:

A	C	F	L	N	P	S	W
42	77	13	5	957	98	98	19

(4.2) Passenger "Name": As each passenger's name includes title, I extracted their title to create another feature and regrouped some less frequent titles into more common titles or "Others" group. The plot below is the age distribution among the Sex and new feature "Title".

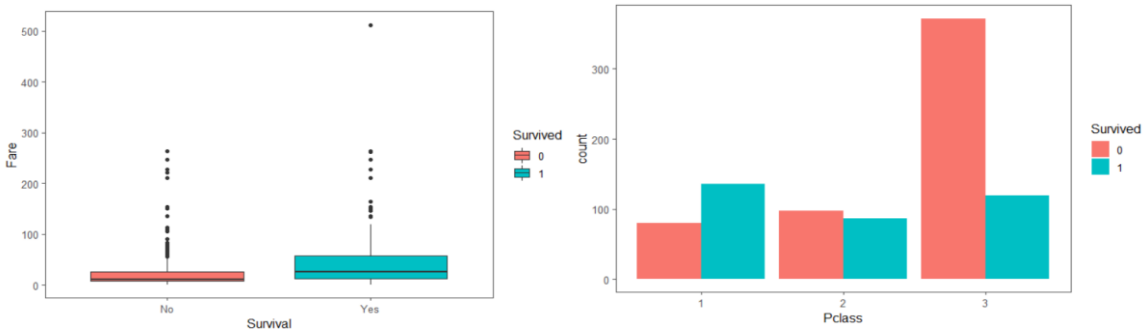


(4.3) Total number of family members: To figure out whether larger size families would have less chance to survive, I observed the relationship between "Survived" and "SibSp and Parch" features first, and created a new feature "familyType". The plots below show that single or large families (>4) have less chance to survive, so I decided to categorize them into 3 groups - Single, Small, and Large family.

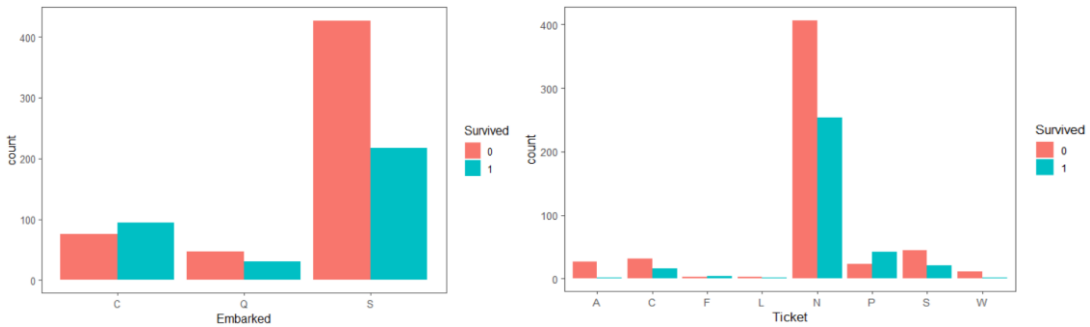


5. More Exploratory Data Analysis: The relationship between features and survival rate.

- **Fare / Pclass vs. Survival:** The higher ticket fare or the upper class is, the more likely passengers survived.

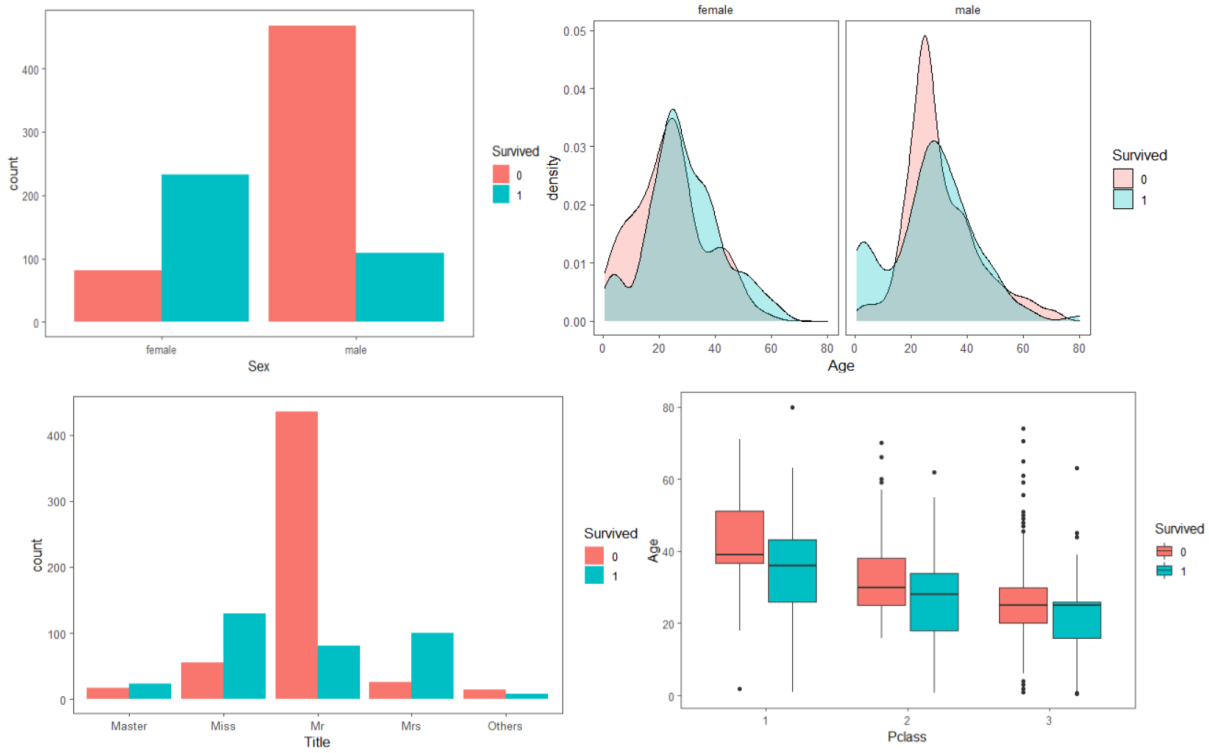


- **Embarked vs. Survival:** Higher percentage of passengers died if they boarded at port Southampton(S), while more passengers boarding at port Cherbourg(C) survived.
- **Ticket vs. Survival:** Higher percentage of passengers whose ticket number starts with "P" survived.



- **Sex / Age vs. Survival:** Females were much more likely to survive than males, and male children younger than 10 years old were more likely to survive than males between 20-30 years old. (see next page top 2 plots)
- **Title vs. Survival:** Higher percentage of passengers with the title Miss, Mrs, and Master survived. (see next page bottom left plot)
- **Age and Pclass vs. Survival:** In the same class, passengers who survived are a bit younger than passengers who did not survive. (see next page bottom right plot)

Overall, the rule of "women and children first" can be observed. Also, higher percentage of passengers survived when they are small family and in upper class, have higher fare tickets, and boarded at port Cherbourg(C).



6. Feature Scaling and Data Splitting

Numerical features were standardized to prepare for modeling, and data was split into 75% training data set and 25% test data set for the purpose of model evaluation.

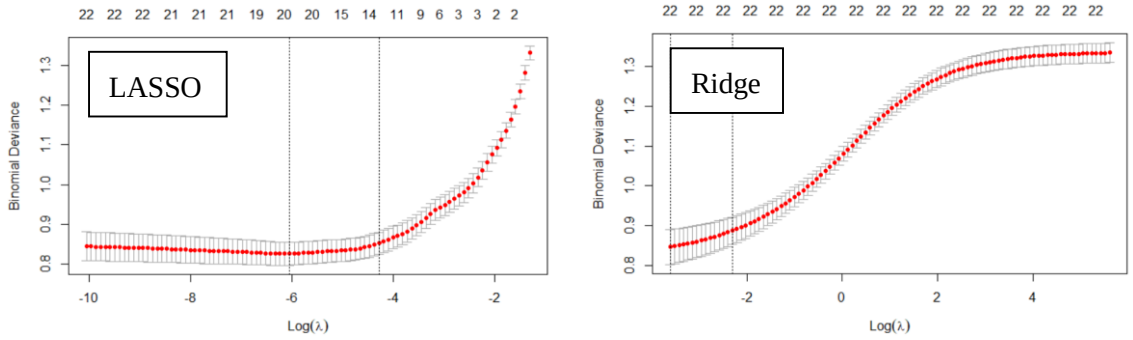
7. Fitting Models: Use 50% threshold to determine the survival.

(7.1) Logistic Regression and Linear Discriminant Analysis (LDA)

Fit the Logistic Regression and LDA model using glm() and lda() function, respectively.

(7.2) LASSO and Ridge Regression

For both LASSO and Ridge Regression models, I used 10-fold cross-validation to choose the parameter λ, and used glmnet() function and the value of λ that generates smallest cross-validated error to fit models.

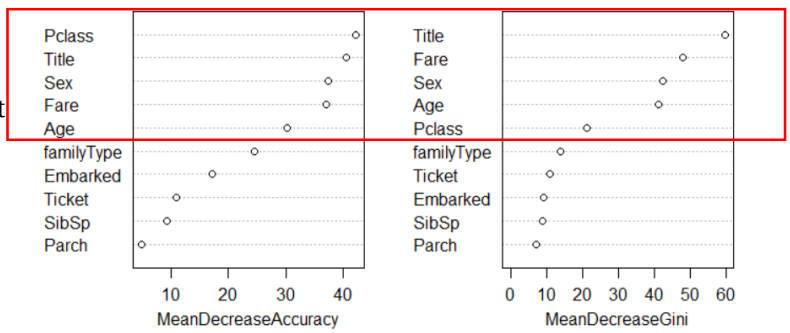


(7.3) Support Vector Machine (SVM)

I tuned the parameters using grid search, and gamma = 0.01 and cost = 5 perform best out of the chosen parameters. Then, I fit the SVM model using SVM() function based on these tuning parameters.

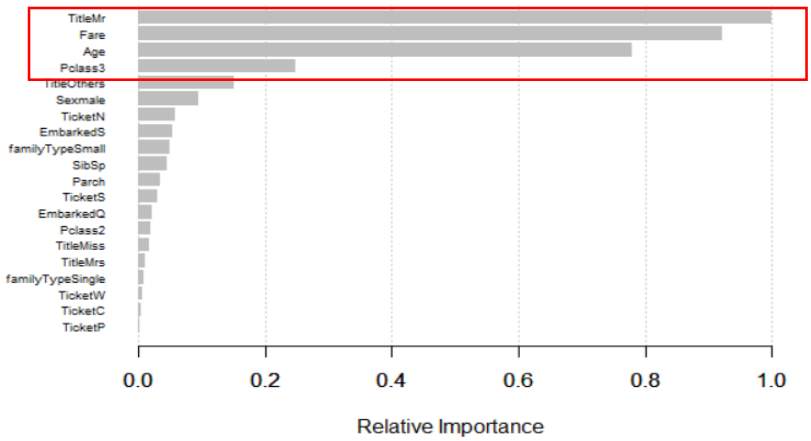
(7.4) Random Forest

I used randomForest() function to fit the model based on 3 randomly sampled variables at each split and 1,000 trees. From the right variable importance ranking plot, we can see that Pclass, Title, Sex, Fare, and Age are all quite important in this model.



(7.5) Gradient Boosting

Again, cross-validation was used to select tuning parameters. Then, I used xgboost() function with tuning parameters to fit Gradient Boosting model. Not surprisingly, the feature Title, Fare, Age, and Pclass are relatively important in the model.

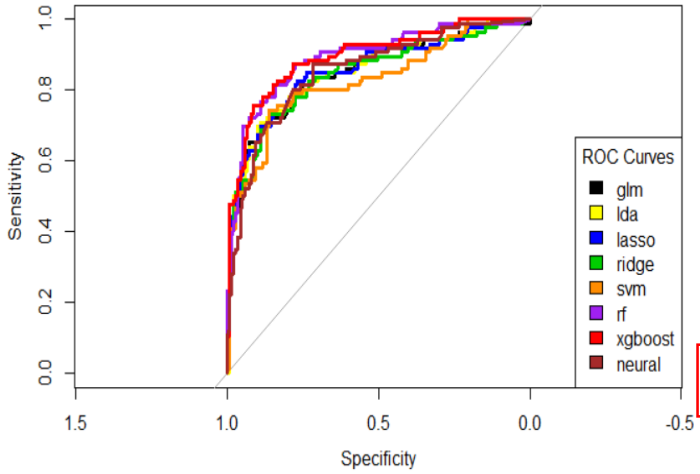


(7.6) Neural Nets

I did some mild tuning for both single and two-layered neural network hyperparameters and selected one of the single-layer neural network models with better performance.

8. Model Evaluation and Conclusion

The ROC curves and the Area under the ROC curve (AUC) for the above classifiers are as follows:



Model	AUC
Logistic Regression	0.8520625
Linear Discriminant Analysis (LDA)	0.8542692
LASSO Regression	0.8544390
Ridge Regression	0.8464607
Support Vector Machine (SVM)	0.8269394
Random Forest	0.8915295
Gradient Boosting	0.8944152
Neural Nets (single-layer)	0.8535053

Among these models, the Gradient Boosting classifier performs best in this setting, and the Random Forest classifier also performs quite well. In terms of the AUC measure, the performance of the classifiers: Gradient Boosting > Random Forest > LASSO Regression > Linear Discriminant Analysis (LDA) > Neural Nets (single-layer) > Logistic Regression > Ridge Regression > Support Vector Machine (SVM), but they're quite close. Overall, the Gradient Boosting and Random Forest models seem to be a bit more competitive than other classifiers in this setting. However, the Ridge Regression model got the highest score in the Kaggle submission, but all models got quite similar scores ranging from 0.75 to 0.8. Thus, for further improvement, I would:

(1) Try different approaches for feature engineering and be more careful about multicollinearity.

(2) Not just use all features to fit models, try several different combinations of features instead.

(3) Find out which observations are outliers and determine whether to exclude some of them.

(4) Run more grid search and repeated k-fold cross-validation to increase the performance.

Note:

Code References: Please refer to the reference list in the end of the R markdown file.