Bike Demand Forecast Report

Report Prepared by:
Devika Bhatt
Yu-Jui Chen
Yixuan (Katie) Jin
Divya Krishnan
Chih-Hsiang Lai
Yu-Cheng (Wally) Liu
Fiona D' Souza

Executive Summary:

The main problem we are helping the business resolve with better forecasting of the demand includes resource allocation, finding alternative markets etc., which helps them better manage their inventory and realize economic gains. We started out by running a regression analysis and found the following important variables - Temperature, season and humidity. We also ran machine learning algorithms, namely Linear regression, Random forest, SVM, Decision Tree and the others and found that SVM gave us the best results. (Appendix 1 shows the result for the machine learning algorithms we ran) Based on our analysis, we found that demand forecasting can help the business better allocate their resources/equipment. With the additional resources, we can tap on other markets, where the conditions are more favorable.

1. Objective of the Forecasting Problem

Bike Sharing Systems are becoming popular and there are over 500+ bike sharing programs in the world. Those bikes are rented and returned to the station daily after the usage. Bike companies need to forecast bike demand data based on the historical data sets to leverage business value in the bike sharing market. This project report provides an analysis and evaluation of both the current and prospective future bike demand that can help the organization to design and allocate their resources in the supply chain systems. Currently, the problem is that we need to consider various data in the BikeDemandDaily.csv and provide a better bike demand forecasting plan to fulfill the real bike demand and enhance our inventory management. The file consist a series of data such as temperature, humidity, wind speed, casual and registered customers which promote us to find the significant correlations between each factors to predict a more accurate demand. To be more specific, we need to develop steps of regression model to forecast the estimated demand and train the machine learning model to evaluate and address the potential business value.

2. Steps of Forecasting

1. **Problem recognition:** The first step in forecasting involves identifying and recognizing the problem. It involves collecting data, maintaining databases, and using the forecasts for future planning. Based on our objective, our problem here is to identify the demand for bikes based on different factors influencing it.
2. **Data Cleaning:** In our second step, we try to find the relevant variables affecting the demand in bikes. By filtering and cleaning the data, we were able to remove the inaccurate, corrupt or irrelevant data from the present database.
3. **Modelling:** We first used Linear Regression model and then tried the same model with added parameters. The RMSPE value changed along with different parameters. We then used Stepwise model to choose the best set of variables and identified them as Trend, Seasonality and Other Factors.
4. **Machine Learning:** This step involves training the machine/system to identify and predict the bike demand. We performed the training using various algorithms including

the Decision Tree, K Nearest Neighbor, Random Forest and SVM algorithms and found the best mechanism of the two on the basis of RMSPE value.
In this study, the most efficient algorithm was found out to be the SVM classifier.

3. Sketch of results

With the historical data and predicted data, we can predict the future demands based on some weather features. Since we try multiple models, we can identify a best model for accurate prediction. We sketch all the result in Appendix 1. Besides, to understand if there are other efficient algorithms, we also predict the data with 7 different machine learning algorithms and do the cross-validation on Python. Although the result is slightly different from the result by using R, we can also identify some additional algorithms for improving forecasting performance. The difference may come from the model we choose. (For example, regression model rather than classifier). For making managerial decisions, we can understand the historical distribution in each season, temperature or year , for allocating the resource or arrange the bicycle maintenance. (Appendix. 4)

4. How can management use the demand forecast for improving profitability?

- *Human Resource Allocation*: Demand forecasting can help the managers with better resource allocation and reduce costs associated with additional labor.
- *Repairment coupled with seasonal distribution:* The cycle demand varies based on seasons and weather conditions. Hence planning resources based on seasons will help in transferring and allocating resources based on the region's demand.
- *Expanding for the new location before the market is full* [with assumption]: This can help the manager with the decision process by achieving in first mover advantages in scenarios where such opportunities are available.
- *Aggregate pricing:* This can help in distribution by coupling goods based on demand.

5. What are the key learnings from the project?

Some of the key learnings from the project include:

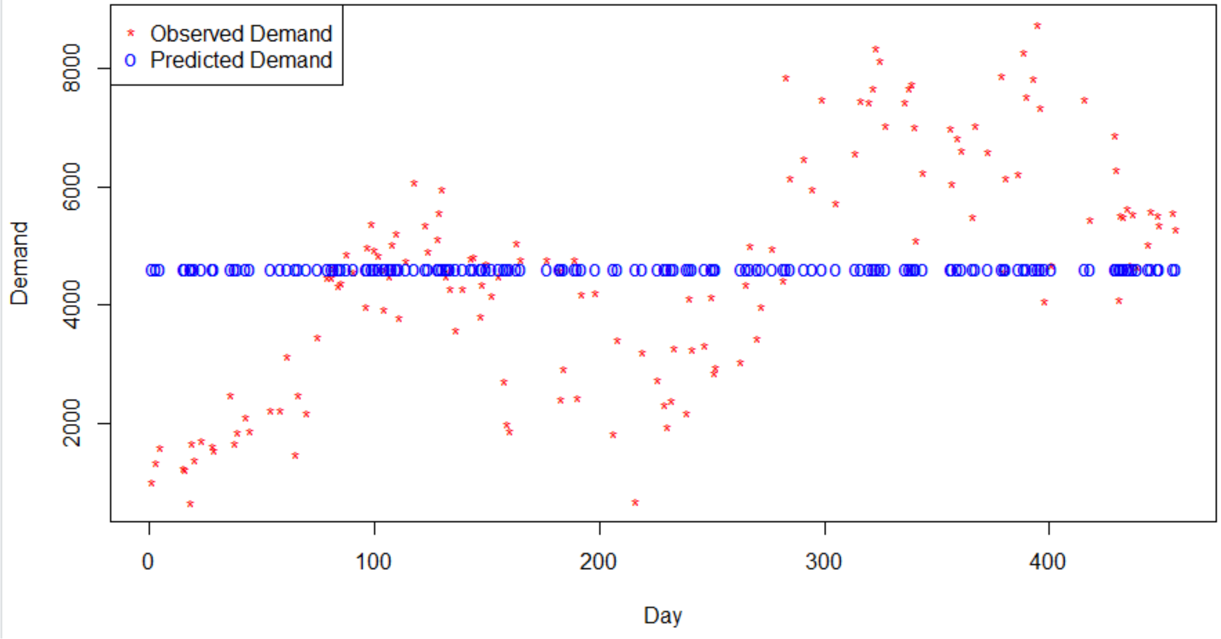
- How demand forecasting can help the organization understand the customers demand as well as their ordering behavior, which in turn helps the organization better manage their inventory.
- It also helps the organization work with less inventory as they are more certain and confident about the demand.
- Knowing the demand enables organizations to ensure the right amount of inventory is available at the right place and the right time.

Appendix

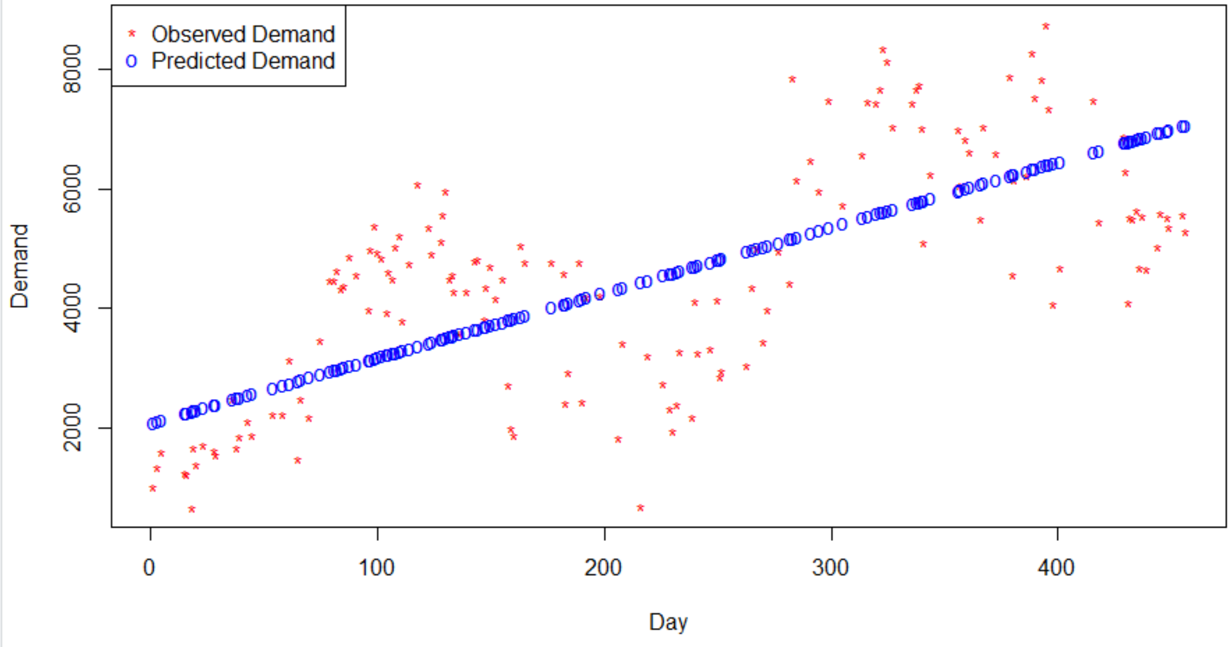
Appendix 1. Table of RMSPE and graph for each prediction model and machine learning algorithm

	Model	RMSPE
	Regression	
1	Linear Model: Intercept Only	1841.84
2	Linear Model: +Trend	1182.102
3	Linear Model: +Seasonality	846.4533
4	Linear Model: +Other Factors	694.8407
5	GLM: Gaussian	694.8407
6	GLM: Poisson	814.7978
7	GLM: Negative Binomial	919.7653
8	Stepwise Regression	638.1186
9	LASSO Regression	650.073
	Machine Learning (Data Mining)	
10	Random Forest	578.7885
11	SVM	572.9129

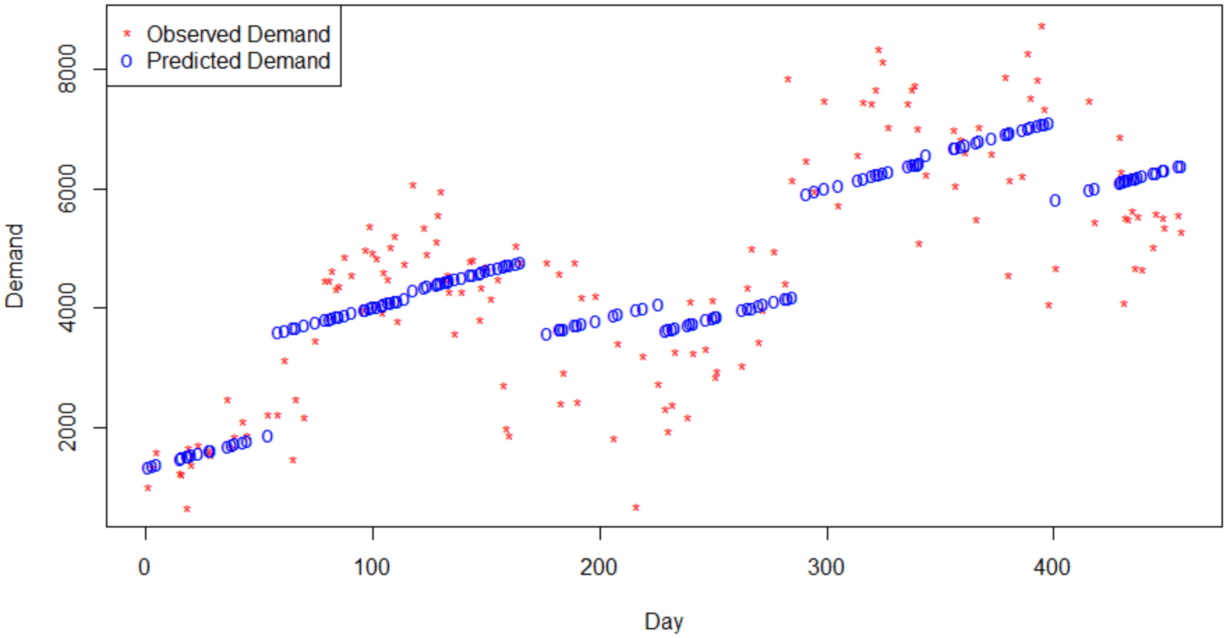
1. Linear Model: Intercept Only (1841.84) ~ lm(Total~1, data=dtrain)



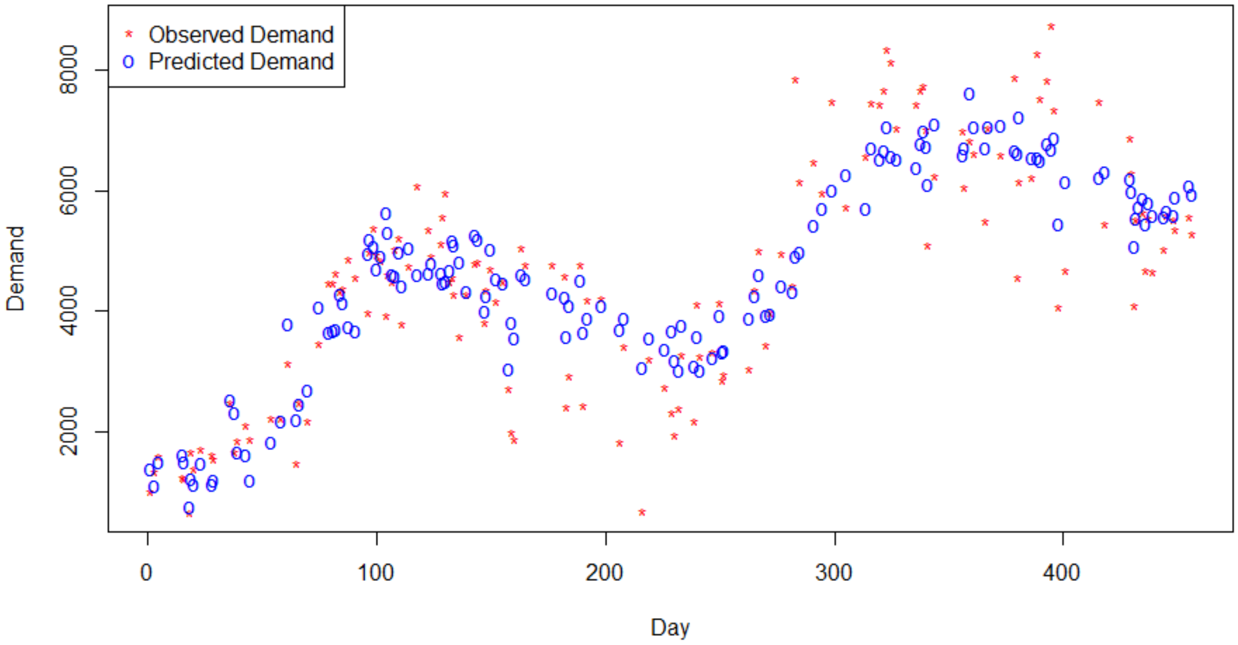
2. Linear Model: +Trend (1182.102) ~ lm(Total~Index, data=dtrain)



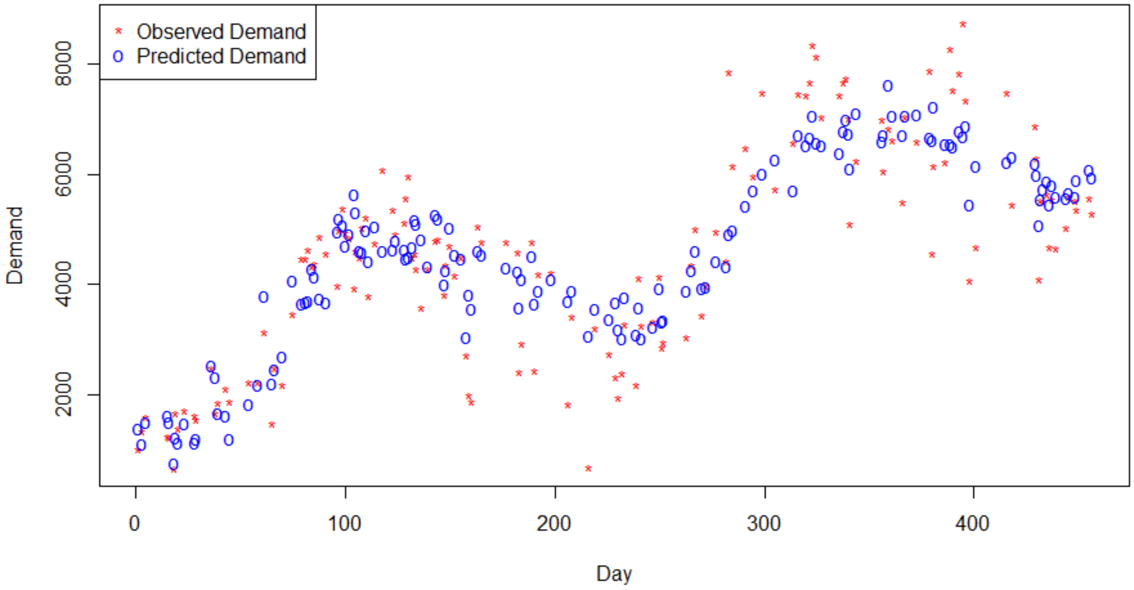
3. Linear Model: +Seasonality (846.4533) ~ lm(Total~Index+as.factor(season), data=dtrain)



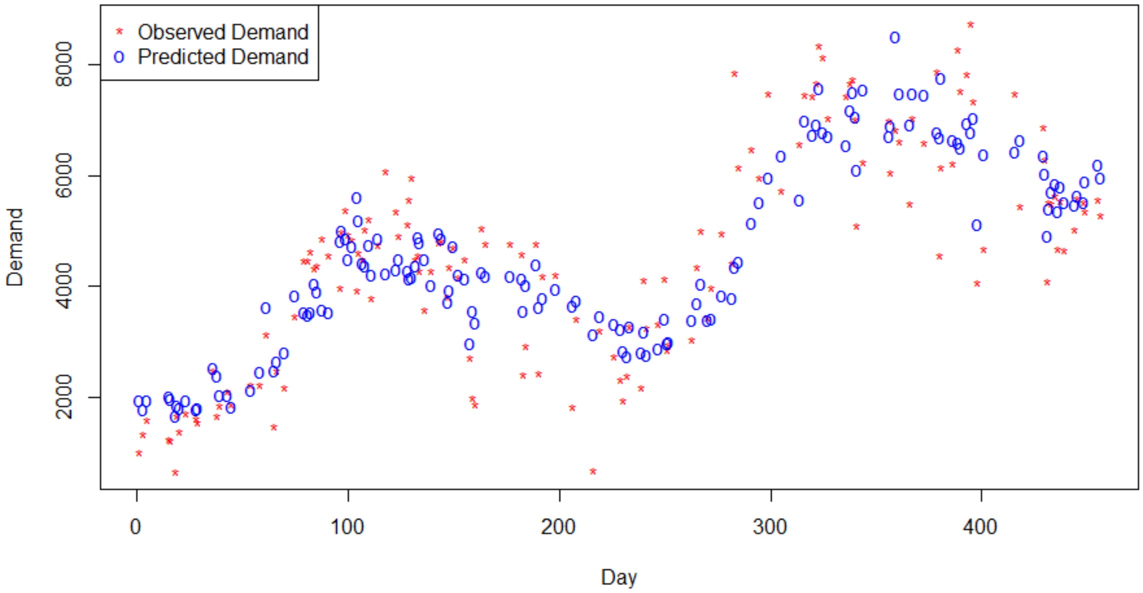
4. Linear Model: +Other Factors (694.8407) ~ lm(Total~Index+as.factor(season) +as.factor(holiday)+meanatemp+meanwindspeed+meanhumidity, data=dtrain)



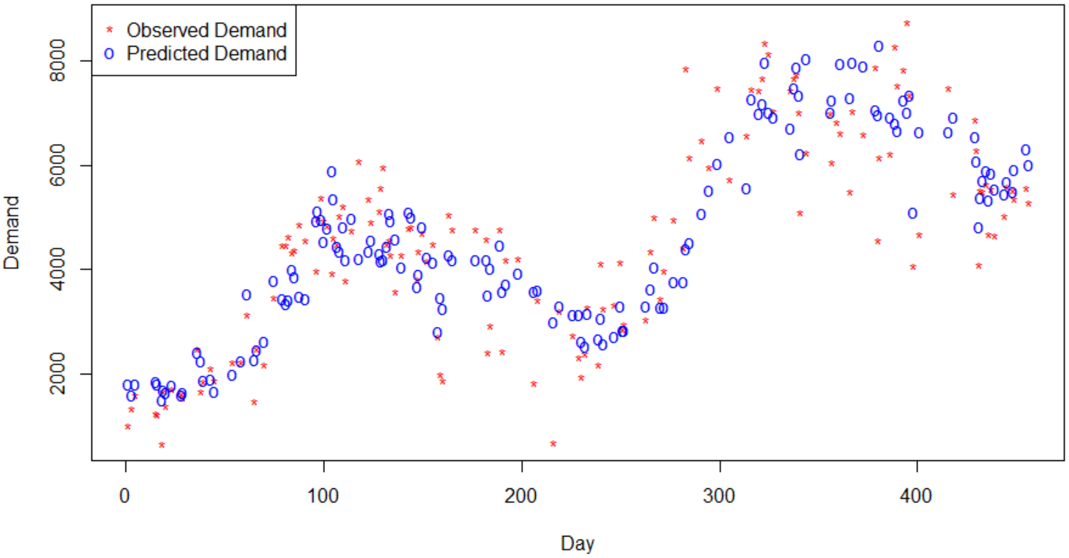
5. GLM: Gaussian (694.8407) ~lm(Total~Index+as.factor(season) +as.factor(holiday)+meanatemp+meanwindspeed+meanhumidity, data=dtrain)



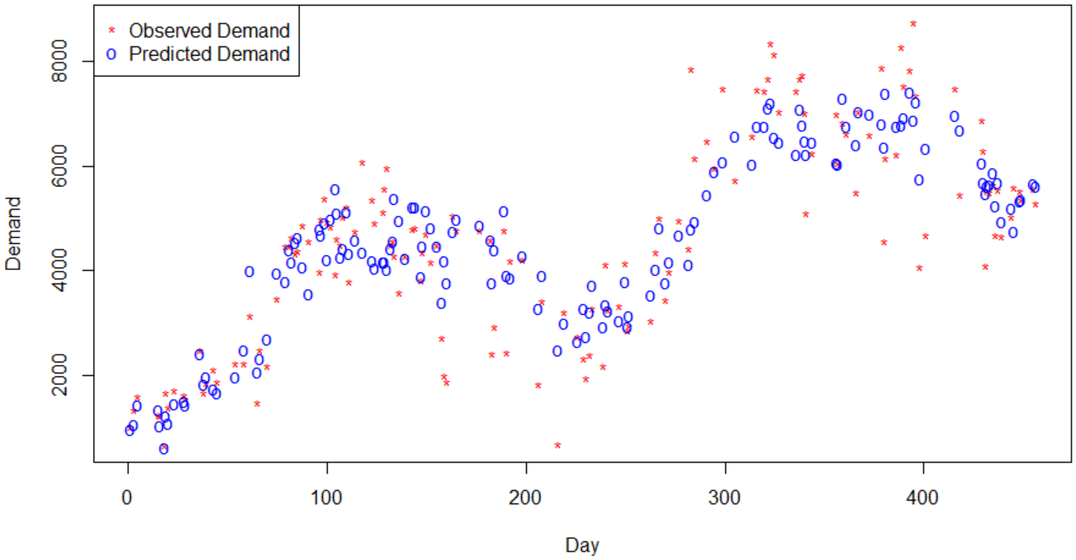
6. GLM: Poisson (814.7978)~lm(Total~Index+as.factor(season) +as.factor(holiday)+meanatemp+meanwindspeed+meanhumidity, data=dtrain)



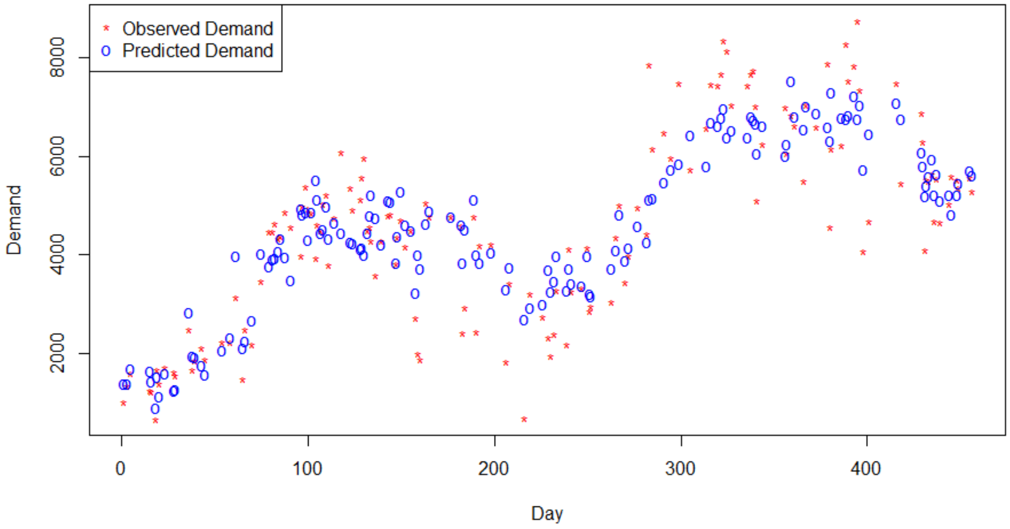
7. GLM: Negative Binomial (919.7653)~lm(Total~Index+as.factor(season) +as.factor(holiday)+meanatemp+meanwindspeed+meanhumidity, data=dtrain)



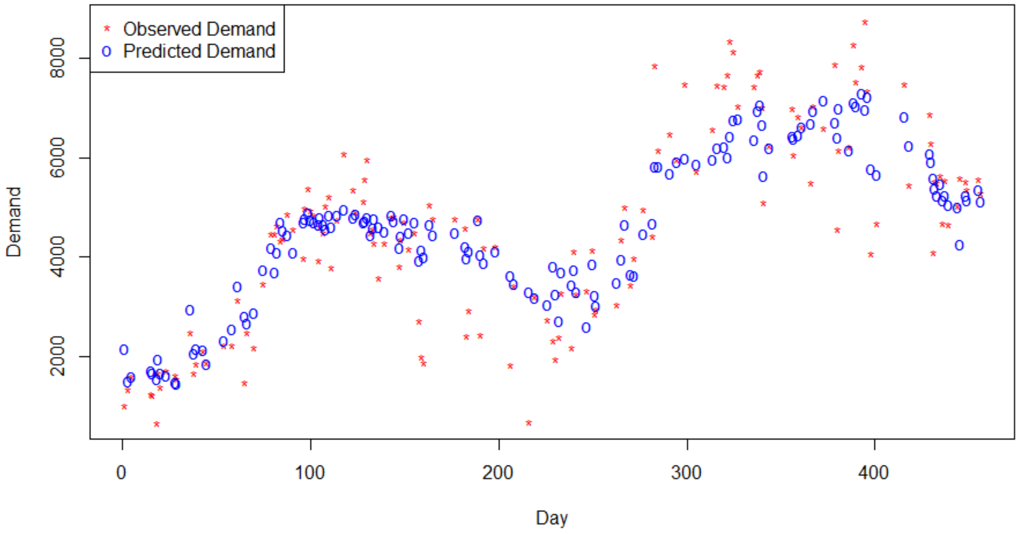
8. Stepwise Regression (638.1186)~lm(Total~Index+year+month+day+season+holiday +workingday+meanatemp+maxatemp+minatemp+sdatemp+meanhumidity+maxhumidity+minhumidity+sdhumidity+meanwindspeed+maxwindspeed+minwindspeed+sdwindspeed)



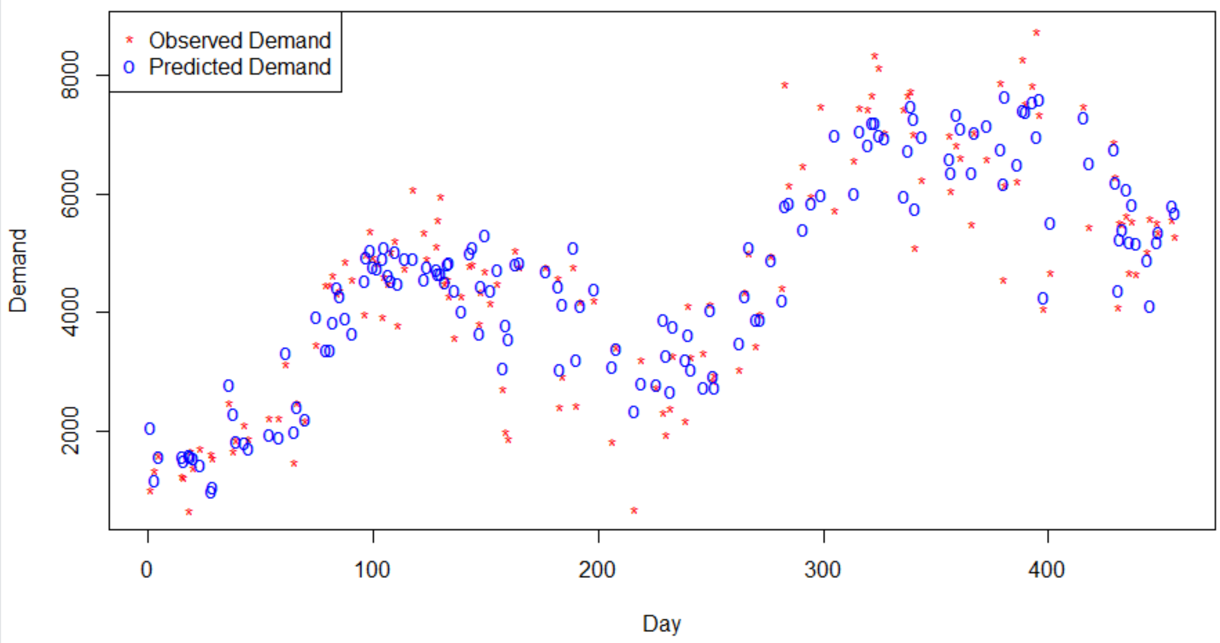
9. LASSO Regression (650.073)
lm(Total~Index+year+month+day+season+holiday+workingday+meanatemp+maxatemp+minatemp+sdatemp+meanhumidity+maxhumidity+minhumidity+sdhumidity+meanwindspeed+maxwindspeed+minwindspeed+sdwindspeed)



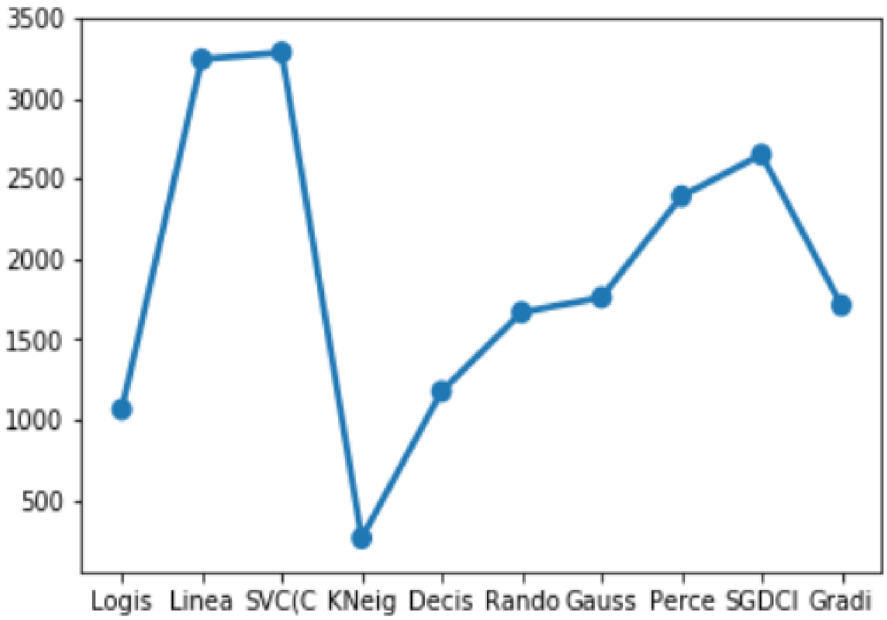
10. Random Forest (578.7885)~lm(Total~Index+year+month+day+season+holiday +workingday+meanatemp+maxatemp+minatemp+sdatemp+meanhumidity+maxhumidity+minhumidity+sdhumidity+meanwindspeed+maxwindspeed+minwindspeed+sdwindspeed)



11. SVM (572.9129)
lm(Total~Index+year+month+day+season+holiday+workingday+meanatemp+maxatemp+minatemp+sdatemp+meanhumidity+maxhumidity+minhumidity+sdhumidity+meanwindspeed+maxwindspeed+minwindspeed+sdwindspeed)

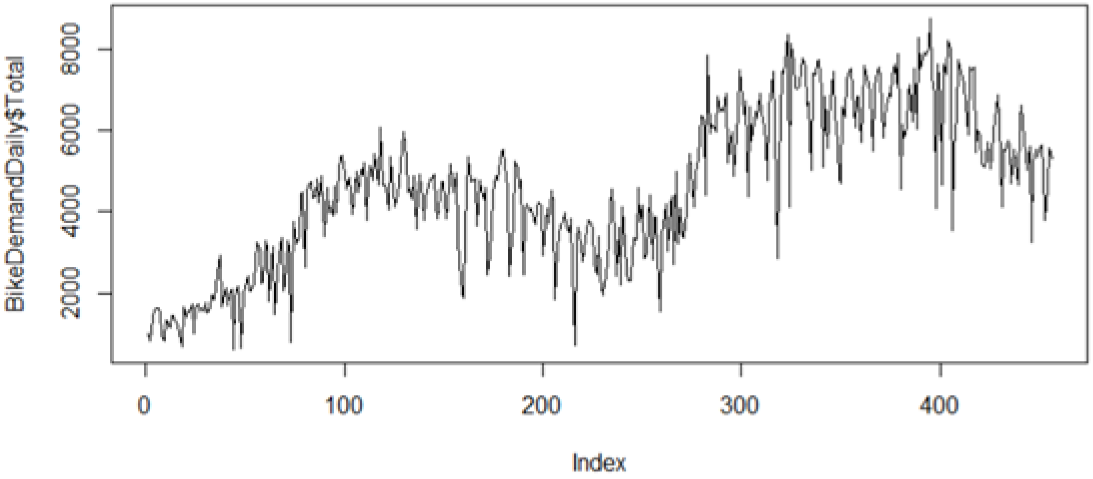


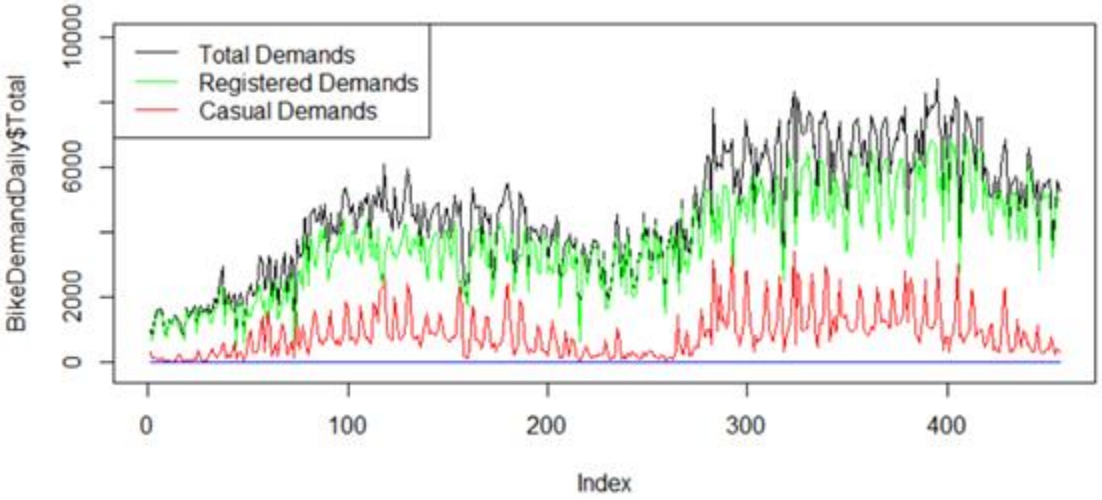
Appendix 2. Comparison of different kinds of machine learning algorithm in Python with Cross-Validation. (Logistic Regression, Linear SVM, SVM, K Nearest Neighbor, Decision Tree, Random Forest, Gaussian Naive Bayes, SGD, Gradient Boosting)

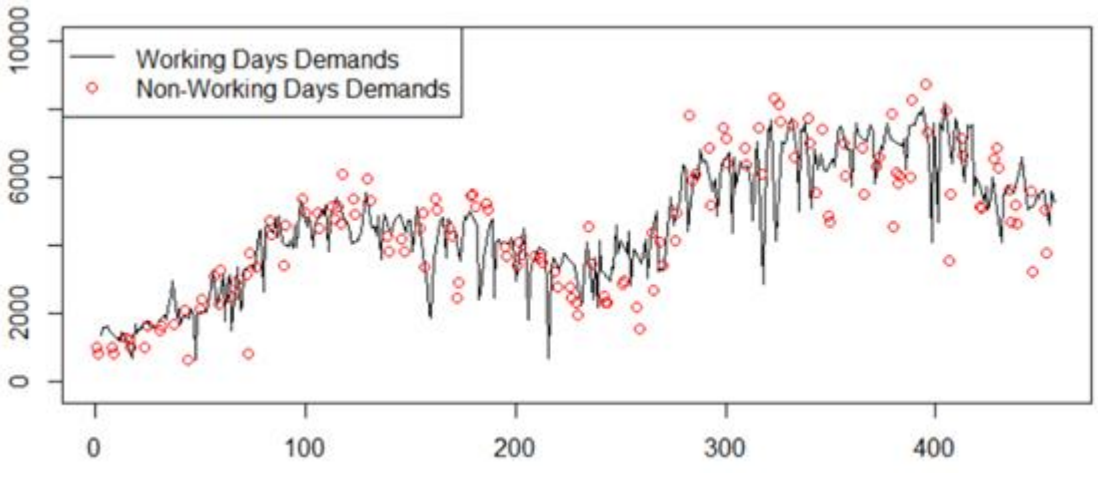


```
[[LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False), 1063.2251918561703],
 [LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
          intercept_scaling=1, loss='squared_hinge', max_iter=1000,
          multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
          verbose=0), 3242.3744936717335],
 [SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
          decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
          max_iter=-1, probability=False, random_state=None, shrinking=True,
          tol=0.001, verbose=False), 3286.3531231091383],
 [KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
          metric_params=None, n_jobs=1, n_neighbors=5, p=2,
          weights='uniform'), 262.04991576356593],
 [DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
          max_features=None, max_leaf_nodes=None,
        min_impurity_decrease=0.0, min_impurity_split=None,
          min_samples_leaf=1, min_samples_split=2,
        min_weight_fraction_leaf=0.0, presort=False, random_state=None,
          splitter='best'), 1175.9073226134485],
 [RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
          max_depth=None, max_features='auto', max_leaf_nodes=None,
        min_impurity_decrease=0.0, min_impurity_split=None,
          min_samples_leaf=1, min_samples_split=2,
        min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
          oob_score=False, random_state=None, verbose=0,
          warm_start=False), 1667.796075703156],
 [GaussianNB(priors=None), 1760.2626153285069],
 [Perceptron(alpha=0.0001, class_weight=None, eta0=1.0, fit_intercept=True,
          max_iter=None, n_iter=None, n_jobs=1, penalty=None, random_state=0,
          shuffle=True, tol=None, verbose=0, warm_start=False),
  2390.2526154555094],
 [SGDClassifier(alpha=0.0001, average=False, class_weight=None, epsilon=0.1,
          eta0=0.0, fit_intercept=True, l1_ratio=0.15,
          learning_rate='optimal', loss='hinge', max_iter=None, n_iter=None,
          n_jobs=1, penalty='l2', power_t=0.5, random_state=None,
          shuffle=True, tol=None, verbose=0, warm_start=False),
  2649.49002929374],
 [GradientBoostingClassifier(criterion='friedman_mse', init=None,
               learning_rate=0.1, loss='deviance', max_depth=3,
          max_features=None, max_leaf_nodes=None,
         min_impurity_decrease=0.0, min_impurity_split=None,
          min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0, n_estimators=100,
          presort='auto', random_state=None, subsample=1.0, verbose=0,
          warm_start=False), 1711.4520342182411]]
```

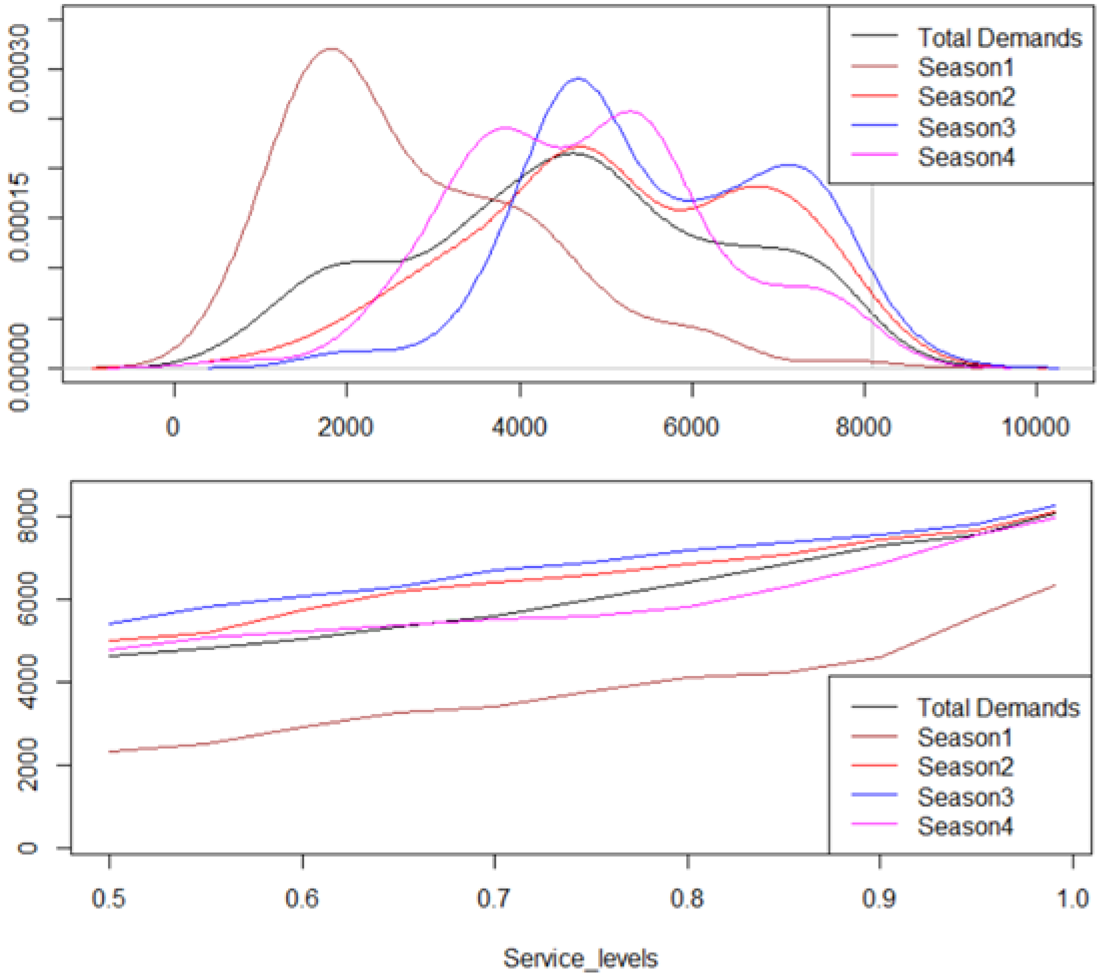
Real Total Demands



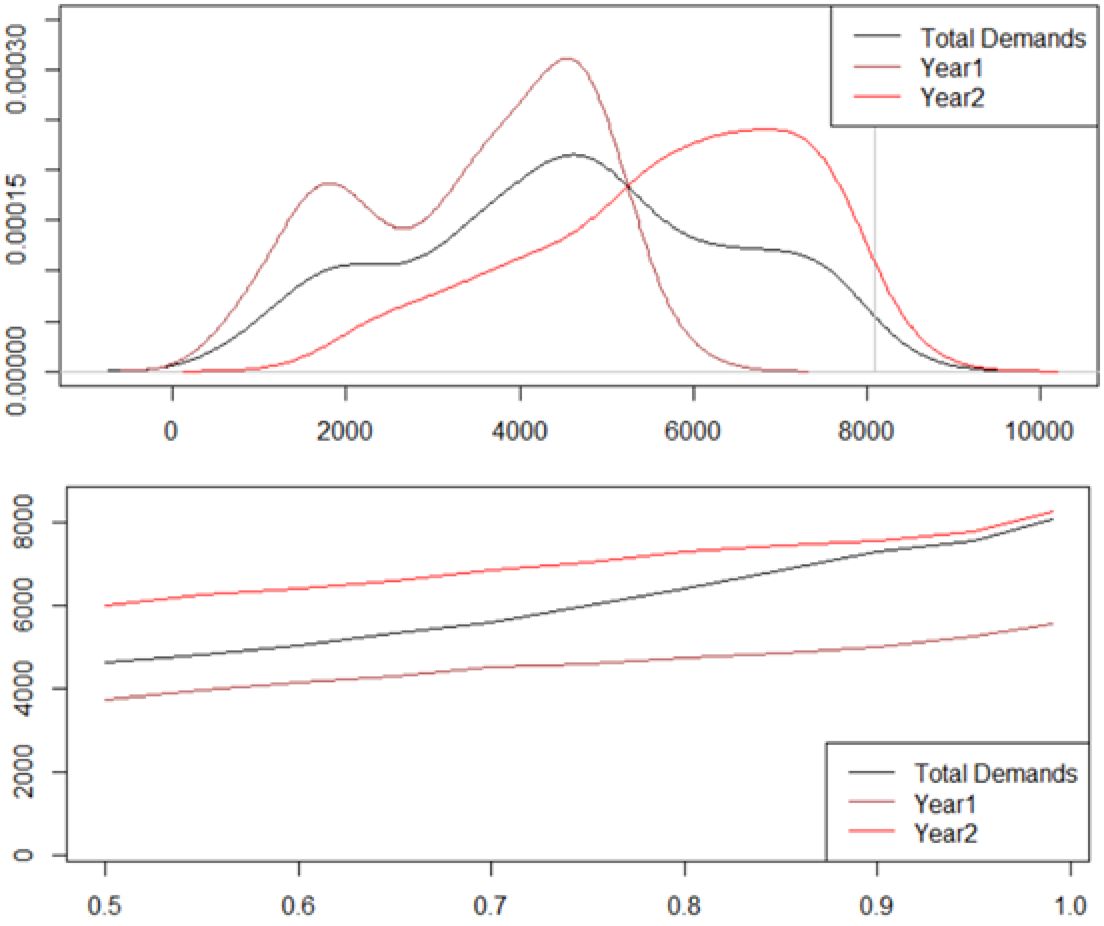




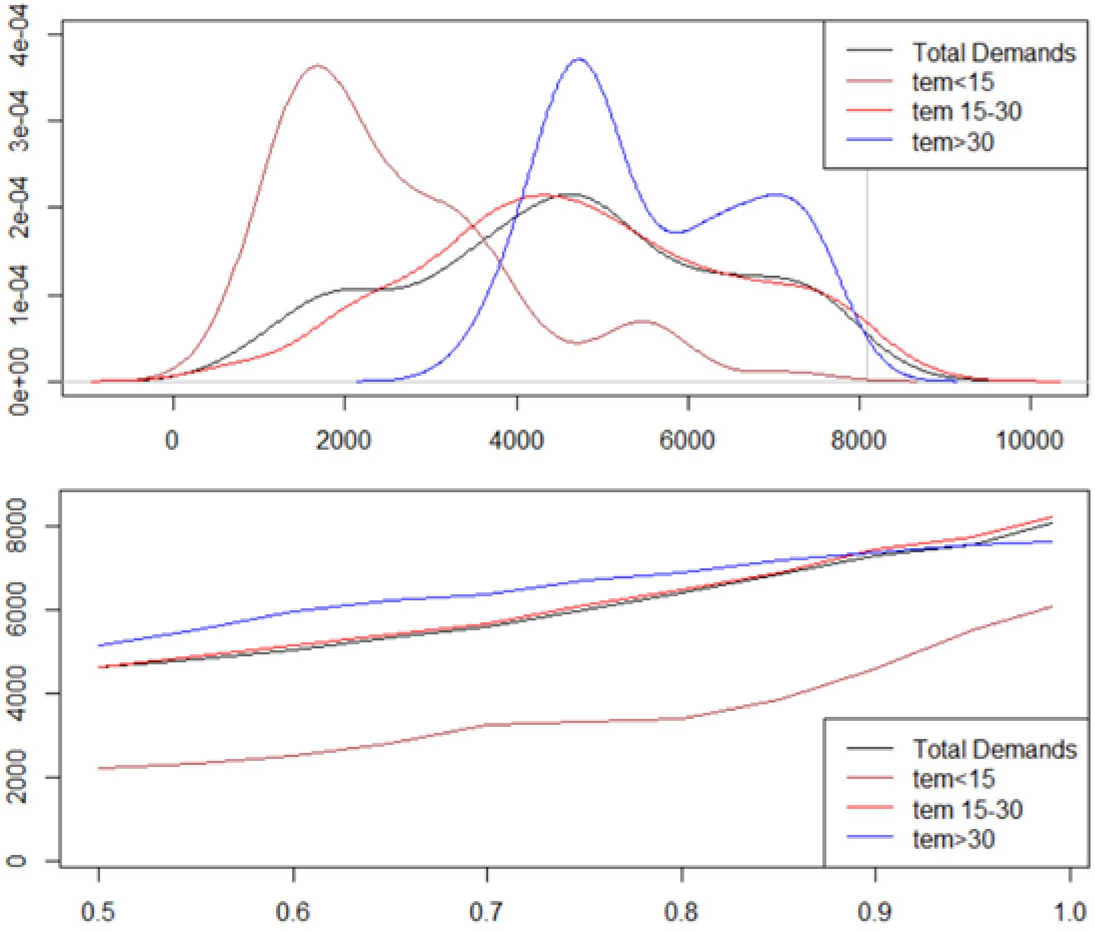
Distribution and Cost by season



Distribution and Cost by year



Distribution and Cost by temperatures


0e+00
1e-04
2e-04
3e-04
4e-04
Total Demands
tem<15
tem 15-30
tem>30
Total Demands
tem<15
tem 15-30
tem>30