Demystifying Forecast of Bike Sharing

Manna Lin 704434171, Peipei Zhou 204176631

CS260 Machine Learning Algorithms, December 2015

1 Abstract

Bike sharing is a new form of sustainable public mobility. There are more than 500 bike sharing systems around the world currently. A common issue in bike sharing systems falls on strategy improving and operation planning to optimize the uses of bike sharing. A recent competition on Kaggle "Bike Sharing Demand" provides us valuable data for predicting and online ranking system for evaluation. In this project, our goal is to **demystify** the forecast of hourly bike rental demand in the Capital Bikeshare program in Washington, D.C., as shown in Fig. 1. Experimental results show that data preprocessing reduce the Root Mean Squared Logarithmic Error (RMSLE) score by up to 58% and improve the average ranking from 2901^{th} to 2432^{th}, best ranking from 2100^{th} to 90^{th}. Moreover, modeling optimization in the parameters tuning and models blending further improve the best ranking from 90^{th} to 65^{th}. We concluded that data preprocessing and modeling optimization are among key factors in improving learning results in similar forecast problems.

In this report, data and evaluation methods are first introduced in Section 2. Baseline machine learning models and results are shown in Section 3. Data preprocessing including feature engineering and logarithmic scaling are explained in details in Section 4. Moreover, modeling optimization including model selection, model tuning and model blending are illustrated in Section 5. Conclusion are made in Section 6. In summary, the report shows machine learning algorithms that achieve good learning results in forecast problem. More importantly, the report demystifies how algorithms are selected, tuned and fit to achieve the satisfactory learning outcome.

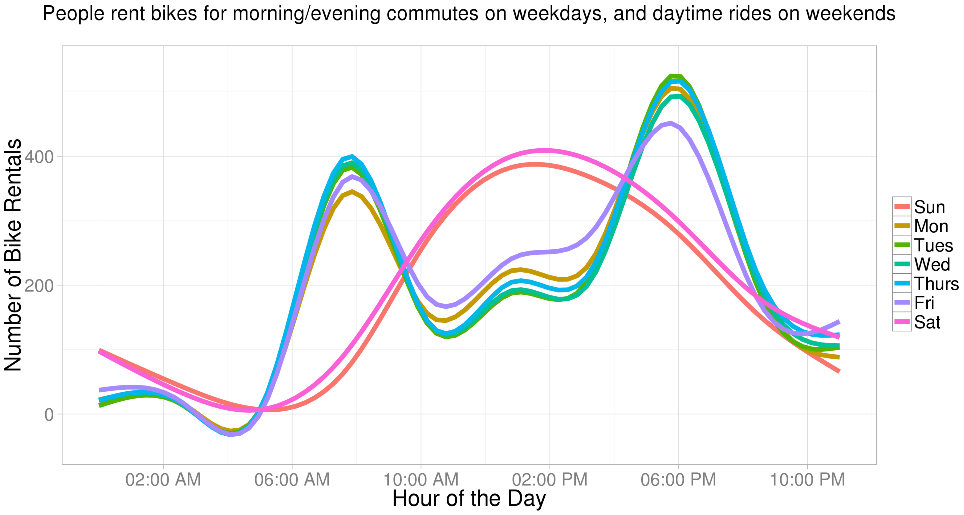


Figure 1: The hourly rental data predicted in a week.

2 Data and Evaluations

The data is the hourly bike rental data that spans for two years. The data features include binary, categorical and quantitative data, which are listed in Table 1. The training data is the rental data from the first 19 days of each month while the test data is the rest days of the month. The task is to predict the total rental bikes in each hour in the test set, based on the prior information.

Table 1: Data fields

	data fields	description
	datetime	hourly date + timestamp
binary	holiday	whether the day is considered a holiday
	workingday	whether the day is neither a weekend nor holiday
categorical	season	1 = spring, 2 = summer, 3 = fall, 4 = winter
	weather	1: Clear,2: Mist Cloudy,3: Light Snow, 4: Heavy Rain
quantitative	temp	temperature in Celsius
	atemp	“feels like” temperature in Celsius
	humidity	relative humidity
	windspeed	wind speed
	casual	number of non-registered user rentals initiated
	registered	number of registered user rentals initiated
	count	number of total rentals

The results are evaluated on the Root Mean Squared Logarithmic Error (RMSLE). The RMSLE score is calculated as Eq. 1, where n is the number of hours in the test set, p_i is your predicted count, a_i is the actual count and $\log x$ is the natural logarithm.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2} \tag{1}$$

In addition, the results can be saved in .csv file format and submitted in the Kaggle online system to get the ranking compared to all the other submissions. In the following sections, both ranking and RMSLE score are used to evaluate the algorithms.

3 Baseline

Firstly, nine popular learning algorithms are trained on the raw train data and predicted on the raw test data. The RMSLE scores and rankings are shown in Table 2 and Fig. 2.

Table 2: RMSLE scores and ranking of baseline algorithms

algorithm	rmsle score	ranking
linear regression	1.27	2936
stocastic gradient descent	1.29	2956
ridge regression	1.27	2934
random forest	0.67	2432
extremely randomized forest	0.66	2417
support vector machines	3.00	3201
naive bayes	2.14	3190
bernoulli naive bayes	3.08	3206
adaboost	1.12	2839
average	**1.61**	**2901**

The average score is 1.61 and ranking is 2901 out of 3252 teams. The baseline models rank at the bottom of the submitted list. In the following sections we show the improvement after data preprocessing and model optimization techniques.

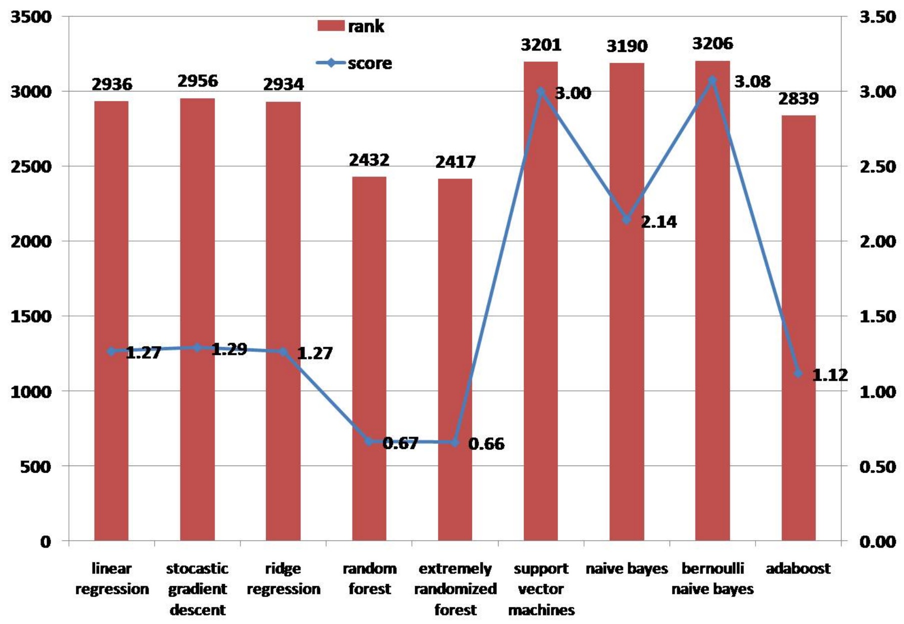


Figure 2: RMSLE scores and ranking of baseline algorithms

4 Data Preprocessing

To improve the learning results, we first make use of random forest feature modeling to give out the relative importance factor for different features, which guides us in examining top five features. Then we carefully profile some features to determine how to transform the features into new features. Lastly, logarithmic scaling is applied to further improve the results.

4.1 Feature Engineering

Random Forest importance metric in the Random Forest models has characteristic that correlated predictors get low importance values. As shown in Fig. 3, hour, workingday are among the most of importance features. Humidity and temperature have less importance.

4.1.1 Registered and Casual Rental Counts Versus Workingday

In the training data, the total counts of bike rentals differ from registered users and casual users. As shown in Fig. 4a, for registered users, from Monday to Friday, there will be two peaks of bike use within 24 hours. The first peak is from 7am to 9am. The other is from 4pm to 7pm. This matches with the traffic peak hours of people who ride bikes to the company in the morning and come home in the evening. In contrast, in Fig. 4b, bike rental count from casual users shows a distinct pattern. As the profiling results show, casual users tend to rent bikes from 10am to 7pm especially on Saturday and Sunday. A new feature is created named “peak” to capture these patterns.

4.1.2 Registered and Casual Rental Counts Versus Humidity

The feature humidity alone does not show great importance in Fig. 3. But we found that humidity and workingdays has strong correlations. In Fig. 5a, we plot the registered bike rentals in 24 hours on weekdays and weekends. The color of each point represents its humidity. In the morning, most of the bikes are rent when humidity is around 75. In contrast, in the afternoon, more bikes are rent when the humidity is around

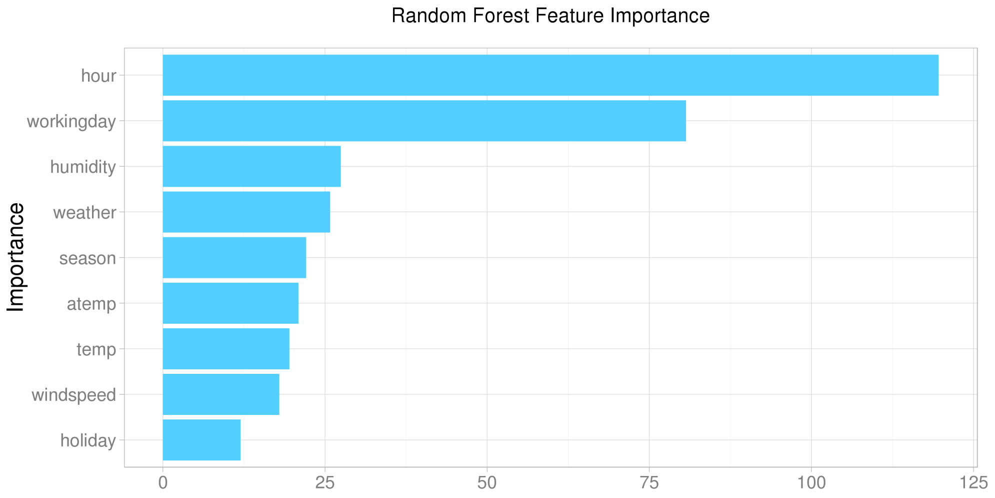
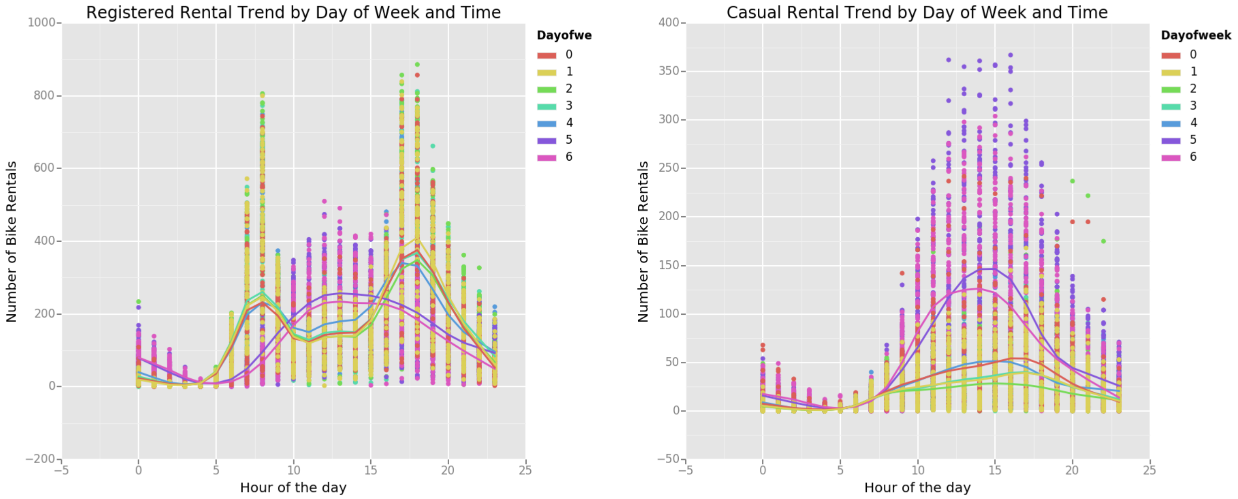


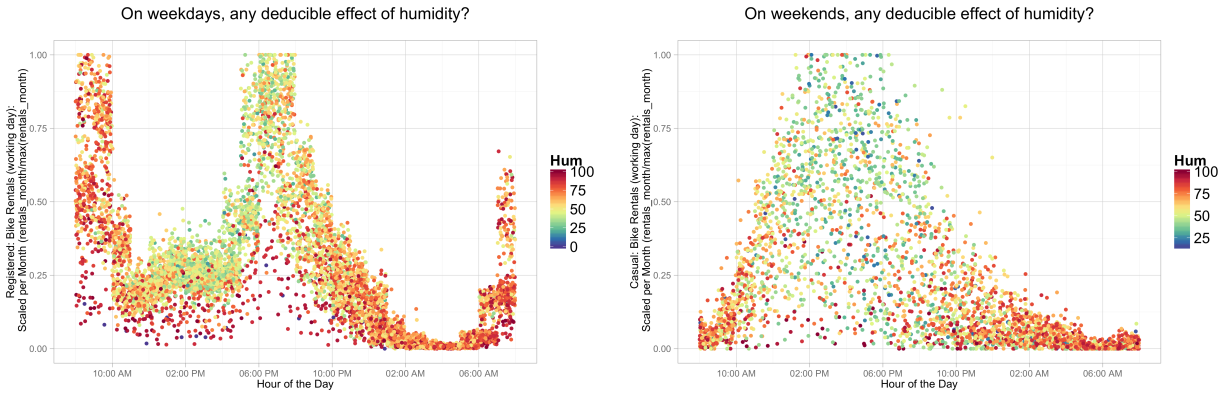
Figure 3: Random Forest importance for features



(a) Registered users bike rental within a week

(b) Casual users bike rental within a week

Figure 4: Registered and casual users bike rental whthin a week



(a) Weekday registered rental versus humidity

(b) Weekend casual rental versus humidity

Figure 5: Registered and casual rental counts versus humidity

40. Compared with Fig. 5b, which shows weekend rental count by casual users, there is no strong correlation between humidity and weekends. Therefore, Another feature named "sticky" is created to capture feature correlation of humidity and workingdays.

4.2 Logarithmic Scale Transform

After feature engineering, 5 models that give out best results are picked and results are shown in Fig. 6. In this figure, we compare the RMSLE scores of models when dataset are without feature processing, with feature processing and feature processing followed by logarithmic scaling (after data preprocessing, feature "temp", ""atemp", "umidity" "indspeed" are normalized through Min-Max transform and logarithmic scale transform). The models are trained on the train dataset after preprocessing and "casual" and "registered" counts are predicted. The two counts are added up as the total counts and submitted to get the RMSLE scores. The scores are listed in Table 3 and RMSLE scores reduce by a range from 6.28 % to 58.74%.

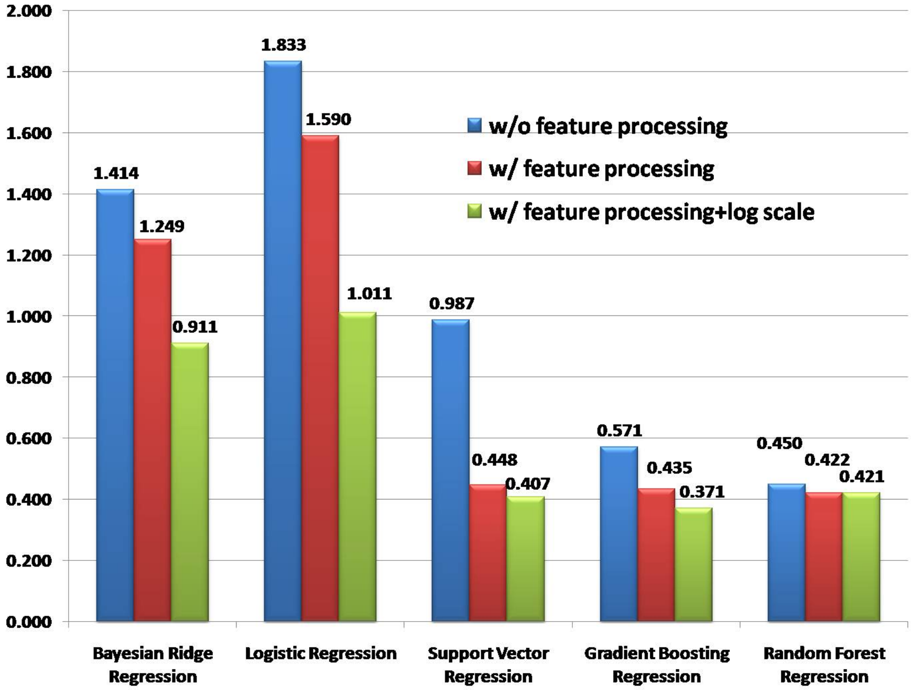


Figure 6: RMSLE scores before and after data preprocessing.

Table 3: RMSLE scores

	w/o feature processing	w/ feature processing	w/ feature processing+log scale	rmsle reduce
Bayesian Ridge Regression	1.414	1.249	0.911	35.54%
Logistic Regression	1.833	1.590	1.011	44.81%
Support Vector Regression	0.987	0.448	0.407	58.74%
Gradient Boosting Regression	0.571	0.435	0.371	34.92%
Random Forest Regression	0.450	0.422	0.421	6.28%

5 Model Optimization and Experimental Results

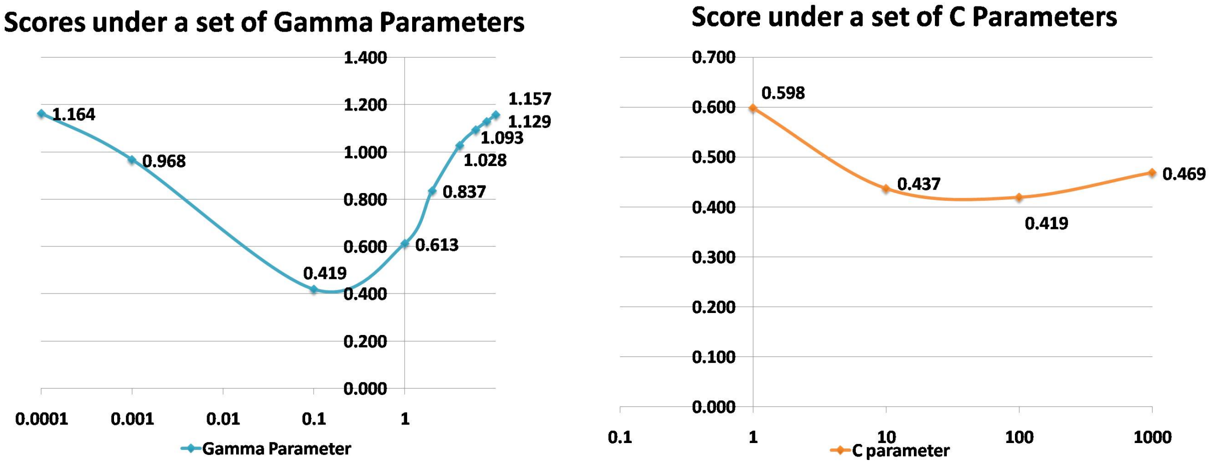
In this section, model optimization including parameter tuning and model blending are performed to further improve the ranking and scores.

5.1 Model Parameters Fine Tuning

We use SVR parameters tuning as an example shown in Fig. 7. We sweep gamma parameters in [10, 8, 6, 4, 2, 1, 0.1, 0.001, 0.0001] and C parameters in [1, 1e1, 1e2, 1e3] to get the RMSLE scores. The best results are achieved when gamma is 0.1 and C is 1e2. Other models parameters are also tuned, for example, n_estimators, min_samples_split, max_depth of Random Forest Regressor (RFR).

5.2 Model Blending

Within five models, three best models, Random Forest Regressor (RFR), Gradient Boosting Regressor (GBR), Support Vector Regressor (GBR) are selected to do model blending. As shown in Eq. 2, y is



(a) Scores versus Gamma Parameters

(b) Scores versus C Parameters

Figure 7: Scores versus Gamma and Parameters

predicted data from model blending, y_1, y_2 are predicted data from two models. w is the weight for one model.

$$y = w \times y_1 + (1 - w) \times y_2 \tag{2}$$

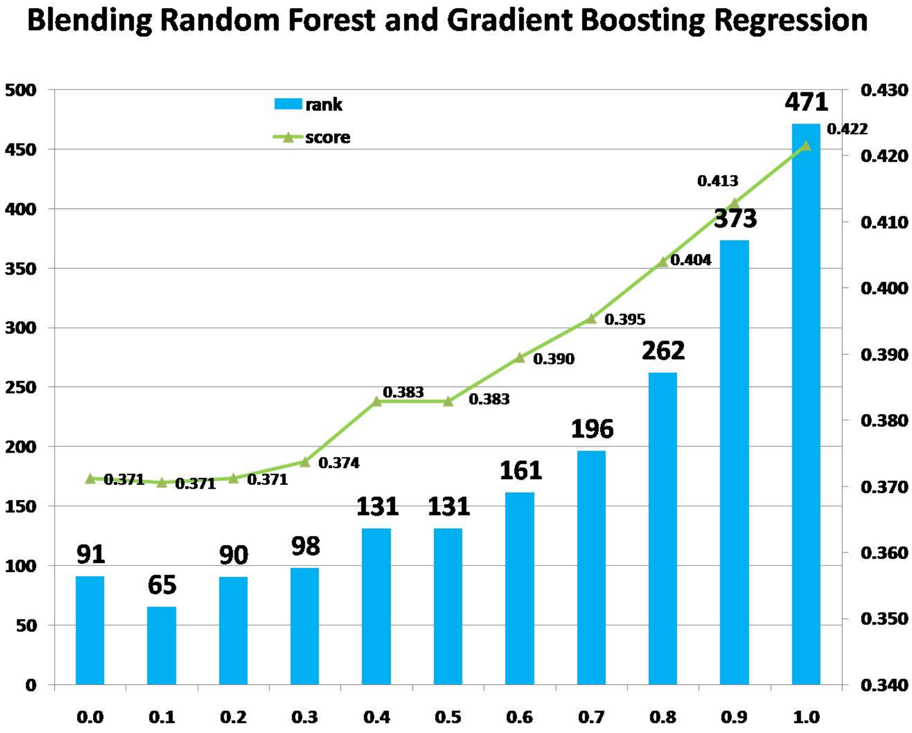


Figure 8: Rankings and scores of blending of RFR and GBR.

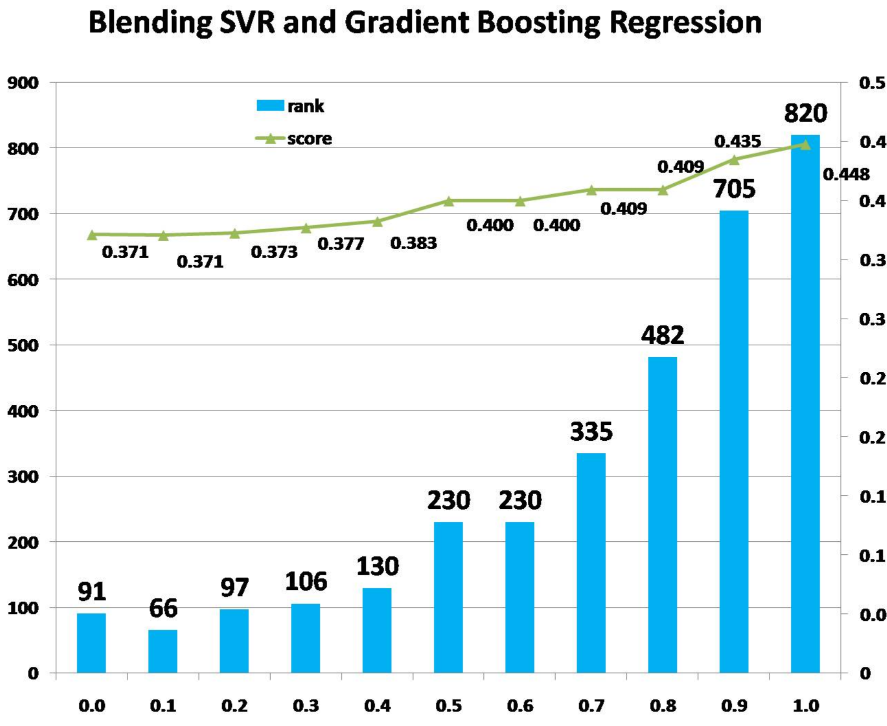


Figure 9: Rankings and scores of blending of SVR and GBR.

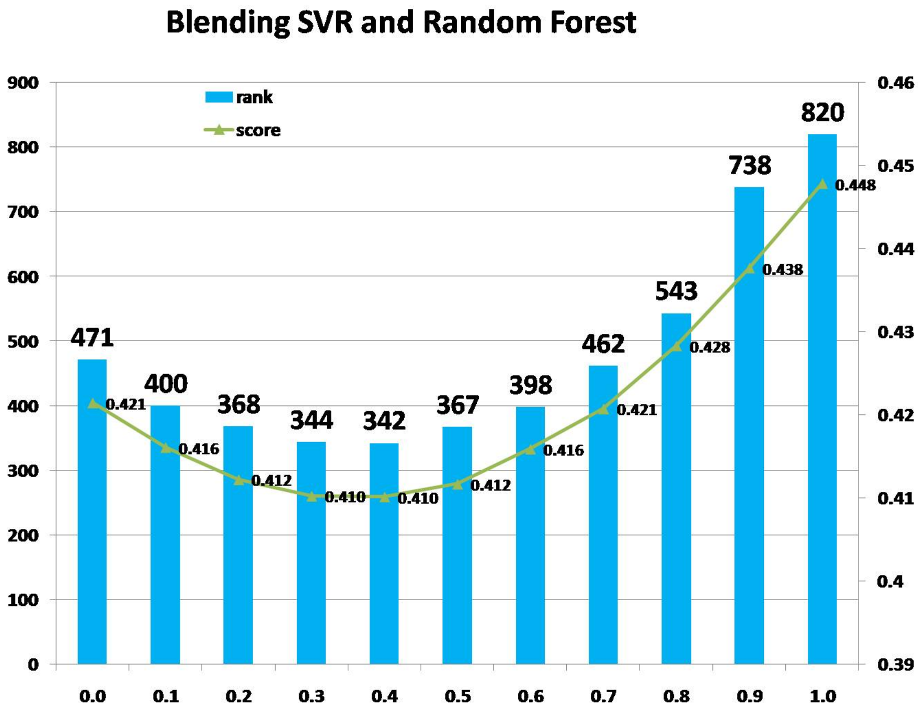


Figure 10: Rankings and scores of blending of SVR and RFR.

As shown in Fig. 8, 9 and 10, blending models will further improve the rankings and scores compared to each individual model. Finally, the model blending of Random Forest Regressor and Gradient Boosting Regressor with weight 0.1 and 0.9 ranks the highest and improves the ranking from 90^{th} to 65^{th}.

6 Conclusion and Future Work

In summary, the project report presents the baseline machine learning modelings and improves the models through data preprocessing and model optimization. The techniques that have been used here could shed light on similar forecast problems. In the future, more models are to be explored and run-time accuracy trade-off is one of the interesting topics that can be further studied.