Bike Sharing Demand Prediction Assignment

EB5102 Data Analytics

Lecturer: Dr. Barry Adrian Shepherd

Student ID	Name	E-mail
A0178551X	Choo Ming Hui Raymond	E0267862@u.nus.edu
A0178431A	Huang Qingyi	E0267742@u.nus.edu
A0178415Y	Jiang Zhiyuan	E0267726@u.nus.edu
A0178365R	Wang Jingli	E0267676@u.nus.edu
A0178329R	Wong Yeng Fai, Edric	E0267640@u.nus.edu
A0178371X	Yang Shuting	E0267682@u.nus.edu

Executive Summary

Results. Numerous predictive models were trained and optimised using Neural Network (NN), Random Forest (RF) and XGBoost (XGB). These models were used to predict bicycle demand for bike sharing company "Capital Bikeshare" using 2011 and 2012 demand logs. They outperformed the benchmark by 3.5% to 7.2% with XGB offering the best results, summarized below.

Overview of Model Performance

Model	Profit $ million	Cost $ million	Profit/Cost	RMSE (% Increase)	RMSE (Absolute Demand)
Benchmark	145.9	408.5	35.7%	-	-
Random Forest	162.6	415.1	39.2%	0.34	981
Neural Network	166.8	389.6	42.8%	0.32	980
XGBoost	167.7	391.3	42.9%	0.35	983
Ensemble - Blending	**168.7**	390.6	43.2%	0.33	949
Ensemble - Stacking	164.9	368.7	**44.7%**	0.36	1136

To further enhance performance, several ensembles were built using enumeration of the various models through either blending or stacking. It was concluded that the ensemble derived from blending NN and XGB models results in the best absolute profit margin while stacking using RF, NN and XGB resulted in the best profit-to-cost ratio.

However, it should be noted that the error statistic, Root Mean Squared Error (RMSE) for stacking was one of worst, suggesting that this ensemble approach is comparably less accurate in predicting absolute demand when compared to the rest.

Recommendations. While "Ensemble – Blending" offers the highest absolute profit, it is recommended for "Ensemble – Stacking" model to be adopted as it was able to achieve the best profit-to-cost ratio. From a business standpoint, this would translate to a better return of investment and is usually more desired.

Table of Contents

1. Introduction 1

2. Data Selection and Preprocessing 1

2.1 Data Understanding and Exploration 1

2.2 Feature Engineering 2

2.3 Data Preparation 3

3. Modeling 5

3.1 Basic Model – Random Forest 7

3.2 Basic Model – Neural Network 9

3.3 Basic Model – XGBoost 12

3.4 Ensemble 14

3.4.1 Ensemble (Blending) 14

3.4.2 Ensemble (Stacking) 15

3.5 Model Comparison Summary 17

3.6 What If Scenario 18

4. Business Performance 20

5. Conclusion and Recommendations 22

Annex A 1

Appendix I 1

1. Introduction

Cycling for short commutes has become immensely popular over the last few years as society becomes more environmental and health conscious. This has led to an increased demand for bike sharing platforms where bicycles are rented for short term use and where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. The accuracy to which a bike sharing company predicts short term user demand would often determine its viability as it affects the company's overall competitiveness and the return of investment that can be achieved.

In this study, we attempt to maximise profit for Capital Bikeshare, Washington D.C by training various predictive models (including the use of ensemble methods) to estimate daily bike demands in the near future based on data collected over the preceding days. To train and validate our models, bike sharing data extracted from their demand log from Jan 2011 to Dec 2012 was used. Thereafter, the revenue and incurred cost based on the predictive models were calculated and compared with the baseline estimation used by Capital Bikeshare, before we conclude and putting forth our recommendations.

2. Data Selection and Preprocessing

2.1 Data Understanding and Exploration

The dataset contains bicycle demand information over 24 months with 16 variables, where the data dictionary can be found in **Appendix I**. Data understanding and exploration was deemed necessary to effectively conduct subsequent data cleaning, feature engineering and model optimisation. The details are described in **Annex A** and some of the key observations are summarised below.

1. Total vs Registered vs Casual rentals. While the total rental (*cnt*) each day is the sum of registered and casual rental, the ratio of registered to casual rental was observed to differ greatly. This was likely because casual users accounted for around 20% of the total demand but had a much larger variance compared to registered users. It was also observed that there was an increasing trend subjected to some seasonal influences on the daily rental from 2011 to 2012.

2. Abnormal near-zero demand. There was a very significant drop for rental on 29th October 2012 which was found out to be due to extreme bad weather (hurricane *Sandy* in this case).

3. Erroneous Weather Classifications. As a result of the observation for hurricane Sandy, several non-obvious weather classification errors in the data were uncovered. While days with low bike rentals were associated mostly with weather classification of class 3, there were some days with class 3 weather where demand was significantly lower. Further online research confirmed that these were cases of extreme weather conditions which should be classified as class 4 instead of class 3 in the data.

4. Correlation Analysis. There were some numeric variables which are highly correlated (e.g. environment temperature and feel-like temperature) and due caution should be exercised depending on the modelling technique used.

5. Temperature, Humidity and Wind. It was observed that while temperature (*temp*) and "feel-like" temperature (*atemp*) were affected by season, humidity and windspeed were not. Furthermore, while these four variables are expected to be highly correlated, the data showed otherwise. Other than close correlation between temperature and "feel-like" temperature, there were no observable patterns with humidity and windspeed. This is postulated to be a result of using average daily weather data, which would result in any correlation being masked or distorted.

6. Interdependence of users. Demand for registered and casual users appears to behave differently and subsequent models would explore if modelling separately would yield better results. If so, the total demand of rental bikes can be derived by combining the separate predictions of registered and casual users.

2.2 Feature Engineering

Target redefinition. During preliminary data exploration and modeling, it was observed that tree-based regressions were not capable of predicting values that were outside of the range of the data used. What this means is that if the model was trained using a max demand of 4,000 rentals, it will not be able to predict any number above that. This is an issue as bike demand for this case follows an increasing trend which will eventually exceed the maximum bike rental number used in model training, severely limiting the usefulness of the model. Hence, to overcome this challenge, the target variable was redefined as the percentage increase in demand

instead. This transformation further converts the data into a stationary time series, eliminating trend related influences.

New features generation. With the redefinition of the target variable, several input variables were also converted from absolute numbers to percentage as well. Furthermore, to address issues of daily random fluctuation (i.e. noise), several variables were generated using data from a period (e.g. week or month) of time. It should be noted that D_{t-1} demand-related data was not used to predict D_t value since it would not be ready in time (i.e. by 4pm on D_{t-1}). As such, new variables which requires data from previous days (e.g. *cnt_inc_ratio_lag*', *cnt_inc_ratio_weekly* and *cnt_inc_ratio_monthly_avg*) were created based on D_{t-2} data. In addition, to reflect the patterns that might be offset by averaging the weekly data, min and max data (e.g. *cnt_inc_ratio_monthly_max* and *cnt_inc_ratio_monthly_min*) within the period were also created. The created features are documented in **Appendix I** and summarized in Table 1 below.

Table 1: Created Features

S/N	Variable	Type	Definition
1	cnt_inc_ratio	Target	% Increase of D_t (demand for time t) against D_{t-2}
2	cnt_inc_ratio_lag2	Numeric	% Increase of D_{t-2} against D_{t-4}
3	cnt_inc_ratio_weekly_avg	Numeric	% Increase of D_{t-2} against the average of D_{t-2} to D_{t-7}
4	cnt_inc_ratio_monthly_avg	Numeric	% Increase of D_{t-2} against the average of D_{t-2} to D_{t-30}
5	cnt_inc_ratio_weekly_max	Numeric	% increase of D_{t-2} against the maximum from D_{t-3} to D_{t-8}
6	cnt_inc_ratio_weekly_min	Numeric	% Increase of D_{t-2} against the minimum from D_{t-3} to D_{t-8}
7	temp_inc	Numeric	% Increase of D_{t-2} against D_{t-4}
8	cnt_avg_aheadWeek	Numeric	Demand for preceding week's average for D_{t-2} to D_{t-7}
9	cnt_avg_ahead3days	Numeric	Preceding 3 days average demand from D_{t-2} to D_{t-4}
10	cnt_avg_aheadMonth	Numeric	Preceding month average from D_{t-2} to D_{t-31}
11	cnt_median_LastWeek	Numeric	Median demand of preceding week from D_{t-2} to D_{t-7}
12	cnt_lastWeekday	Numeric	Demand of the same day last week, D_{t-7}

2.3 Data Preparation

Test and Training Data Split. As this is a set of time-series data, sequence was assessed to be important. Hence, the training subset was defined to be the 1st year (from Jan to Dec 2011)

while the test subset was defined to be the 2nd year (from Jan to Dec 12). As part of additional[1] exploration, models were also built using the first 18 months (Jan 11 to Jun 12) as training data to investigate if this improves overall model predictive power.

Outlier analysis. Outliers in this case were defined as datapoints that exist outside of ±3 standard deviations of the mean. In this dataset, there is only 1 outlier identified. That point correspond to the day hurricane Sandy hit Washington DC. This point was retained in the dataset but was subsequently removed in some of the models (e.g. Neural Network) to improve model performance.

Data Cleaning. Minimal cleaning is required as the dataset is generally complete. As explained in Section 2.1, several points appear to be misclassified. These were corrected accordingly as shown in Table 2 below.

Table 2: Extreme Weather Events in Washington DC, 2011-2012

Date	Weather	OriginalWeathersit	ModifiedWeathersit
26/01/2011	winter storm, heavy snow	3	4
16/04/2011	tornado outbreaks, thunderstorm	3	4
07/09/2011	Tropical Storm *Lee*	3	4
08/09/2011	Tropical Storm *Lee*	3	4
29/10/2011	heavy rain	3	4
07/12/2011	recorded breaking rain storm	3	4
22/04/2012	light rain, fog	2	3
29/10/2012	Hurricane *Sandy*	3	4
30/10/2012	Hurricane *Sandy*	3	4
26/12/2012	winter storm, heavy snow	3	4

Data Standardization. To ensure consistency during the modelling phase, standardisation was performed for all the continuous variables.

Dummy Coding. Categorical variables which were used for subsequent modelling were dummy coded as described in **Appendix I**. These were particularly required since many of these non-ordinal categorical variables are deemed to be important factors in predicting demand and cannot be used without dummy coding.

[1] These models were built after the initial work of training and optimizing individual models (RF, NN and XGB) and ensemble (blending and stacking). The initial work uses training / test data split of 1 year each.

3. Modeling

Modelling Approach. In Phase 1 of modelling, three algorithms were used, namely Random Forest (RF), Neural Network (NN) and XGBoost (XGB). For each algorithm, a model was built and optimised to achieve the best possible predictive power. In Phase 2, two ensembles were built, one by blending the models from the first phase, while the second one was built by stacking. Table 3 summarises the variables used in each model at the end of Phase 1.

Table 3: Variables for different models

S/N	Variable	Types	RF	NN	XGB
1	cnt_inc_ratio	Target	✓	✓	✓
2	workingday	Categorical	✓	✓	✓
3	weathersit	Categorical	✓	✓	✓
4	weekday	Categorical	✓	✓	
5	season	Categorical	✓	✓	✓
6	cnt_inc_ratio_lag2	Numeric	✓	✓	✓
7	cnt_inc_ratio_weekly	Numeric	✓	✓	✓
8	cnt_inc_ratio_monthly	Numeric	✓	✓	✓
9	cnt_inc_ratio_max	Numeric	✓	✓	
10	cnt_inc_ratio_min	Numeric	✓	✓	
11	temp_inc	Numeric	✓	✓	✓
12	cnt_avg_aheadWeek	Numeric		✓	✓
13	cnt_avg_ahead3days	Numeric		✓	✓
14	cnt_avg_aheadMonth	Numeric		✓	✓
15	cnt_median_LastWeek	Numeric		✓	✓
16	cnt_lastWeekday	Numeric		✓	✓

Modelling Objective. The objective of this study is to maximise profit, which is simplified to be the difference between the revenue (from bike rental to users) and cost (bike loan cost). From a modelling perspective, since revenue is closely tied to the accuracy in predicting bike demand, it is expected that the model with the best error statistics (i.e. lower RMSE) would also be the best in maximising profit. Hence, the modelling objective would be to train each model and optimise them by minimising RMSE.

Benchmark. The benchmark for comparison for the models is based on the current practice of the bike sharing company, which uses a simple Random Walk model. Through this, the company was able to achieve a profit of 146 million dollars at a 35.7% profit-to-cost ratio.

Combination of Bike Demand. As described in section 2.1 para 6, it was observed that the demand for registered and casual users were not correlated. Hence, initial models attempted to predict these two demands separately. However, the results were only marginally better than

the benchmark (approximately $20,000 improvement). Subsequent analysis suggest that this may be due to the large variance of casual user demand, which significantly affects the prediction accuracy for the demand of casual users. By combining registered and casual users, the effect of this variance is minimised as casual user forms a much smaller proportion of the total demand than registered users. As such, subsequent models reverted to using total demand as the target variable directly.

Model Selection Criteria. To measure the accuracy of the models, root mean squared error (RMSE) was used instead of mean absolute error (MAE) as it is better at amplifying the effects of large errors. The formula is given as follow,

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2}$$

In addition, it was noted that the effect of overestimation and underestimation of demand is different, with overestimation being costlier. This is because the cost of renting a bike is $3, while the profit (i.e. potential lost opportunity cost for renting one bike less than the demand) generated from renting a bike is only $1.

Hence, during model comparison, the absolute profit and the profit-to-cost ratio were included. It was expected that for models with similar RMSE, those which predict a lower demand but still achieve comparable profit will result in a higher profit-to-cost ratio. This will help to determine the best model from a business perspective as the optimal solution would be the one that offers the highest return of investment.

3.1 Basic Model – Random Forest

Overview. RF is an ensemble method that can be used to build predictive models for both classification and regression problems. It is designed to address the limitation of decision trees performing poorly on unseen data through randomly sampling of the training data as well as feature sets. Essentially, it attempts to decorrelate the trees and prune them by setting a stopping criterion for node splits. As RF can handle both categorical and numeric variables, there is no need for dummy coding during data preparation.

Modelling Approach. The first step (Step 1) was to generate a basic RF model using all the original variables available. To further improve the model, variables were added and dropped (Step 2 and 3) to obtain the best performance using the same hyperparameters from Step 1. From this, the most optimal RF model (*best_rf*) was derived and further tuning of the hyper parameters based on '*Gridsearch*' was implemented to further improve performance. Generally, the number of trees in the forest (*n_estimators*), the max number of features considered for splitting at each leaf node (*max_features*) and the max number of levels in each decision tree (*max_depth*) are considered to be the most important settings. Thus, optimisation in this case mainly involves turning these three parameters. After two iteration, the best performance of random forest model was observed, with a potential profit of 163 million dollars achieved in the test data (i.e. 2012). The results of the various modelling steps are summarized in Table 4.

Table 4: Overview of RF Modelling Step and Results

			RMSE			Hyperparameters		
Step	Model	Profit $ million	Percentage Increase	Absolute Demand	#of features	n_estimators	max_depth	max_features
-	benchmark	145	-	-	-	-	-	-
1	raw	156	0.44	1468	15	200	10	auto
2	feature_selection	159	0.41	1273	8	200	10	auto
3	best_rf	161	0.38	1150	10	200	10	auto
4	first_grid	162	0.38	1108	10	150	10	sqrt
5	**second_grid**	**163**	**0.37**	**1015**	**10**	**100**	**15**	**sqrt**

Figure 1 below illustrate the model performance by comparing predicted percentage increase against the actual increase in the test data. This is done by plotting predicted increase (x-axis) against the actual values (y-axis). For ease of comparison between predicted and actual values,

the results were transformed back into absolute numbers as shown in Figure 2. From Figure 2, it is noted that there is generally a good fit between prediction and actual results.

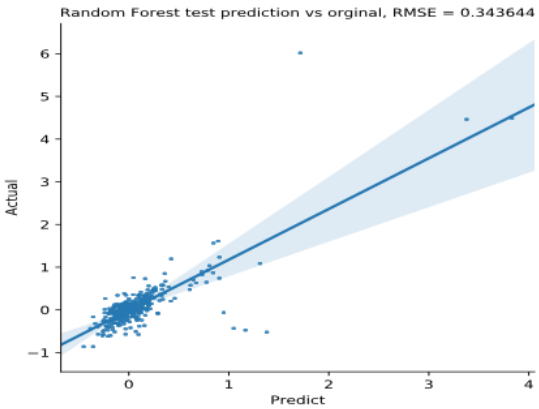


Figure 1: Actual vs Predicted % Increase

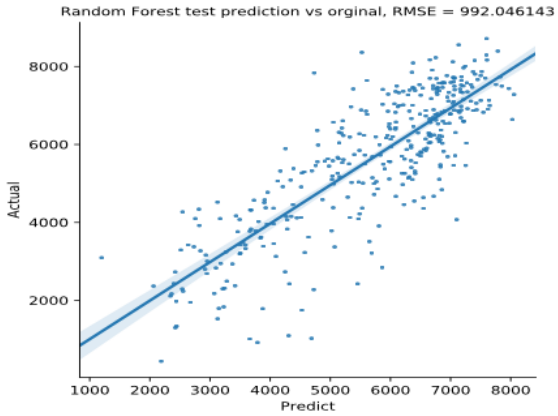


Figure 2:Actual vs Predicted Absolute Demand

The predicted demand of bikes (orange line, in absolute numbers) for 2012 were plot along with the actual daily demand (blue line) in Figure 3. It was observed that the model generated using RF had a similar pattern with the actual data, where the overall values of predicted values were slightly lower than the actual, suggesting an acceptable model. However, one point to note was that RF was not able to learn drastic drops fully as seen by the actual data having larger fluctuation then the predicted values.

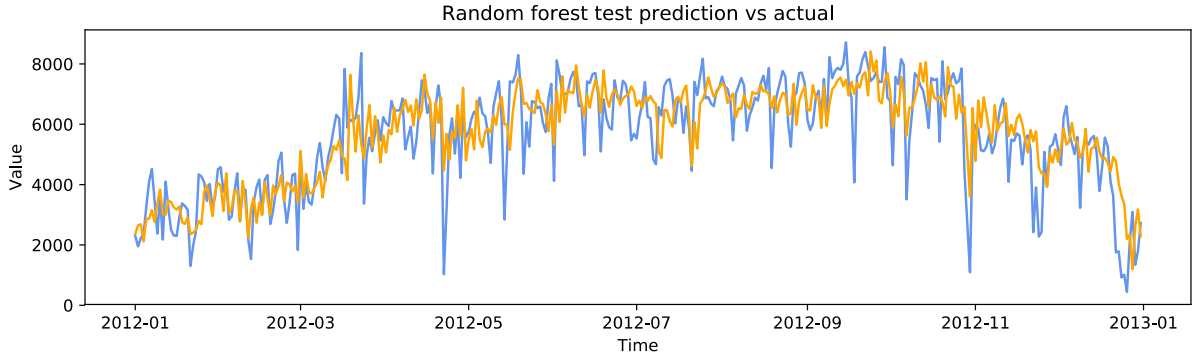


Figure 3: Predicted vs Actual Bike Demand over 2012

3.2 Basic Model – Neural Network

Overview. NN is a versatile machine learning method used for various applications. It is modelled as an interconnected network of nodes that simulate how the neurons in the brain function. During model training, the nodes take in the input variables, process them according to the activation function chosen, and output the predicted results that best match the actual data. This NN model would then be used for prediction based on future data inputs.

Modelling Approach. In our case, *Tensorflow-Keras* is used to construct the NN model with the structural configuration (including the number of nodes of the hidden layer) summarised in Table 5.

Table 5: Neural Network Structure and Parameters Configuration

Layer (Activation type)	Output Shape	Param
Input Layer	31	992
Dropout Layer 1 (relu)	31	0
Hidden Layer 2 (relu)	15	480
Output Layer	1	16
		Total params 1488

After several iterations, it was concluded an Epoch value of 20 and batch size of 3 should be used to train the NN model. Adaptive Moment Estimation (Adam), which allows faster convergence speed during model training, was used. Adam was applied to compute the adaptive step-wise learning rate, where the loss function was defined as Mean Square Error (MSE) to optimize the target. To avoid overfitting when training the model, a dropout layer was introduced and after several iterations, a dropout rate of 0.25 was used with the model loss shown in Figure 4 below.

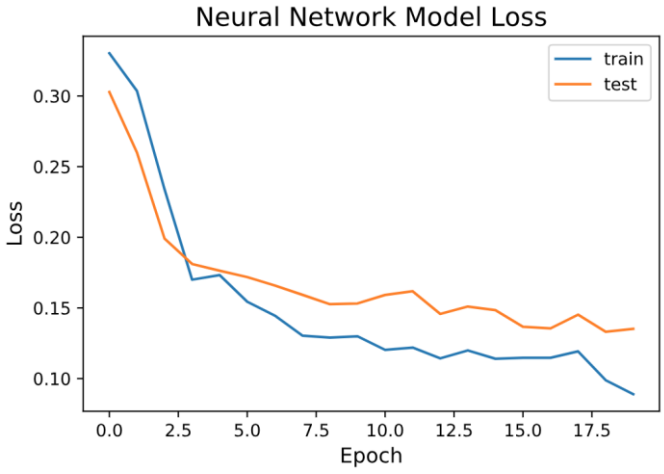


Figure 4: Model Loss when NN is applied to training and test data

Based on the dropout rate of 0.25, it can be seen that the model loss for both training and test data are fairly consistent over the entire Epoch range, with no signs of significant overfitting. The overall performance in terms of profit is summarised in Table 6 below.

Table 6: Results of Neural Network

Model	Profit $ million	Cost $ million	Profit/Cost	RMSE Percentage Increase	RMSE Absolute Demand
benchmark	145.9	408.5	35.7%	-	-
nn_best	166.8	389.6	42.8%	0.32	980

Similar to the earlier RF model, model performance is illustrated by comparing predicted percentage increase against the actual increase in the test data in Figure 5 below. The results were also transformed back into absolute numbers as shown in Figure 6. From Figure 6, it was noted that there was generally a good fit between prediction and actual results.

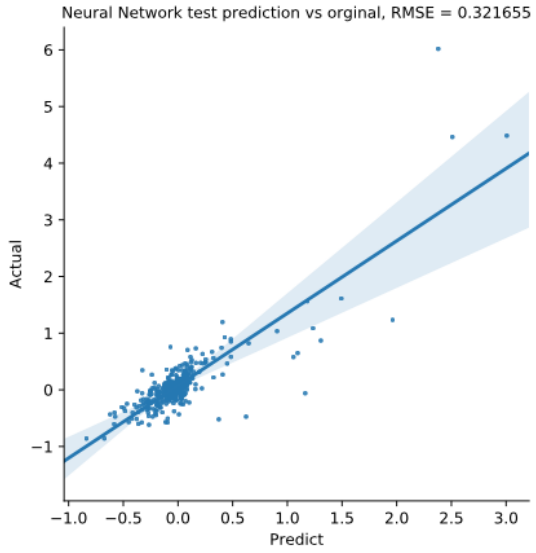


Figure 5: Actual vs Predicted % Increase

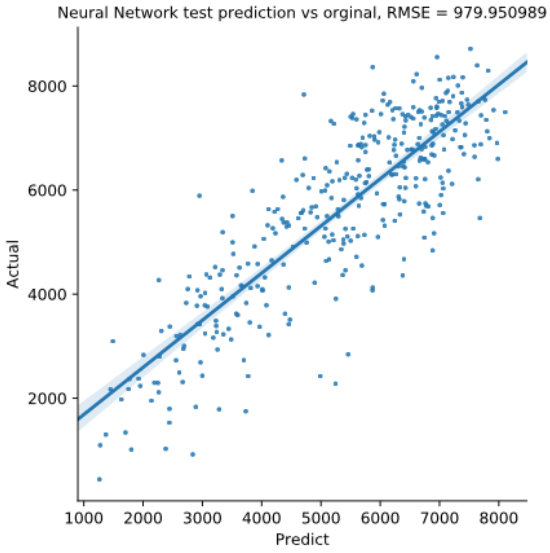


Figure 6: Actual vs Predicted Absolute Demand

From the learning curve of train and test, the loss of MSE reflected declining trend with development of epochs. The final prediction ratio RMSE was 0.32 and the absolute demand RMSE was about 979, which displayed the best performances among the single models.

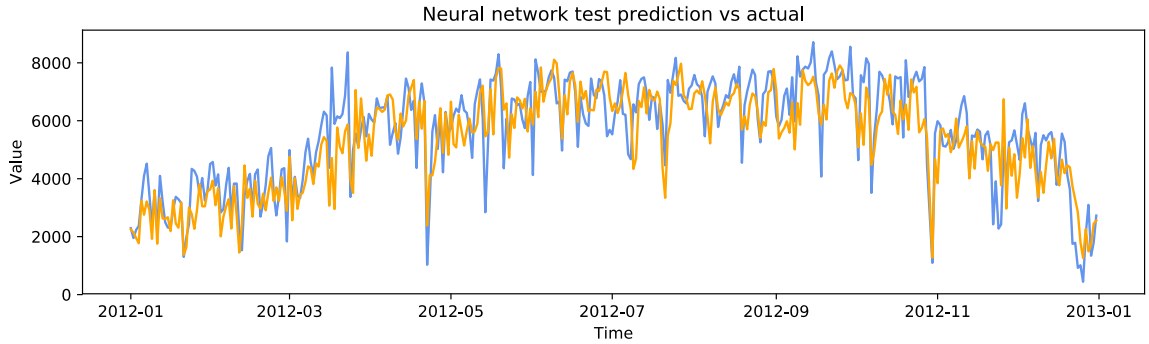


Figure 7: Predicted vs Actual Bike Demand over 2012

The predicted demand of bikes (orange line, in absolute numbers) for 2012 were plot along with the actual daily demand (blue line) in Figure 7. Compared to RF, it was noted that NN was much better at predicting large drop but tends to underestimate its predictions for peak values. This can be seen from the chart whereby the orange line was able to mimic the large drop in the actual data but was not able to do so for the peak data points.

3.3 Basic Model – XGBoost

Overview. Extreme Gradient Boosting (XGBoost) is an ensemble boosting method, which is based on the Gradient Boosted Decision Tree algorithm but uses a more regularised model formalisation to control over-fitting. Boosting refers to the technique of adding new models to correct the mistakes of previous models. Models are added until no further improvements can be made. It is called gradient boosting because it uses a gradient descent algorithm to minimise the loss when adding new models.

XGBoost is known for its performance, speed and flexible parameter tuning and the team wanted to try out if this algorithm would be able to provide a better predictive model and also forms the basis for a better ensemble in the next section.

Modelling Approach. Much exploration and iterations were done to obtain the optimal model. The details can be found in the Python codes that are submitted together with this paper. For the optimised model, the L2 regularization is implemented to reduce the overfitting in the training period and the parameters are tuned step by step as follows:

Step1: Fix learning rate and number of estimators for tuning tree-based parameters.
Step2: Maximum tree depth "*max_depth*" in range (3, 10, step = 1), *max_depth*=8.
Step3: Minimum loss reduction "*gamma*" in range (0, 0.5, step = 0.1), *gamma*=0.1.
Step4: Tuning "*subsample*" in range (0.5, 1, step = 0.01) and "*colsample_bytree*" in range (0.6, 1, step = 0.1), *subsample*=0.51, *colsample_bytree*=0.9.
Step5: Tuning Regularization Parameters "*alpha*" in set (0.001, 0.01, 0.1, 1, 10), *alpha*=1.
Step6: Tuning "*reg_lambda*" in the range (1, 10, step = 1), *reg_lambda*=9.
Step7: Reduce the Learning rate "*learning_rate*" to 0.09.

The overall results are shown in Table 7 and it was concluded that XGBoost was indeed able to outperform both RF and NN in maximising profit, although it has a slightly higher RMSE when compared to both RF and NN.

Table 7: Results of XGBoost

Model	Profit $ million	Cost $ million	Profit/Cost	RMSE Percentage Increase	RMSE Absolute Demand
benchmark	145.9	408.5	35.7%	-	-
xgb_best	167.7	391.3	42.9%	0.35	983

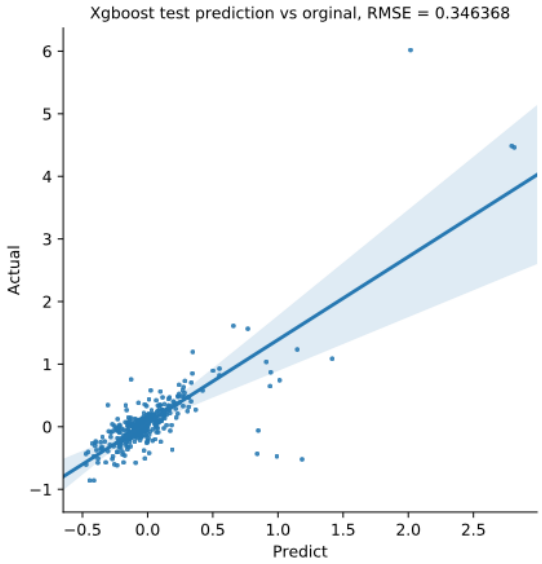


Figure 8: Actual vs Predicted % Increase

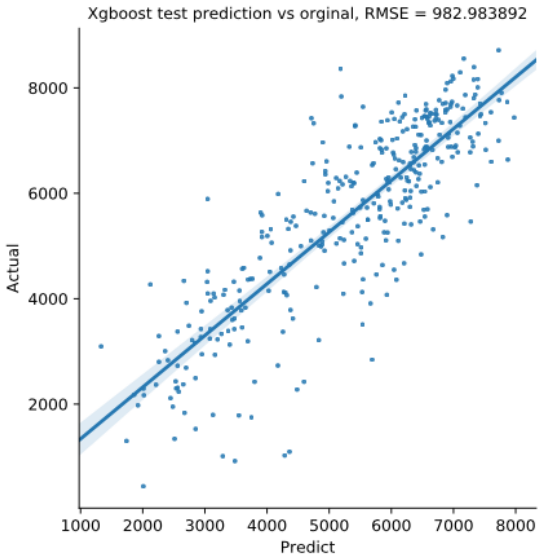


Figure 9: Actual vs Predicted Absolute Demand

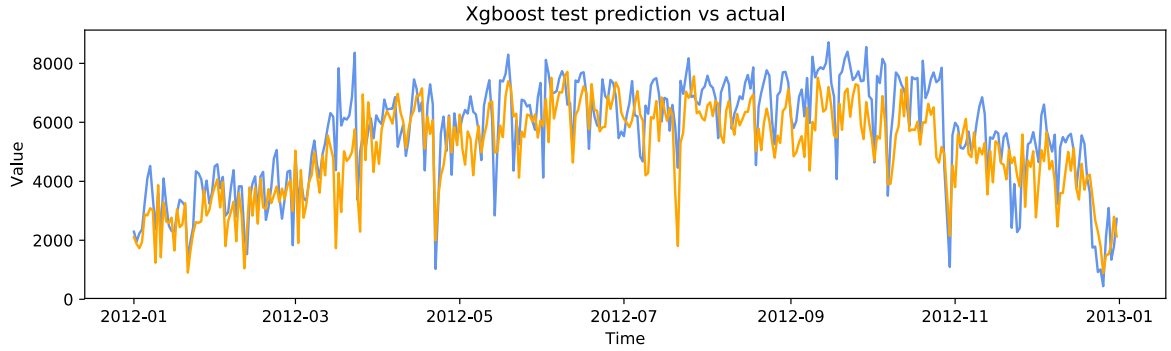


Figure 10: Predicted vs Actual Bike Demand over 2012

From the results shown in Figure 8, 9 and 10, it was observed that XGBoost predictive capability is comparable to NN and is also better at predicting sudden drop in demand when compared to RF. However, it was also noted that XGBoost tends to further underestimate (even when compared to NN) peak values, which may have resulted in its larger RMSE. Notwithstanding that, XGBoost was able to achieve the best performance from a business standpoint, this may be because the business objective is one that is more skewed towards underestimation (and penalises overestimation more severely).

3.4 Ensemble

While all three models described above managed to outperform the benchmark, it was postulated that an ensemble using some or all of the three models would further improve performance. Hence, two major model ensemble methods – namely, blending and stacking – were used to investigate if predictive capability of the models can be further improved.

3.4.1 Ensemble (Blending)

One of the most commonly used ensemble approaches is uniform blending. It works by assigning a uniform weight to all the models used in the ensemble, which adjust their individual outputs. The adjusted outputs are then used to derive the final prediction. Linear blending can basically be seen as a generalised form of Uniform blending whereby the weight for each model is different. This is done to enable tuning and the application domain knowledge to be incorporated in order to optimise the ensemble.

In our case, we adopt linear blending approach. After several iterations, it was concluded that XGBoost and NN would result in the best linear blending model through training. After parameter tuning, the derived optimal weights to minimize RMSE and maximize profit is,

$$Linear\ blending\ ensemble = 0.62 * Xgboost + 0.38 * Neural\_Network$$

The associated profit and profit-to-cost ratio for the ensemble is shown in Table 9 below and the predicted vs actual bike demand plot is shown in Figure 11.

Table 9: Summary of results – Ensemble (Blending)

Model	Profit $ million	Cost $ million	Profit/Cost	RMSE Percentage Increase	RMSE Absolute Demand
benchmark	145.9	408.5	35.7%	-	-
blending	168.7	390.6	43.2%	0.33	949

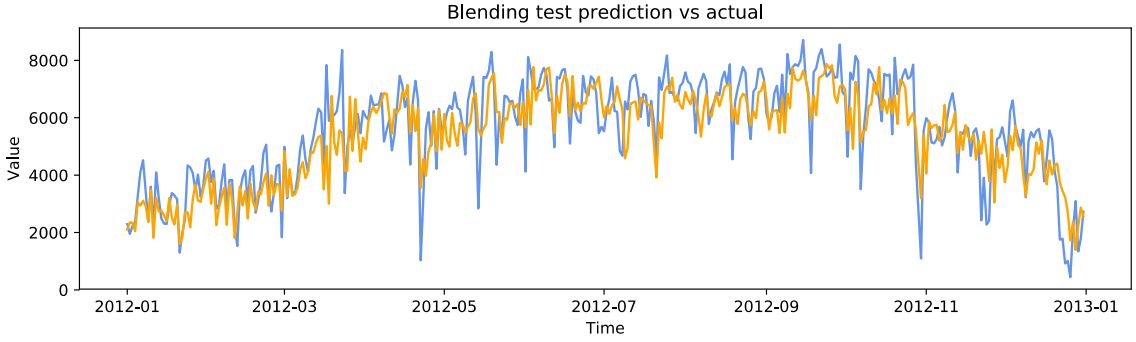


Figure 11: Predicted vs Actual Bike Demand over 2012

3.4.2 Ensemble (Stacking)

Compared to linear blending, stacking is a more sophisticated way that uses a second level algorithm to optimise the combination of the models used in the ensemble. Figure 12 shows a schematic view of the stacking approach while the steps used in constructing the stacking ensemble for our case are listed after that.

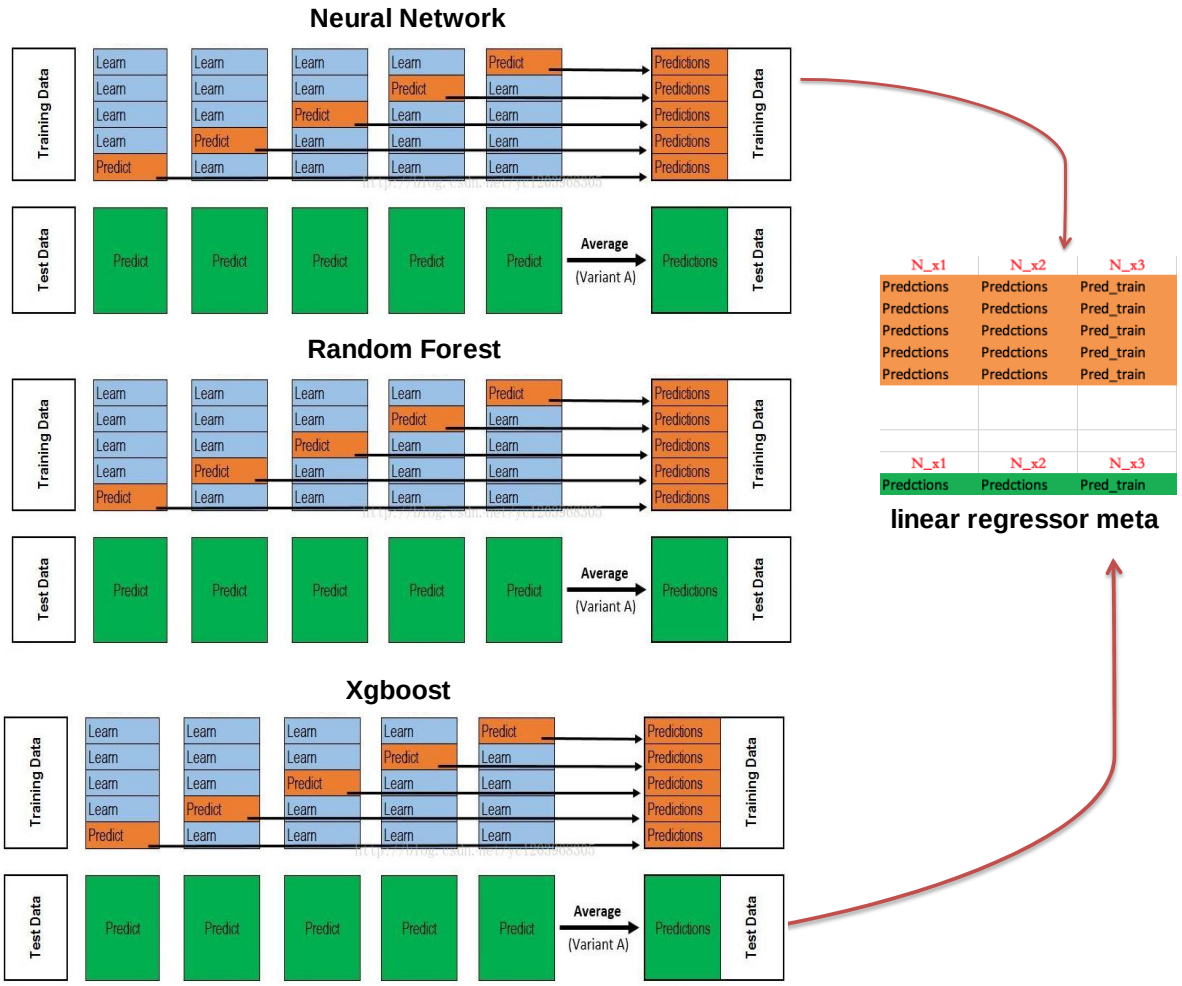


Figure 12: Schematic of Stacking Ensemble approach

Ensemble (Stacking) Steps

Input: Train data, test data, 3 models (RF, NN and XGBoost)

Step1: Slice the training data into 5 folds (using K-Fold)

Step2: For each model, i^{th} fold is retained to be the validation, the rest 4 folds are used to train the i^{th} base-level regressor. Use i^{th} base-level regressor to predict the i^{th} validation and test, where 5 validation prediction (from 1^{st} to 5^{th}) and 5 test predictions are generated.

Step3: For each model, 5 validation predictions are concatenate from 1st to 5th into a second-round input dataset, and 5 test predictions based on i^{th} models will be used to calculate the average which will be used as a second-round test dataset. Therefore, 3 new pairs of input and test dataset for each model are generated.

Step4: Construct a meta regressor (linear regressor) to fit the input dataset, use the meta regressor to predict the test dataset and return the final test prediction result.

Output:Test prediction

The associated profit and profit-to-cost ratio for the ensemble is shown in Table 10 below and the predicted vs actual bike demand plot is shown in Figure 13.

Table 10: Results of Ensemble, Stacking

Model	Profit $ million	Cost $ million	Profit/Cost	RMSE Percentage Increase	RMSE Absolute Demand
benchmark	145.9	408.5	35.7%	-	-
stacking	164.9	368.7	44.7%	0.36	1136

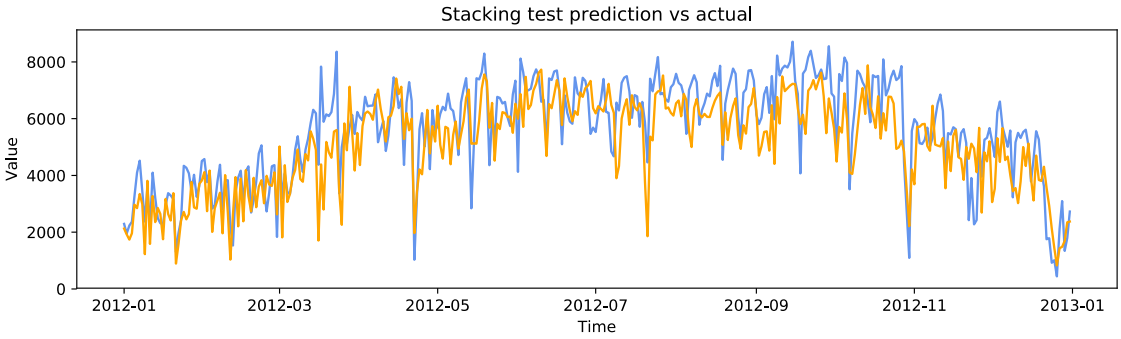


Figure 13: Predicted vs Actual Bike Demand over 2012

From the results of both ensemble models (stacking and blending), it was shown that they outperform the RF, NN and XGBoost models that was initially trained. Specifically, Ensemble (blending) was able to achieve the highest absolute profit while Ensemble (stacking) was able to achieve a comparable profit with the highest cost-to-profit ratio. This potentially suggest that the models incorporated into the ensembles had individual strength in prediction and the ensemble approach managed to combine these strengths into an overall better predictive model.

3.5 Model Comparison and Selection

The results of all the various models are summarised in Table 11 below. All 5 models were able to outperform the benchmark significantly by 11.4% to 15.6% (16.7 to 22.8 million dollars) increase in profit. While the NN model had the lowest RMSE, its prediction did not result in the best profit or profit-to-cost ratio. Ensemble (blending), which had slightly worse RMSE than the NN model, returns with the highest absolute profit margin. Interestingly, Ensemble (stacking), which had the worst RMSE, returned with the highest profit-to-cost ratio.

Table 11: Results of Models based on 1 Year Training Data

Model	Profit $ million	Cost $ million	Profit/Cost	RMSE Percentage Increase	RMSE Absolute Demand
benchmark	145.9	408.5	35.7%	-	-
rf_best	162.6	415.1	39.2%	0.34	981
nn_best	166.8	389.6	42.8%	0.32	980
xgb_best	167.7	391.3	42.9%	0.35	983
blending	**168.7**	**390.6**	**43.2%**	**0.33**	**949**
stacking	164.9	368.7	44.7%	0.36	1136

As the objective of this study was to maximise profit (not demand prediction accuracy), profit (instead of RMSE) was used as the criterion for final model selection. Considering profit-to-cost ratio (which directly impacts return of investment) is often more important than absolute profit in the business context, Ensemble (stacking) is assessed to be the best model for implementation by the bike sharing company.

3.6 What If Scenario

As an additional exercise to explore and potentially reduce the effect of model degradation with age, the training / test data set was adjusted such that the first 18 months was used as the training data while the remaining 6 months was used as the test data subset.

To get a quick sensing on how this newly defined training data subset would affect modelling results, the data was fitted in a random forest basic model with constant hyperparameters. From the chart showing absolute errors of demand by date (Figure 14), it was obvious that the absolute errors of the model fitted by 18 months were much lower than the model based on one-year training data. However, the performance of these two models began to converge from November 2012 onwards. This may be due to age degradation of the models and the occurrence of extreme weather events during this period.

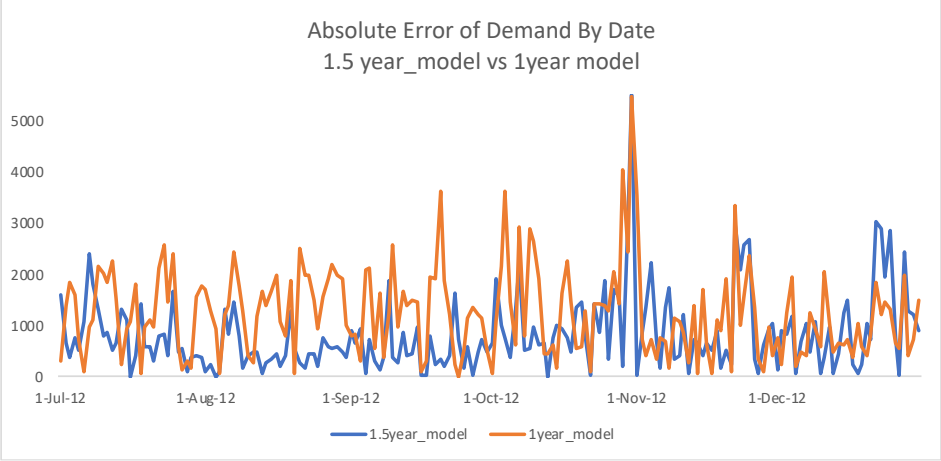


Figure 14: Absolute Error of Demand by Date

A comparison between the results of the RF model using 12 months and 18 months training data is summarised in table 12 below. It was noted that while profit increases slightly by about $31,923 (3.8%), profit-to-cost ratio dropped by 4.4%.

Table 12: Profit based on Random Forest from Jul to Dec 2012

Random Forest	Revenue / $	Cost / $	Profit (Revenue – Cost) / $	Profit/Cost Ratio
18 months Training data	3,158,155	2,282,010	876,145	38.4%
12 months Training data	2,815,078	1,970,856	844,222	42.8%
		Difference	31,923 (+3.8%)	- 4.4%

Data Balancing. As the 18 months data was deemed to be imbalanced with regards to seasonal patterns and effects, the team attempted to balance the data by replicating data from Jul to Dec 2011 to be used as replacement for Jul to Dec 2012 in the training dataset. However, from the absolute error by date chart (Figure 15), it was observed that there was no significant improvement to the model performance. This was likely because the replicated data did not capture the increasing trend from 2011 to 2012, and as a result was not useful in improving model performance.

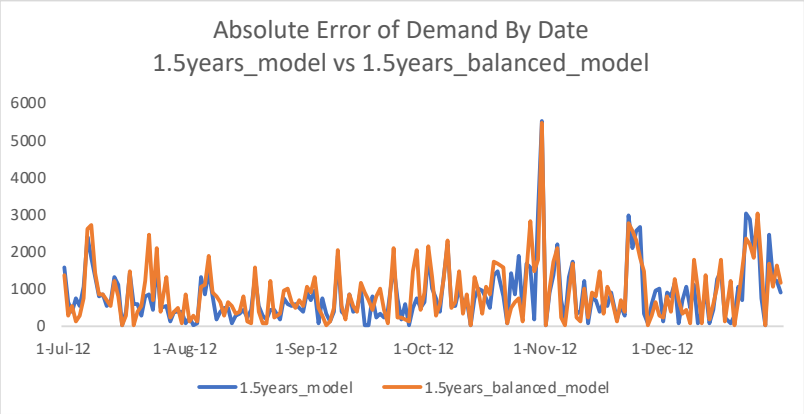


Figure 15: Absolute Error of Demand by Date (With Data Balancing)

Subsequently, different models were trained using 18 months training data. From a RMSE perspective, it was concluded that using 18 months as training data did not result in model improvement. While the results from table 12 suggest slight improvement in revenue, this came at the cost of a reduced profit-to-cost ratio. In fact, the cost-to-profit[2] ratio for most of the models were slightly worse off compared to earlier models trained. However, it was observed that Ensemble (stacking) outperform the rest in both absolute profit ($91.1m) and profit-to-cost ratio (43.7%) this time round. The result of this exercise is summarized in Table 13.

Table 13: Results of Models based on 1.5 Year Training Data (without Data Balancing)

Model	Profit $ million	Cost $ million	Profit/Cost	RMSE Percentage Increase	RMSE Absolute Demand
benchmark	81.9	408.5	20.0%	-	-
rf_best	89	222.4	40.0%	0.38	973
nn_best	90.8	219.1	41.4%	0.44	1026
xgb_best	89	222.5	40.0%	0.4	999
blending	91.1	219.9	41.4%	0.38	911
stacking	**91.1**	**208.3**	**43.7%**	**0.37**	**1022**

[2] It should be noted that absolute profit was not used for comparison as the period of assessment for both set of models were different. As such, comparisons were made using profit to cost ratio over the entire period. By looking at the results, it is also clear that revenue for the 2nd half of the year is higher than the 1st half in this dataset.

4. Business Performance

Performance Overview. Based on the rationale stated above, the best performing model was assessed to be Ensemble (Stacking) as it has the best Profit-to-Cost ratio. Compared with the benchmark (Table 14), it was able to achieve an improvement of 9% to Profit-to-Cost ratio and a corresponding 13% increase ($19m) in profit. The results for Ensemble (Blending) was also included for reference as it was able to achieve the highest absolute profit, which is about $22.8m (15.6%) more than the benchmark, with the corresponding Profit-to-Cost ratio improvement of about 7.5%.

Table 14: Comparison of Benchmark with Proposed and Alternative Model

Model	Profit ($ million)	Cost ($ million)	Profit/Cost
Benchmark	145.9	408.5	35.7%
Proposed Model (Ensemble – Stacking)	164.9	368.7	44.7%
Alternative Model (Ensemble – Blending)	168.7	390.6	43.2%

Proposed Model vs Benchmark. From visual inspection of Figure 16, it is not evident which model perform the best. It is however noted that in general (e.g. from a weekly or monthly basis), the proposed models always outperform the benchmark. Despite so, if we focus deep enough and compare both of them on a day to day basis, then there will always be cases where the benchmark outperforms the model. This is especially true if there is erratic oscillation of data points causes by extreme weathers. However, notwithstanding those cases, the proposed model generally outperforms the benchmark.

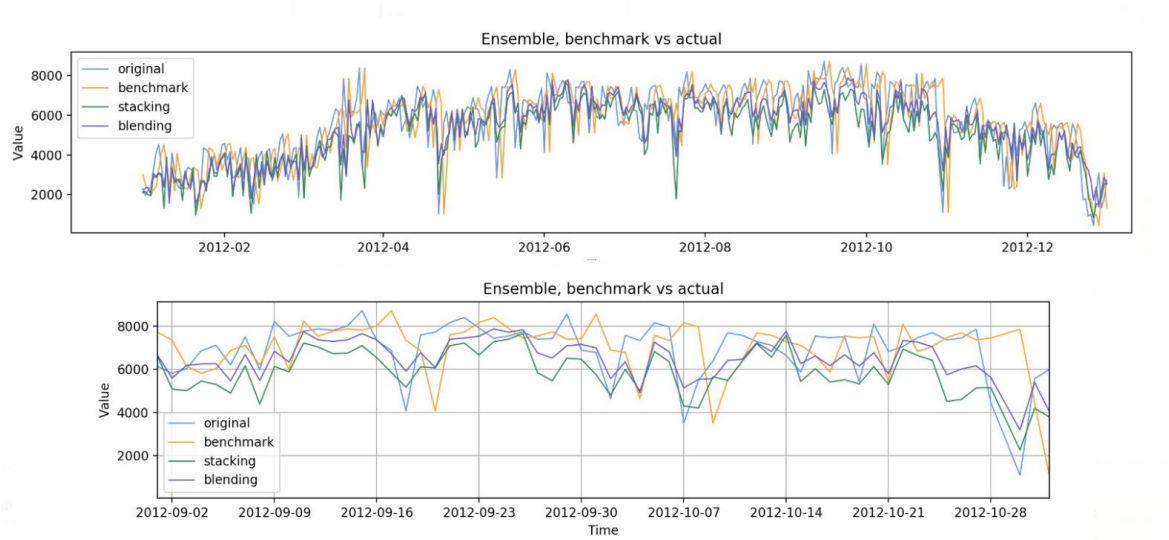


Figure 16: Actual vs Predicted demand (Top – Entire Test dataset, Bottom – Sep and Oct 12)

Model Degradation with Age. From Figure 17, it can be observed that the absolute error of demand is gradually increase with each passing month. While higher error is somewhat correlated to the occurrence of bad weather conditions, there is also a distinguishable upwards trend that potentially reflects model degradation with age.

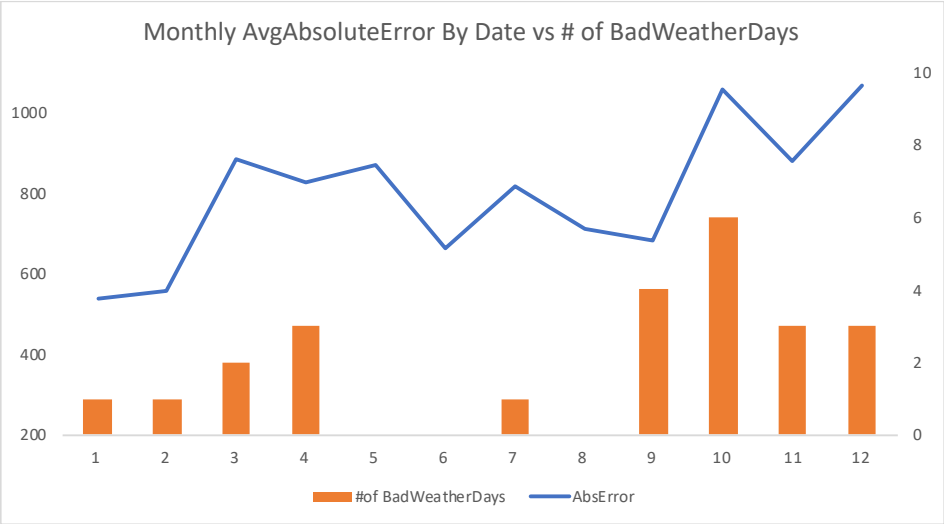


Figure 17: Monthly Absolute Error of Demand

Season Influences on Models. In addition, strong seasonal influence was also observed from Figure 17. The performance of the model was generally worse off in Spring (Mar to Jun) and Autumn/Falls (Sep-Dec). This is assessed to be due to the increased occurrences of extreme and fluctuating weather events, which the model was not adequately trained to predict (extreme shift in weather data point in the entire dataset is limited compared to other data points).

What-If Scenario. From the what-if scenario explained in detail in Section 3.6, it can be concluded that models generated using 18 months training data were actually worse off than the model built using 12 months training data. Data balancing using Jul to Dec 2011 for the 18 months training data did not yield significant improvement to the model predictive power as well. In fact, the cost-to-profit ratio for both cases (with/without data balancing) had worse profit-to-cost ratio and RMSE when compared to the model that was built using 12 months training data. Absolute profit was higher, however that is not a useful indicator as the benchmark also showed a much higher profit, indicating that the revenue/profit generated in the 2nd half of the year is higher than the 1st half of the year.

5. Conclusion and Recommendations

This study was successful in generating various predictive models using Random Forest, Neural Network, XGBoost, Ensemble (Blending) and Ensemble (Stacking) algorithms. All the models generated was capable of outperforming the benchmark performance from the client, Capital Bikeshare. It is recommended that the client adopt Ensemble (Stacking) model as it offers a significant improvement in profit with the best cost-to-profit ratio amongst all models trained. Furthermore, to ensure the model remains current, it is recommended for the model to be updated every 6 (recommended) to 12 months (maximum) to minimise age-related degradation of the predictive model. On top of that, to automate the updating process, it is recommended for future work to look into a tracking and self-learning system that uses a 12-month rolling window for real-time prediction in the future.

Enclosed:

Annex A – Data Understanding and Exploration of Bike Demand from Jan 2011 to Dec 2012

Appendix I – Data Dictionary for Original Dataset and Newly Created Features

Annex A

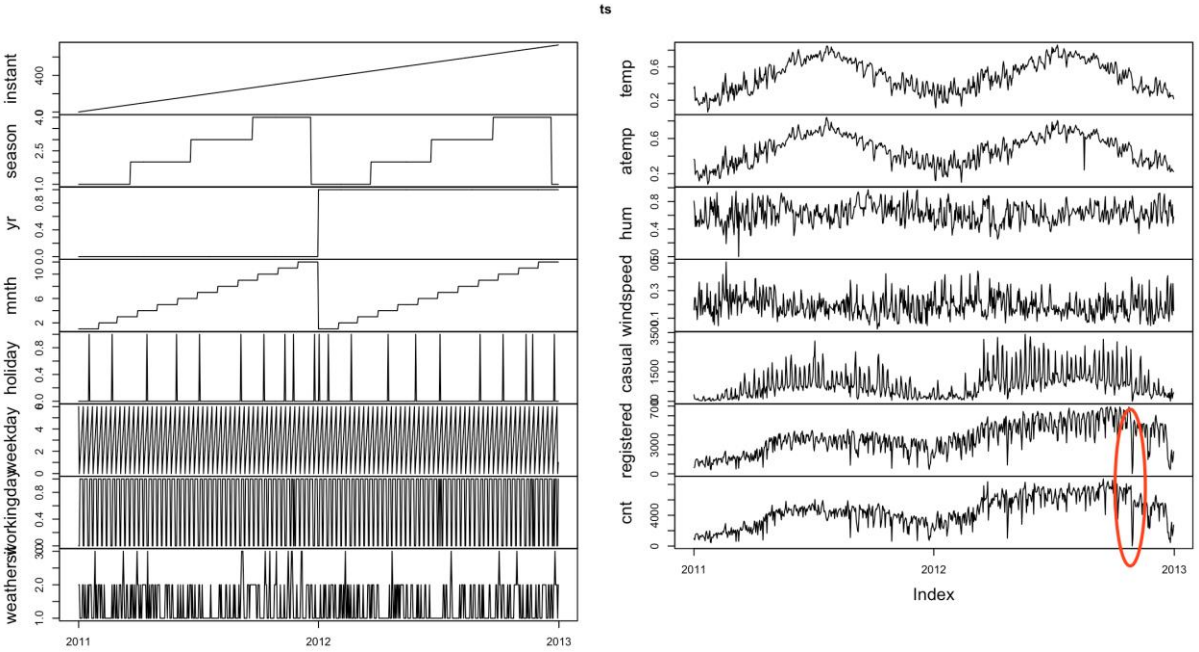
Data Understanding and Exploration of Bike Demand from Jan 2011 to Dec 2012

Data Understanding

The dataset was reviewed to better understand the data and to look for anomalies which is necessary to facilitate subsequent data cleaning, feature engineering and model optimisation of the project.

1. Total vs Registered vs Casual rentals. While the total count (cnt) of rental each day is the summation of registered and casual rental, the ratio of registered to casual rental differs greatly. This is likely because casual users accounted for only 20% of the total demand but had a much larger variance compared to registered users. It was also observed that there is an increasing trend subjected to some seasonal influences on the daily rental from 2011 to 2012,
2. Abnormal near-zero demand. There was a very significant drop for rental on 29th October 2012 (circled in red in Figure A-1). Further research suggest that this is due to extreme bad weather, in particular, hurricane *'Sandy'* was reported to hit the region on that day.

Figure A-1: Time-based Distribution of All Variables



3. It was also observed that while *Temp* and *aTemp* were affected by season, humidity and windspeed were not.

Data Exploration

1. Correlation Analysis. The relationship between variables are examined by examining the correlation matrix (Figure A-2) and scatterplot matrix (Figure A-3) for all numerical variables. It was observed that temperature (*temp*) and feel-like temperature (*atemp*) are almost perfectly correlated with each other and are both highly correlated with total counts (*cnt*). As feel-like temperature is assessed to be directly derived from temperature, only temperature is used for subsequent analysis.

Figure A-2: Correlation Analysis for numeric variables

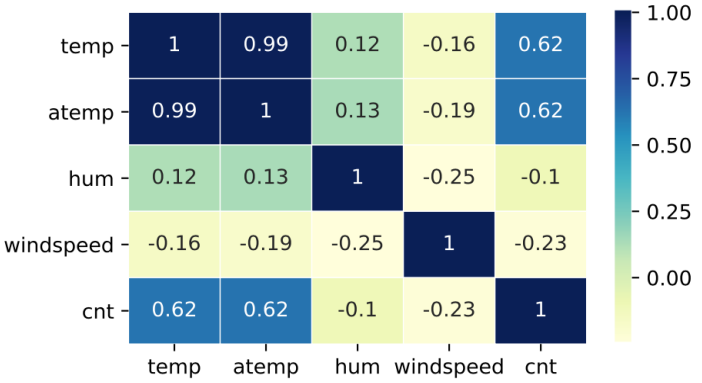
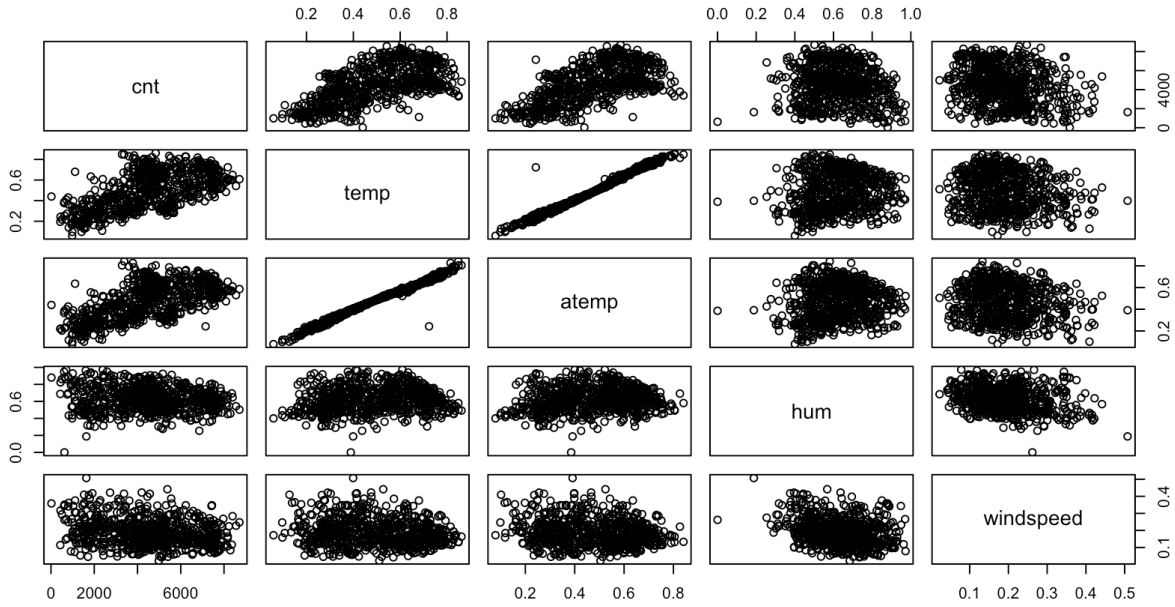


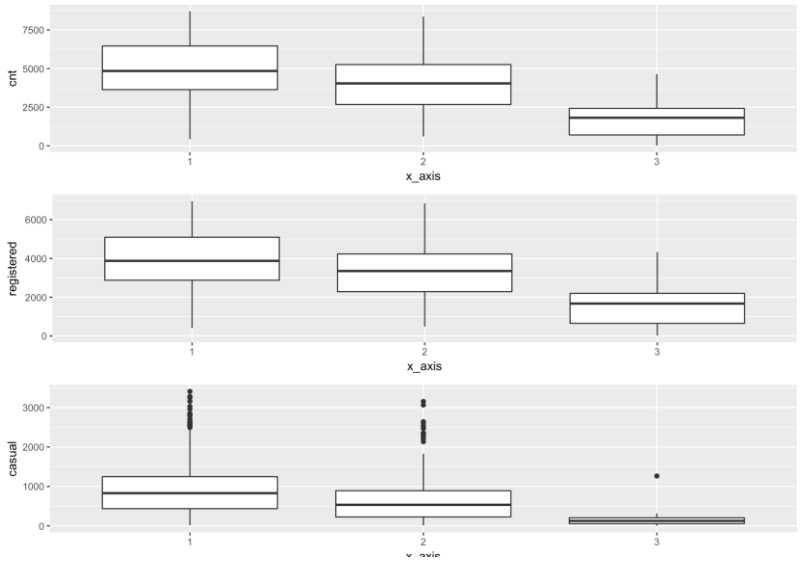
Figure A-3: Scatter Plot of Numeric Variables



2. Effects of categorical variables. Scatterplots and boxplots were used to explore the relationship between the dependent and independent (categorical) variables. It was observed that weather condition [3] (*weathersit*) directly impacts bicycle demand, particularly for days with more extreme weather (blue dots in Figure A-4), which accounts for most of the outliers in the data. This is also observed in the boxplots illustrated in Figure A-5 where the bike rental varies with weather, and where the least number of bikes were rented during Class 3 weather.

Figure A-4: Scatter Plot of *'weathersit'*

Figure A-5: Boxplot of *'weathersit'*



[3] Weather condition is broadly classified into class 1, 2 and 3, with class 3 representing the most severe weather condition.

3. Erroneous Weather Condition Classifications. During initial data exploration, it was observed there were several outlying points in the dataset that shows significantly low bike demand. Unlike other data points with similar demand, these instances show humidity and temperature values which are significantly different. It was also observed that while there were four classes for *weathersit* based on the data dictionary, no records of class 4 was observed. Subsequent online research[4] shows that the weather for these days are more extreme and should have been classified as class 4 instead of class 3. This might be due to input error, or a general reluctance to assign class 4, which includes heavy rain, snowstorm and ice pallets. The list of affected inputs is shown in Table A-1 below and has been reclassified accordingly.

Table A-1: Extreme Weather Events in Washington DC, 2011-2012

Date	Weather	OriginalWeathersit	ModifiedWeathersit
26/01/2011	winter storm, heavy snow	3	4
16/04/2011	tornado outbreaks, thunderstorm	3	4
07/09/2011	Tropical Storm *Lee*	3	4
08/09/2011	Tropical Storm *Lee*	3	4
29/10/2011	heavy rain	3	4
07/12/2011	recorded breaking rain storm	3	4
22/04/2012	light rain, fog	2	3
29/10/2012	Hurricane *Sandy*	3	4
30/10/2012	Hurricane *Sandy*	3	4
26/12/2012	winter storm, heavy snow	3	4

4. Effects of working and non-working day on bike rental. It was observed that there is no significant difference in overall bike rental on working days and non-working days. Despite so, it was noted that that registered users were more likely to rent bikes on working days and casual users, on the contrary, rents more bike on non-working days like weekends and holidays and particularly during summer and autumn. This is because users who use the bike regularly (e.g. daily commute to work) is more likely to register for the service while non-registered users (i.e. casual) are expected to use these bikes on a casual basis and more likely to do so during non-working days.

[4] Previous data weather in Washington D.C. extracted [here] and [here] shows that there was days with extreme weather condition during the 2-year period.

Figure A-6: Line Graph of *'workingday'*

5. Highly correlated variables should be used with caution depending on the modelling techniques used
6. Erroneous weather classification (i.e. Class 3 *weathersit* which should be classified as Class 4) should be rectified accordingly
7. Demand for registered and casual users to be predicted separately as they behave differently. Total demand of rental bikes can be derived by combining the output of the two predictions.

Appendix I

Data Dictionary for Original Dataset and Newly Created Features

Table I-1: Original Variables from Dataset

S/N	Variable	Type	Description	Value
1	*instant*	-	Index	-
2	*dteday*	DateTime	Date	-
3	*season*	Categorical	Season	1:spring, 2:summer, 3:autumn, 4:winter
4	*yr*	Categorical	Year	0: 2011, 1:2012
5	*mnth*	Categorical	Month	1 to 12
6	*holiday*	Categorical	Hour	0: not holiday, 1: holiday
7	*weekday*	Categorical	Weather	0 to 6
8	*workingday*	Categorical	If day is neither weekend nor holiday is 1, otherwise is 0	0: not working day, 1: working day
9	*weathersit*	Categorical	Weather Condition	1: Clear, Few cloud, 2:Mist, cloudy, 3:Light Snow, rain, 4: Thunderstorm
10	*temp*	Numeric	Normalized temperature in Celsius.	-
11	*atemp*	Numeric	Normalized feeling temperature in Celsius	-
12	*hum*	Numeric	Normalized humidity	-
13	*windspeed*	Numeric	Normalized wind speed	-
14	*casual*	Numeric	Count of casual users	-
15	*registered*	Numeric	Count of registered users	-
16	*cnt*	Numeric	Count of total rental bikes including both casual and registered	-

Table I-2: Created Features

S/N	Variable	Type	Definition
1	*cnt_inc_ratio*	Target	% Increase of D_t (demand for time t) against D_{t-2}
2	*cnt_inc_ratio_lag2*	Numeric	% Increase of D_{t-2} against D_{t-4}
3	*cnt_inc_ratio_weekly_avg*	Numeric	% Increase of D_{t-2} against the average of D_{t-2} to D_{t-7}
4	*cnt_inc_ratio_monthly_avg*	Numeric	% Increase of D_{t-2} against the average of D_{t-2} to D_{t-30}
5	*cnt_inc_ratio_weekly_max*	Numeric	% increase of D_{t-2} against the maximum from D_{t-3} to D_{t-8}
6	*cnt_inc_ratio_weekly_min*	Numeric	% Increase of D_{t-2} against the minimum from D_{t-3} to D_{t-8}
7	*temp_inc*	Numeric	% Increase of D_{t-2} against D_{t-4}
8	*cnt_avg_aheadWeek*	Numeric	Demand for preceding week's average for D_{t-2} to D_{t-7}
9	*cnt_avg_ahead3days*	Numeric	Preceding 3 days average demand from D_{t-2} to D_{t-4}
10	*cnt_avg_aheadMonth*	Numeric	Preceding month average from D_{t-2} to D_{t-31}
11	*cnt_median_LastWeek*	Numeric	Median demand of preceding week from D_{t-2} to D_{t-7}
12	*cnt_lastWeekday*	Numeric	Demand of the same day last week, D_{t-7}