Machine Learning Paradigms

Jonathan Erskine

nl21501

January 2022

Abstract

This report demonstrates three approaches to a highly non-linear prediction task of bicycle availability at numerous stations across Valencia. Gaussian process regression, tree regression and random forest regression are implemented and compared. Input data is modified to observe the effects, and it is found that random forest regression generally performs better than other models across all data sets.

Predefined linear models are assessed as an alternative prediction method and finally an attempt is made to combine a linear regression and machine learning approach to produce better results.

The linear models outperform the non-linear predictive models when filtered on a training set. The best performance is achieved by averaging the predictions of the top contenders from each of the linear and non-linear models.

1 Introduction

This report describes a solution to the **University of Bristol - Machine Learning Paradigms Kaggle competition**, which requires prediction of bicycle availability at rental stations across Valencia. The competition consists of three phases:

- **Phase 1**: Given a month of historical data from 75 [201:275] stations, predict 3 months of bicycle availability for each station, by:
  (a) Generating a learning model for each individual station, trained on it's own data
  (b) Generating a combined learning model which trains on all station data.
- **Phase 2**: Predict bicycle availability by applying linear models trained on 200 other stations [1:200], where multiple models for each station have been provided. Determining the best blend of these models to produce optimum results is determined experimentally.
- **Phase 3**: Attempt improved prediction performance by combining both the learning model from **Phase 1** and the linear models from **Phase 2**.

2 Methodology

2.1 Data Exploration

The data set provided contains information, or 'features' regarding the individual **station** and the **time** and **weather**, and task-related features relating to the time-history of number of bikes at each station. All features are listed in Table 2.1.

Category	Features	Description
Station	Station ID	Non-correlated station ID number
	Latitude	-
	Longitude	-
	Number of Docks	Max amount of bikes per station (varies between stations)
Time	Timestamp	Seconds
	Year	-
	Month	-
	Day	-
	Hour	-
	Weekday	Mon - Sun
	Week-hour	1:168
	Holiday	1 or 0
Weather	Max Wind Speed	ms^{-1}
	Mean Wind Speed	ms^{-1}
	Wind Direction	$\circ$
	Temperature	C°
	Relative Humidity	%
	Air Pressure	$mBar$
	Precipitation	L/m^2
Task-Specific	Bikes 3h Ago	Number of bikes at a given station 3 hrs prior to the current timestamp
Profile	Full profile	Average bike numbers for the current week-hour during all previous weeks
	Full profile 3 hours ago	As above for 3 hrs prior
	Short Profile	Average bike numbers for the current week-hour over the last four weeks
	Short Profile three hours ago	As above for 3 hrs prior
Target	Bikes	Number of bikes at the station i.e. the variable we are trying to predict

Table 1: Features and associated descriptions for competition data set

Understanding the impact of each feature on variance of bikes can be achieved statistically, but first we will simply observe them, to see if any obvious inferences can be made. We will look at the variation of number of bikes with respect to each feature, and also perform a single-component partial least squares (PLS) regression analysis. This yields a metric to help us better understand the correlation between each feature and the number of bikes.

Figure 1 shows a plot of each feature (x) against number of bikes (y). A green line shows the mean number of bikes, and a red line indicates our ”predictions” from the PLS analysis, which in this context is our line of best fit. The coefficient of determination (R^2), which represents the proportion of variation in the dependent variable that is predictable from the independent variable, is calculated for each variable and is presented in Figure 2.

Looking at Figure 1, it is clear that the **month**, **year** and **precipitation** will have zero effect on the results as they do not vary, and so they are omitted (this would not be the case if the models in Phase 2 considered the month and year). Here are some other statements which can be inferred from our observations:

1. A small number of stations have a much higher capacity of docks than the rest
2. On holidays, and at weekends, these high capacity stations are never full
3. There is a concentration of bicycles in the South-East of Valencia
4. Both lower temperature and higher humidity correlate with higher numbers of bikes
5. The number of bikes 3 hours ago is the feature with the most influence
6. The each pair of full and short profiles are identical

We can already begin to guess at some deeper meaning. For example, the high capacity stations could be in the city centre and therefore are not full on the weekends, as people take them to their home station. The weather observations (low temperature and high humidity) could be indicating that the stations are more full at night when people have returned their daily rentals.

For a final look at the data, we can plot latitude and longitude on a map of Valencia, with markers representing the station capacity (gray) and the average number of bikes per station (red). This view provides a perspective of station activity, as we can see many stations that are rarely, if ever full. It also confirms our suspicion of a higher number of bikes in the South East; perhaps this is more of a leisure destination and doesn’t befall quite as much commuter traffic.

So, how does this affect the modelling strategy? What the data shows is that the prediction function is going to be complex, and probably not very smooth. While the profile variables give a relatively good line of best fit, there are other factors such as location, capacity and day of the week which create more of a classification problem due to working patterns and other human behaviour. It is also hard to know what is a driving feature and what features are consequences of others. For example, there is a pattern which shows high bicycle numbers for a small pressure range; could this be because people detest cycling within that specific pressure range? It is more likely that this region of pressure corresponds with periods of low traffic during the day or night; we can still train on this feature and use it to our advantage, but the model might collapse if we looked at a different season.

2.2 Feature Engineering

For the purposes of investigation, we define distinct data sets which will be used in our modelling to show the effects of including and excluding certain variable. Our control group will contain all of our data, with the exception of **"Weekday"**, which is converted to 7 one-hot encodings for days of the week. We then create two reduced versions of the control, removing our redundant variables (those which are observed not to vary or those which are duplicated).

For the case of training on individual stations, this includes station ID, latitude and longitude and number of docks, but these are kept in for training our collective model. A second sub-group consists only of the profile variables (and latitude, longitude and number of docks for the cumulative model).

Finally we experiment with a blend of profile data and some meta-data, creating one-hot encodings to indicate whether it is dark or the weekend (replacing our weekday encodings). Our sub-groups are defined in Table 2.2 and an 80:20 train-validation split is chosen for dividing data. Data is split at random.

Group	Model	Features
A (control)	Individual/ Combined	'station', 'latitude', 'longitude', 'numDocks', 'timestamp','year', 'month', 'day', 'hour', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday', 'weekhour', 'isHoliday', 'windMaxSpeed.m.s', 'windMeanSpeed.m.s', 'windDirection.grades', 'temperature.C', 'relHumidity.HR', 'airPressure.mb', 'precipitation.l.m2', 'bikes 3h ago', 'full profile 3h diff bikes','full profile bikes', 'short profile 3h diff bikes', 'short profile bikes'
B_i	Individual	'timestamp', 'day', 'hour', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday', 'weekhour', 'isHoliday', 'windMaxSpeed.m.s', 'windMeanSpeed.m.s', 'windDirection.grades', 'temperature.C', 'relHumidity.HR', 'airPressure.mb', 'bikes 3h ago', 'full profile 3h diff bikes', 'full profile bikes'
B_c	Combined	As above, plus 'station', 'latitude', 'longitude', 'numDocks'
C_i	Individual	'bikes 3h ago', 'full profile 3h diff bikes', 'full profile bikes'
C_c	Combined	As above, plus 'latitude', 'longitude', 'numDocks'
D_i	Individual	'weekhour', 'isDark', 'isWeekend', 'isHoliday', 'bikes 3h ago', 'full profile 3h diff bikes', 'full profile bikes'
D_c	Combined	As above, plus 'latitude', 'longitude', 'numDocks'

Table 2: Features and associated descriptions for competition data set

2.3 Prediction Algorithms

We approach this problem with three different methodologies to discern whether any approach stands out, namely:

- Bayesian Regression
- Tree Regression
- Random Forest Regression

The aim of this approach is to understand how these algorithms can handle this type of data, and so any conclusions drawn are constrained to similar contexts.

2.3.1 Bayesian Regression

Our chosen approach to the regression problem is Gaussian process regression, as we can experiment with different covariance kernels and draw inferences about the results with respect to the smoothness of the predicted function. We will apply the Matern class of covariance kernels, with the covariance matrix, k, defined as:

$$k_{(x_i,x_j)} = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}}{l} d(x_i, x_j) \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}}{l} d(x_i, x_j) \right) \tag{1}$$

where $d(\cdot, \cdot)$ is euclidian distance, $K_\nu(\cdot)$ is a modified Bessel function, and $\Gamma(\cdot)$ is the gamma function. We modify the parameter ν to reflect our expectation of the smoothness of the data; as ν increases, so does our expectation that the data can be presented by a smooth function. Variation of the Matérn covariance kernels and associated sample functions is presented in Figure 4. For more detailed derivations and information on the Matérn class of covariance kernels, see [1].

2.3.2 Tree Regression

We test a tree-based regression method due to the presence of categorical data in our input features. For each feature, the regression tree will iterate splitting the data into clusters, and measure the squared residual error from assigning one output variable to each cluster. By finding the minimum squared residual error per split, and then recursively performing this step, a tree can be developed which segments the data (hopefully by our categorical features) then learns a function for each cluster. A good description of regression trees and their motivations is given in [2]. We will modify the minimum samples per leaf and maximum depth of the regression tree to understand whether or not the model is over-fitting on the validation set.

2.3.3 Tree Ensemble Methods

One problem with classic tree regression is accuracy; specifically, tree regression commonly fits well to training data, but can struggle to apply classifications to new samples i.e. they are inflexible. One solution to this is to use a random forest ensemble method. This involves bootstrap aggregating, or bagging, whereby we randomly select subsets of variables (bootstrapping) in order to create a large variety of decision trees (hence, random forest!).

Although each of these trees may produce poor predictions individually, by aggregating the results of each tree, it becomes possible to make a decision on the final output based on the most popular output term. Again, an in-depth description of this method is provide in [2]. Like our simple tree method, we can modify minimum samples per leaf and maximum depth to investigate the effects of tree pruning.

2.4 Test Plan

Table 3 presents a summary of the methods under test and associated parameter variation. All models used are from the SciKit Learn family and are tuned in line with the steps outlined in the documentation [3]. The parameter values are referred to as **starting** values as we expect to stray from this set in the fine tuning of these models; these ranges indicate our initial search space.

Model	Parameters	Parameter (Starting) Values
Gaussian Process Regression	ν	0.5; 1.5; 2.5
Tree Regression	Min. Samples per Leaf	1; 3; 5
	Maximum Depth	Max; $\text{Max}_D/1.1$; $\text{Max}_D/1.3$; $\text{Max}_D/2$
Random Forest Regression	Min. Samples Per Leaf	1; 3; 5
	Maximum Depth	Max; $\text{Max}_D/1.1$; $\text{Max}_D/1.3$; $\text{Max}_D/2$

Table 3: Parameters for tuning during model learning and evaluation

2.5 Linear Models

Following from the generation of results from the first phase of the assignment, we will generate results for each of the six linear models provided for Phase 2. Each type of model is trained on a subset of the features provided in the training data, and assigns weights each of these features. These models are described in Table 4. The models are trained on different stations, and there is no additional context provided, which removes the possibility of matching similar stations. Therefore, the weights and intercepts of each type of model are averaged over the 200 station models to form a generalised model.

To achieve better performance, each type of linear model is filtered by running through the test set and removing candidates which perform below a threshold. The threshold is manually set to $MAE = 2.9$ as any less completely empties some of the model groups.

2.6 Model Combination

The strongest candidate models (one non-linear and one linear) are selected from Phase 1 and Phase 2. These models are combined to try and improve the individual model results.

Model Name	Features
short	'bikes 3h ago', 'short profile 3h diff bikes', 'short profile bikes'
short temp	'bikes 3h ago', 'short profile 3h diff bikes', 'short profile bikes', 'temperature.C'
full	'bikes 3h ago', 'full profile 3h diff bikes', 'full profile bikes'
full temp	'bikes 3h ago', 'full profile 3h diff bikes', 'full profile bikes', 'temperature.C'
short full	'bikes 3h ago', 'short profile 3h diff bikes', 'short profile bikes', 'full profile 3h diff bikes', 'full profile bikes'
short full temp	'bikes 3h ago', 'short profile 3h diff bikes', 'short profile bikes', 'full profile 3h diff bikes', 'full profile bikes', 'temperature.C'

Table 4: Phase 2 linear models and associated features

Two methods are tested; (1) taking the average of the two models, and (2) using the output of the linear model as an input to the non-linear model. It is thought that including a general model in the training data for the non-linear model may help to reduce over-fitting.

3 Results

3.1 Gaussian Process Regression

The GPR performs best on the significantly reduced Group C dataset, with a mean absolute error (MAE) test score of $MAE = 2.63$. For Group A and B, the GPR model suffers from high levels of over-fitting. Over-fitting is mitigated by adding white noise to the covariance kernel. No results are available for a model trained on all stations. The best results are achieved with a covariance parameter $\nu = 0.5$, which indicates non-linear data.

A full set of training, validation and test results for the Gaussian Process regression model are shown in Table 5.

3.2 Regression Trees

The regression tree method achieves the lowest high score of $MAE = 2.63$ on Group C. For Group A and B, scores are comparable, if not slightly better than, the GPR method.

A full set of training, validation and test results for the regression tree model are shown in Table 6.

3.3 Random Forest

The random forest method produces the best performance, with the model trained on all stattions achieving high (or, low) scores of $MAE = 2.41$ on both Group B and Group D. Random forest outperforms GPR on Group C (although the difference is marginal) and achieves significantly lower MAE than either tree regression or GPR on Group A. For both high scores, pruning by setting maximum features **and** minimum-samples-per-leaf was required.

A full set of training, validation and test results for the random forest model are shown in Table 6.

3.4 Linear Models

Of the six linear models, the "full" linear model" achieves best performance, with a score of $MAE = 2.838$. Three significant figures are used to discriminate this performance with the other linear models; they all performed similarly. Filtering achieved improved performance across the group.

The test data for both the filtered and unfiltered linear models showed surprising results, with the test data performing better than the training data. This could be due to some mistake in our calculations of training error, but every effort was made to ensure correct reporting. The linear models which do not consider temperature outperform the ones which do. It's possible that, given the dataset contains a full month with no rain, that the predictions were atypical and therefore the test set is more comparable to the previous years data. The filtered linear models produce very good results on the test set.

A full set of results for the linear models is shown in Table 8.

3.5 Mixed models

The first attempt, which involved averaging the high-scoring random forest model trained on the Group D dataset and the full linear model, produced an error of $MAE = 2.24$. This was a minor improvement on the filtered linear model but it was enough to push the performance to the top of the leaderboard.

4 Discussion

4.1 Model Behaviour

4.1.1 GPR

The GPR model training required some fine-tuning. During training it was found that the GPR was very sensitive to over-fitting on per-station models. This was mitigated by adding white noise to the covariance kernel. The result was a much more reasonable validation score, with a training score which showed reasonable fitting as opposed to the aggressive over-fitting observed originally. As suspected, the best performance on the validation set is found when the Matérn modification parameter $\nu = 0.5$, which implies that our data is highly non-linear.

The GPR method caused several optimisation warnings, and there was a delicate balance between the capability of the model and hardware limitations. The attempt to learn on the control dataset for the combined station data failed in every instance due to hardware limitations. By significantly reducing the dataset (Group C) it was hoped that a model trained on all station data could be successfully generated, but memory remained a limitation.

4.1.2 Tree Regression

This model was subject to over-fitting and required immediate parameter tuning. However, it is extremely computationally light in comparison to both other methods and implementation was much more straight forward than for the GPR method.

4.1.3 Random Forest

This was the most impressive model with respect to ease of implementation, speed of learning and ability to produce a relatively good set of results quickly.

4.2 Model Results

4.2.1 GPR

The results for group A were quite poor, and could only be trained on individual training sets. This was likely a significant contributing factor to the over-fitting of the model, as GPR's are typically very sensitive, particularly when applying low-matern covariance kernel parameters. Group B saw a significant improvement in mean absolute error for individually trained models, but this was still far behind both tree regression methods. Again, the model failed when attempting to learn on all stations. Group C, which used only three parameters for the dataset (the profile variables), saw significant improvement in GPR predicition with a mean average error of 2.6 achieved on the test set.

4.2.2 Tree Regression

The unconstrained initial attempts were observed to over-fit on the training data for Group A and Group B. This was alleviated by adapting the minimum members in a leaf and the maximum number of features. Both parameters were suitable for reducing over-fitting.

Training on group B, it was observed that the change in MAE metrics was insignificant and so the complete results set is not populated. Setting the maximum number of features to 1 resulted in a very small amount of fitting and similar results to the linear models, which was to be expected.

For group C, of the two parameters which were modified (maximum features, minimum leaf members), the results tended to show more improvement when pruning with minimum samples per leaf, potentially due to the reduction in the number of features in the dataset. Even with fewer features, over-fitting can be observed if minimum samples per leaf is not increased above 1.

Group D saw the best score for regression trees trained on all models. This may be due to the additional categories which were added providing good distinctions for the tree to split on.

4.2.3 Random Forest

While similarly affected by the over-fitting issues for regression trees, this model approach provided the strongest estimate prior to parameter tuning; scoring the highest in Group A and Group B, with the model trained on all stations providing a marginal boost over the individually trained stations. Reducing the input space in Group C reduced the performance of the model, but it was still the strongest contender with an MAE of 2.61, although the GPR predicted close to this value.

Testing on Group D illustrates that only a small subset of the actual data is required to predict bicycle numbers, as the random forest approach achieved almost identical error to the random forest trained on Group B data. This also highlights the statistical bagging approach as an effective method to eke out the most significant predictors for the output prediction, compared to both the tree regression and Gaussian methods, which both struggled with over-fitting. The test score is higher than the validation score for our final attempt; this is assumed to be because our validation set was quite large (20%: 9000 points) and it is likely that the test set was smaller, which would explain fluctuations around the mean error.

4.2.4 Linear Models

The linear models performed better than the initial learning models, due to their approximation methods avoiding the large over-fitting we observed on the learning models.

4.2.5 Combined Model

The combined model is able to beat other algorithms. Due to the extended time history which the linear models are trained on, it is thought that the averaged model pulls the random forest predictions towards the mean, and therefore makes it less sensitive. This is of particular importance on the test set where we must predict for unknown contexts.

5 Conclusion

We have presented three potential methods for utilising machine learning techniques to predict on a non-linear problem. Of these models, random forest seems to achieve the best results with the least amount of effort. We recommend random forest as a lightweight solution. By tuning parameters, we show that all models are capable of achieving improved performance, and therefore these conclusions only apply to the tuning approach demonstrated in this paper.

We have also demonstrated that different selection of input data leads to significant difference in performance, and that GPR models tend to perform better on reduced features. Any conclusions relating to Gaussian regression are subject to the caveat that we were unable to run full scale model training.

The linear models, which have the benefit of stability due to a year of data, produce better results than the ;earned models. It would be interesting to compare methods with full datasets for both methods. The combined model proves to improve performance slightly. We assume this is because implementing that machine learned component allows for better prediction of extremes, but that the model is stabilised by the influence of the linear model.

The best result is achieved with an average of both linear and non-linear top contenders, with a mean absolute error of 2.24.

References

[1] CE. Rasmussen and CKI. Williams. *Gaussian Processes for Machine Learning.* Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, January 2006.

[2] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R.* Springer, 2013.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

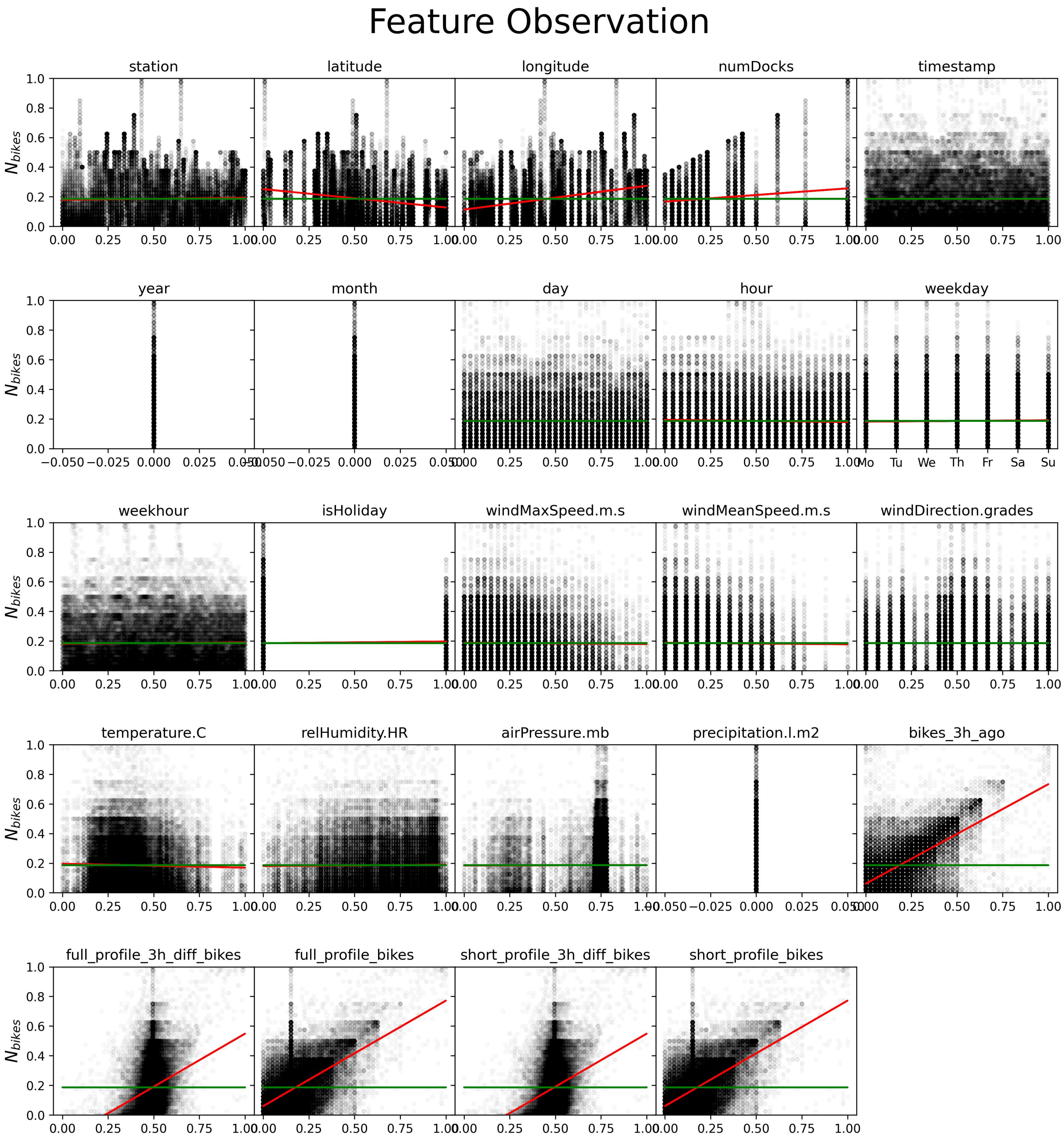

Figure 1: Features of the competition data set (X) plotted against bicycle capacity (Y)

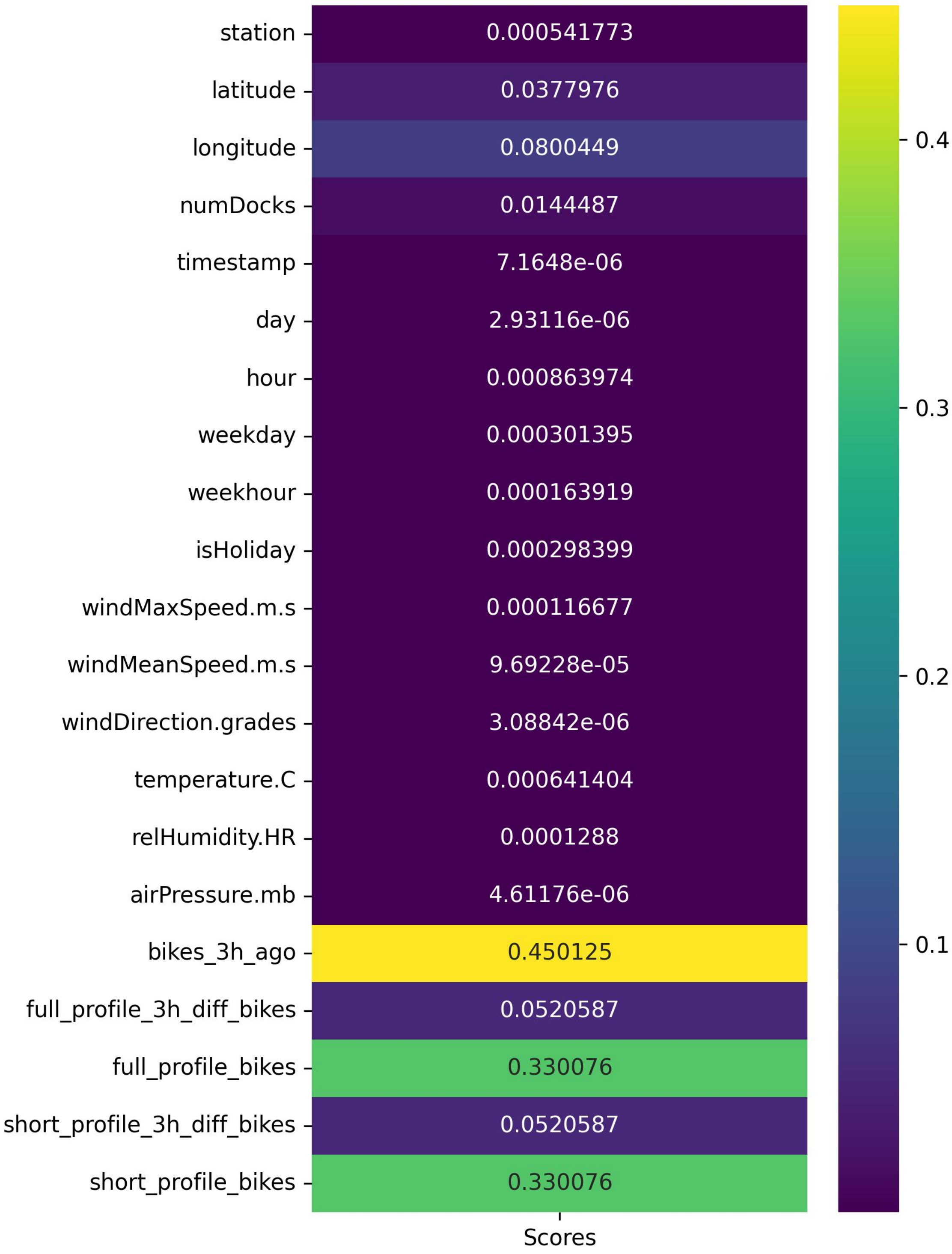

Figure 2: R^2 score for features of the competition data set with respect to bicycle capacity

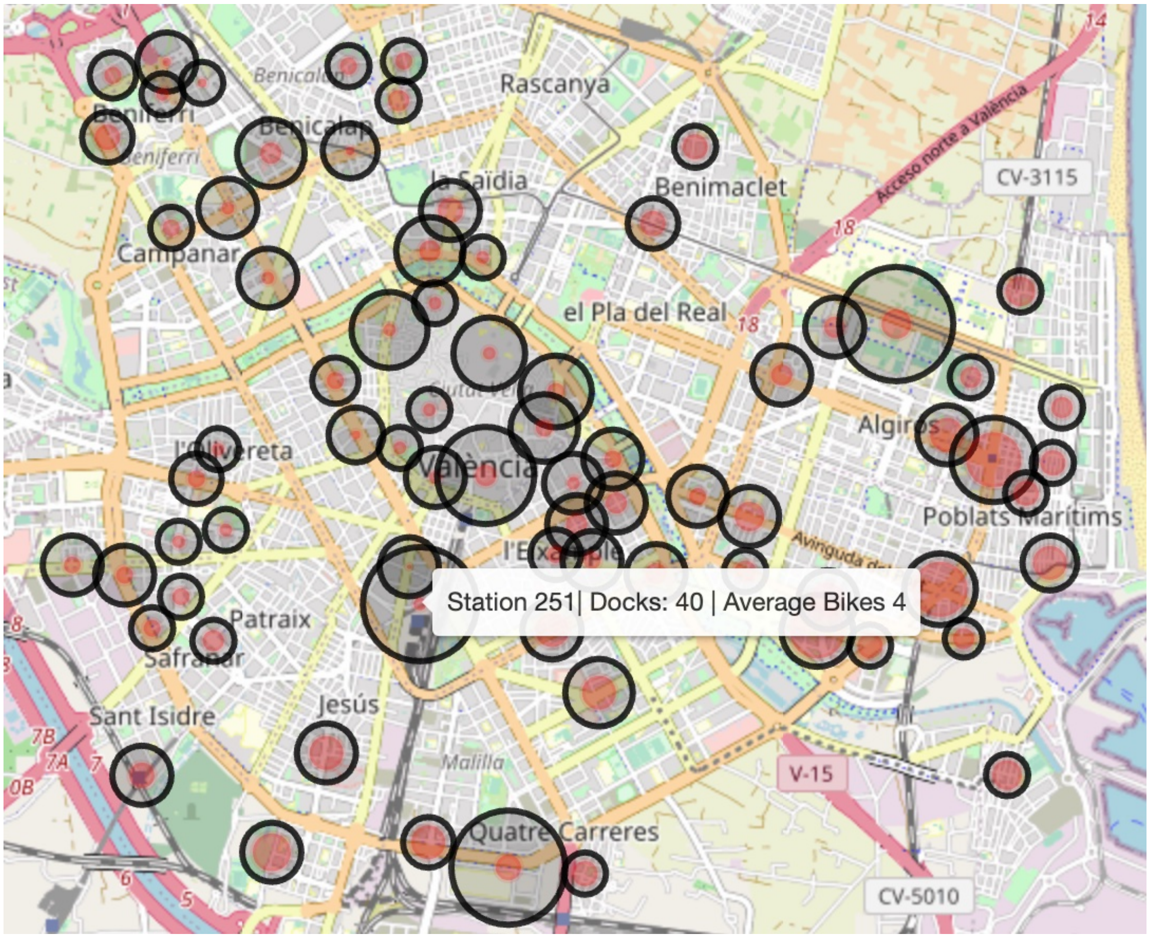

Figure 3: Station map of valencia; Grey marker size proportional to station capacity; red marker size proportional to average bicycle availability; Example tag shows statistics for Station 251 (from mouseover view of interactive map in notebook

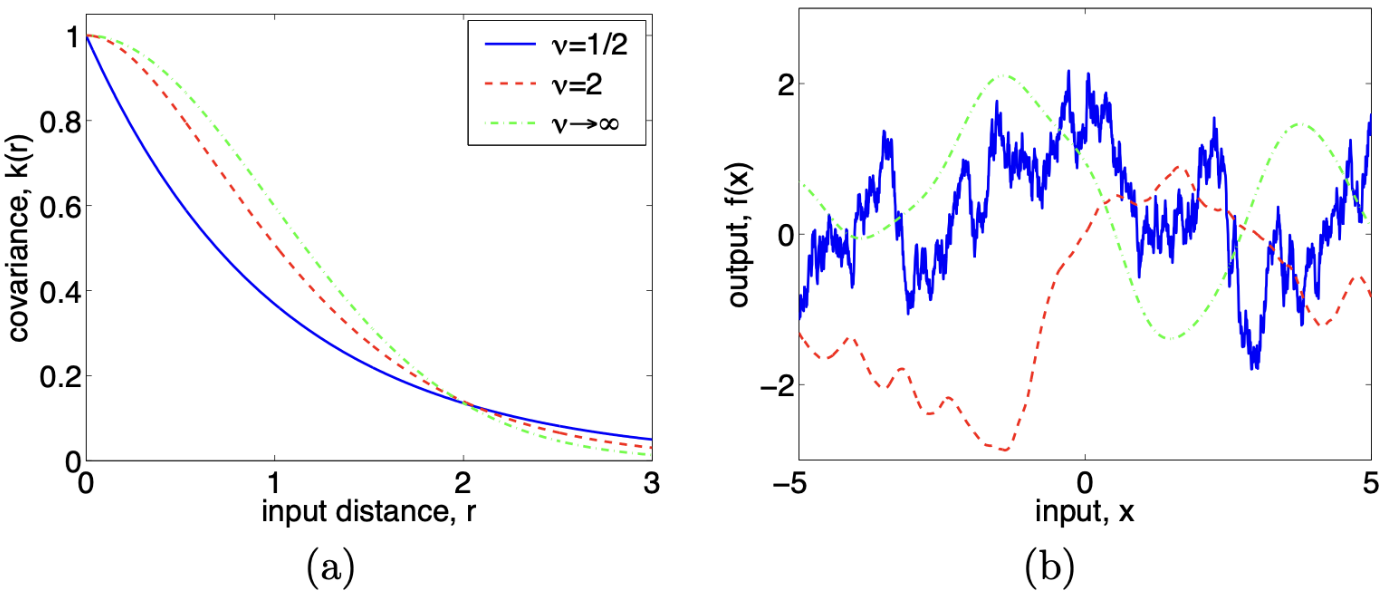

Figure 4: Panel (a): covariance functions, and (b): random functions drawn from Gaussian processes with Matérn covariance functions, eq. (1), for different values of ν, with $l = 1$. The sample functions on the right were obtained using a discretization of the x-axis of 2000 equally-spaced points.[1]

Data set	Parameter	Individual Model	Combined Model
Group A	$\nu = 0.5$	Training MAE = 3e-11 Validation MAE = 6.475	Failed
	$\nu = 0.5$ noise = 2.0	Training MAE = 0.3 Validation MAE = 1.82	Failed
	$\nu = 0.5$ noise = 3.0	Training MAE = 0.16 Validation MAE = 1.66	Failed
	$\nu = 0.5$ noise = 4.0	Training MAE = 0.02 Validation MAE = 1.60 Test MAE = 3.8	Failed
	$\nu = 0.5$ noise = 8.0.	Training MAE = 0.02 Validation MAE = 1.60	Failed
	$\nu = 1.0$ noise = 3.0	Training MAE = 0.92 Validation MAE = 1.98	Failed
	$\nu = 1.5$	Training MAE = 9.6e-12 Validation MAE = 7.478	Failed
	$\nu = 1.5$ noise = 2.0	Training MAE = 1.45 Validation MAE = 2.42	Failed
	$\nu = 1.5$ noise = 3.0	Training MAE = 0.95. Validation MAE = 1.89	Failed
	$\nu = 1.5$ noise = 4.0	Training MAE = 0.83 Validation MAE = 1.83 Test MAE = 4.54	Failed
	$\nu = 1.5$ noise = 8.0	Training MAE = 0.73 Validation MAE = 1.76	. Failed
	$\nu = 2.5$ noise = 2.0	Training MAE = 1.69 Validation MAE = 2.53	Failed
	$\nu = 2.5$ noise = 4.0	Training MAE = 0.92 Validation MAE = 1.86 Test MAE = 4.85	Failed
	$\nu = 2.5$ noise = 8.0	Training MAE = 0.97 Validation MAE = 1.86	Failed
Group B	$\nu = 0.5$ noise = 2.0	Training MAE = 0.38 Validation MAE = 1.87	Failed
	$\nu = 0.5$ noise = 4.0	Training MAE = 0.16 Validation MAE = 1.69 Test MAE = 3.55	Failed
	$\nu = 1.0$ noise = 2.0	Training MAE = 1.51 Validation MAE = 2.52	Failed
	$\nu = 1.0$ noise = 4.0	Training MAE = 0.68. Validation MAE = 1.81 Test MAE = 3.89	Failed
Group C	$\nu = 0.5$ noise = 2.0	Training MAE = 2.29 Validation MAE = 2.59 Test MAE = 2.63	Failed
	$\nu = 0.5$ noise = 2.0	Training MAE = 2.30 Validation MAE = 2.51	Failed
	$\nu = 1.5$ noise = 2.0	Training MAE = 2.48 Validation MAE = 2.69	Failed
Group D	$\nu = 0.5$ noise = 2.0	Training MAE = 0.87 Validation MAE = 2.14 Test MAE = 2.96	Failed
	$\nu = 0.5$ noise = 4.0	Training MAE = 0.87. Validation MAE = 2.14 Test MAE = 2.97	Failed

Table 5: Results for GPR Learning Algorithms on Bike Prediction Problem

Data set	Parameter	Individual Model	Combined Model
Group A	$n_{features} = 30$ $min_{samples} = 1$	Training MAE = 0.00 Validation MAE = 2.25	Training MAE = 0.00 Validation MAE = 2.97
	$n_{features} = 27$ $min_{samples} = 1$	Training MAE = 0.00 Validation MAE = 2.30	Training MAE = 0.00 Validation MAE = 3.02
	$n_{features} = 23$ $min_{samples} = 1$	Training MAE = 0.00 Validation MAE = 2.29	Training MAE = 0.00 Validation MAE = 2.98
	$n_{features} = 15$ $min_{samples} = 1$	Training MAE = 0.00 Validation MAE = 2.30	Training MAE = 0.00 Validation MAE = 3.08
	$n_{features} = 30$ $min_{samples} = 3$	Training MAE = 0.83 Validation MAE = 2.24 Test MAE = 3.81	Training MAE = 1.04 Validation MAE = 2.88 Test MAE = 3.65
	$n_{features} = 27$ $min_{samples} = 3$	Training MAE = 0.85 Validation MAE = 2.27	Training MAE = 1.05 . Validation MAE = 2.84
	$n_{features} = 23$ $min_{samples} = 3$	Training MAE = 0.89 Validation MAE = 2.28	Training MAE = 1.07 Validation MAE = 2.89
	$n_{features} = 15$ $min_{samples} = 3$	Training MAE = 0.96 Validation MAE = 2.31	Training MAE = 1.18 Validation MAE = 2.88
	$n_{features} = 30$ $min_{samples} = 5$	Training MAE = 1.25 Validation MAE = 2.26	Training MAE = 1.47 Validation MAE = 2.75
	$n_{features} = 27$ $min_{samples} = 5$	Training MAE = 1.27 Validation MAE = 2.26 Test MAE = 3.72	Training MAE = 1.47 Validation MAE = 2.75 Test MAE = 3.44
	$n_{features} = 23$ $min_{samples} = 5$	Training MAE = 1.29 Validation MAE = 2.28	Training MAE = 1.51 Validation MAE = 2.81
	$n_{features} = 15$ $min_{samples} = 5$	Training MAE = 1.39 Validation MAE = 2.33	Training MAE = 1.58 Validation MAE = 2.80
Group B	$n_{features} = 30$ $min_{samples} = 1$	Training MAE = 0.0 Validation MAE = 2.25	Training MAE = 0.0 Validation MAE = 2.96
	$n_{features} = 15$ $min_{samples} = 5$	Training MAE = 1.32 Validation MAE = 2.32	Training MAE = 1.53 Validation MAE = 2.79
Group C	$n_{features} = 3$ $min_{samples} = 10$	Training MAE = 2.09 Validation MAE = 2.74 Test MAE = 2.83	Training MAE = 2.07 Validation MAE = 2.96 Test MAE = 2.99
Group D	$n_{features} = 6$ $min_{samples} = 1$	Training MAE = 0.00 Validation MAE = 2.62	Training MAE = 0.00 Validation MAE = 3.39
	$n_{features} = 6$ $min_{samples} = 5$	Training MAE = 1.67 Validation MAE = 2.53	Training MAE = 1.83 Validation MAE = 3.04
	$n_{features} = 6$ $min_{samples} = 10$	Training MAE = 2.08 Validation MAE = 2.55 Test MAE = 3.09	Training MAE = 2.12 Validation MAE = 2.77 Test MAE = 2.91
	$n_{features} = 6$ $min_{samples} = 15$	Training MAE = 2.26 Validation MAE = 2.58	Training MAE = 2.34 Validation MAE = 2.85
	$n_{features} = 3$ $min_{samples} = 1$	Training MAE = 0.00 Validation MAE = 2.68	Training MAE = 0.00 Validation MAE = 3.33
	$n_{features} = 3$ $min_{samples} = 5$	Training MAE = 1.84 Validation MAE = 2.59	Training MAE = 2.04 Validation MAE = 2.96
	$n_{features} = 3$ $min_{samples} = 10$	Training MAE = 2.21 Validation MAE = 2.57 Test MAE = 3.14	Training MAE = 2.29 Validation MAE = 2.75 Test MAE = 2.77
	$n_{features} = 3$ $min_{samples} = 15$	Training MAE = 2.38 Validation MAE = 2.64	Training MAE = 2.52 Validation MAE = 2.92

Table 6: Results for Regression Tree Learning Algorithms on Bike Prediction Problem

Data set	Parameter	Individual Model	Combined Model
Group A	$n_{features} = 30$ $min_{samples} = 1$	Training MAE = 0.70 Validation MAE = 1.83	Training MAE = 0.82 Validation MAE = 2.26
	$n_{features} = 23$ $min_{samples} = 1$	Training MAE = 0.70 Validation MAE = 1.81 Test MAE = 3.05	Training MAE = 0.82 Validation MAE = 2.26 Test MAE = 2.59
	$n_{features} = 15$ $min_{samples} = 1$	Training MAE = 0.69 Validation MAE = 1.81	Training MAE = 1.80 Validation MAE = 2.24
	$n_{features} = 30$ $min_{samples} = 3$	Training MAE = 1.08 Validation MAE = 1.90	Training MAE = 1.22 Validation MAE = 2.30
	$n_{features} = 23$ $min_{samples} = 3$	Training MAE = 1.40 Validation MAE = 2.00	Training MAE = 1.52 Validation MAE = 2.33
	$n_{features} = 15$ $min_{samples} = 3$	Training MAE = 1.12 Validation MAE = 1.91 Test MAE = 3.11	Training MAE = 1.27 Validation MAE = 2.20 Test MAE = 2.51
	$n_{features} = 30$ $min_{samples} = 5$	Training MAE = 1.37 Validation MAE = 2.00	Training MAE = 1.51 Validation MAE = 2.33
	$n_{features} = 23$ $min_{samples} = 5$	Training MAE = 1.40 Validation MAE = 2.00	Training MAE = 1.52 Validation MAE = 2.33
	$n_{features} = 15$ $min_{samples} = 5$	Training MAE = 1.49 Validation MAE = 2.01	Training MAE = 2.01 Validation MAE = 2.34
Group B	$n_{features} = 30$ $min_{samples} = 1$	Training MAE = 0.70 Validation MAE = 1.83	Training MAE = 0.82 Validation MAE = 2.26
	$n_{features} = 15$ $min_{samples} = 5$	Training MAE = 1.39 Validation MAE = 1.99 Test MAE = 2.94	Training MAE = 1.54 Validation MAE = 2.32 Test MAE = 2.41
Group C	$n_{features} = 3$ $min_{samples} = 1$	Training MAE = 1.43 Validation MAE = 2.60	Training MAE = 1.40 Validation MAE = 2.71
	$n_{features} = 3$ $min_{samples} = 5$	Training MAE = 2.16 Validation MAE = 2.54	Training MAE = 2.10 Validation MAE = 2.69
	$n_{features} = 3$ $min_{samples} = 10$	Training MAE = 2.37 Validation MAE = 2.54 Test MAE = 2.61	Training MAE = 2.31 Validation MAE = 2.70 Test MAE = 2.79
Group D	$n_{features} = 6$ $min_{samples} = 1$	Training MAE = 2.37 Validation MAE = 2.54 Test MAE = 2.61	Training MAE = 2.31 Validation MAE = 2.70 Test MAE = 2.92
	$n_{features} = 6$ $min_{samples} = 5$	Training MAE = 1.76 Validation MAE = 2.28 Test MAE = 2.75	Training MAE = 1.76 Validation MAE = 2.47 Test MAE = 2.46
	$n_{features} = 6$ $min_{samples} = 10$	Training MAE = 2.12 Validation MAE = 2.40 Test MAE = 2.74	Training MAE = 2.09 Validation MAE = 2.52 Test MAE = 2.44
	$n_{features} = 3$ $min_{samples} = 1$	Training MAE = 0.80 Validation MAE = 2.13 Test MAE = 2.95	Training MAE = 0.85 Validation MAE = 2.32 Test MAE = 2.54
	$n_{features} = 3$ $min_{samples} = 5$	Training MAE = 1.87 Validation MAE = 2.31	Training MAE = 1.94 Validation MAE = 2.47
	$n_{features} = 3$ $min_{samples} = 10$	Training MAE = 2.19 Validation MAE = 2.42 Test MAE = 2.78	Training MAE = 2.20 Validation MAE = 2.53 Test MAE = 2.41

Table 7: Results for Random Forest Regression Learning Algorithms on Bike Prediction Problem

Model Type	Filtered/ Unfiltered	MAE (Training)	MAE (Test)
full temp	Filtered	2.841	2.29
full temp	Unfiltered	2.933	2.40
full	Filtered	2.838	2.28
full	Unfiltered	2.923	2.40
short full temp	Filtered	2.841	2.29
short full temp	Unfiltered	2.931	2.39
short full	Filtered	2.840	2.28
short full	Unfiltered	2.926	2.40
short temp	Filtered	2.844	2.36
short temp	Unfiltered	2.933	2.47
short	Filtered	2.840	2.35
short	Unfiltered	2.930	2.47

Table 8: Results for Linear Models on Bike Prediction Problem