More Bikes: Experiments in Univariate Regression

Tim Lawson

December 22, 2023

1 Task description

The assignment is to predict the number of available bikes at 75 rental stations in three hours' time for a period of three months, beginning in November 2014, i.e., a supervised univariate regression problem. It is divided into three sub-tasks, which differ in the information that is available:

- **Sub-task 1**. The number of available bikes at each of the 75 stations for the month of October 2014. This sub-task may be approached by building a separate model for each station or a single model for all 75 stations.
- **Sub-task 2**. A set of linear models that were trained on the number of available bikes at each of a separate set of 200 stations for a year. For the first ten stations, this data is available for analysis but not training.
- **Sub-task 3**. Both of the above.

Sub-tasks 2 and, optionally, 3 require the use of ensemble methods. The predictions are evaluated by the mean absolute error (MAE) between the predicted and true numbers of available bikes over the period of three months, beginning in November 2014. The evaluation data is not available to participants but the score achieved on a held-out test set is reported on the task leaderboard. This report begins with a preliminary analysis of the data and then describes the approach taken to each sub-task and the cross-validation results obtained.

2 Data analysis

The data is recorded at hourly intervals. A summary of the features is given in fig. 1. Notably, the meteorological features are constant for all stations at a given timestamp. The 'profile' features, i.e., the features derived from the numbers of available bikes at preceding times, are not defined for the first week of instances at each station, except for `bikes_3h`. Naturally, the number of available bikes at a given station is bounded by zero and the number of docks at that station. The variances and pairwise correlations of the features are described in section 3.1.

3 Methods

The experiments in this report were conducted with the `scikit-learn` Python package (Pedregosa et al. 2011). In each case, preprocessing and feature selection were performed by *estimators* that implemented the *transformer* interface; prediction was performed by estimators that implemented the *predictor* interface; and estimators were composed into `Pipeline` objects over which hyperparameter search was performed (Buitinck et al. 2013, pp. 4–9). The bounds on the number of available bikes at a given

Category	Feature	Data type	Kind	Variance
Station	`station`	`int`	ordinal	–
	`latitude`	`float`		1.68×10^{-4}
	`longitude`	`float`		5.03×10^{-4}
	`docks`	`int`		3.28×10^{1}
Temporal	`timestamp`	`int`		5.99×10^{11}
	`year`	`int`		5.71×10^{-26}
	`month`	`int`		0
	`day`	`int`		8.00×10^{1}
	`hour`	`int`		4.80×10^{1}
	`weekday`	`str`	ordinal	–
	`weekhour`	`int`		2.17×10^{3}
	`is_holiday`	`bool`	categorical	–
Meteorological	`wind_speed_max`	`float`		7.51×10^{1}
	`wind_speed_avg`	`float`		2.10×10^{1}
	`wind_direction`	`float`		7.55×10^{3}
	`temperature`	`float`		1.07×10^{1}
	`humidity`	`float`		2.80×10^{2}
	`pressure`	`float`		1.81×10^{3}
	`precipitation`	`float`		0
Bikes	`bikes`	`int`		4.31×10^{1}
	`bikes_avg_full`	`float`		3.58×10^{1}
	`bikes_avg_short`	`float`		3.58×10^{1}
	`bikes_3h`	`int`		4.32×10^{1}
	`bikes_3h_diff_avg_full`	`float`		2.24×10^{1}
	`bikes_3h_diff_avg_short`	`float`		2.24×10^{1}

Figure 1: A summary of the features of the data. Except where indicated, the features are quantitative. The variances of the quantitative features are discussed in section 3.1 and their pairwise correlations are shown in fig. 3.

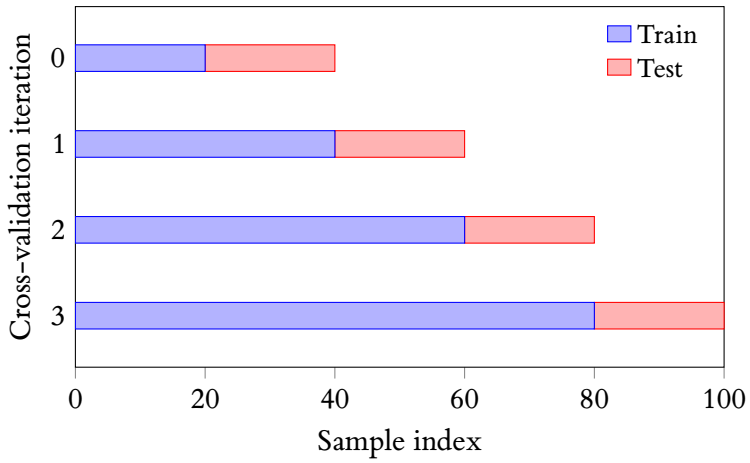

Figure 2: A visualisation of the nested time-series cross-validation behaviour, after the visualisation of `sklearn.model_selection.TimeSeriesSplit`.

station were enforced by predicting the *fraction* of bikes, i.e., the number of bikes divided by the number of docks at the station. This was implemented by extending the `TransformedTargetRegressor` meta-estimator to permit data-dependent transforms[1].

Generally, standard k-fold cross-validation is disfavoured for time-series data due to the inherent correlation between successive folds (Bergmeir et al. 2018). Instead, nested time-series cross-validation[2] with ten folds was performed to evaluate the models. This behaviour is illustrated in fig. 2.

- Hyperparameter search.
- Evaluation metric.
- Statistical significance tests (paired t-tests).

3.1 Feature selection

In each case, zero-variance features were automatically excluded[3] because they are individually uninformative. The available data for sub-task 1 is limited to the month of October 2014; therefore, the month and year were excluded. Additionally, the 'station' features (fig. 1) are constant for all instances at a given station; hence, for the first case of sub-task 1, the variances of these features are zero. Finally, the `precipitation` feature is zero for all instances.

To determine which features are redundant, i.e., uninformative in combination, the Pearson correlation coefficients between pairs of quantitative features were computed (fig. 3). This analysis yielded the following observations:

- `bikes_3h_diff_avg_full` and `bikes_3h_diff_avg_short` are fully correlated ($r = 1.00$).
- `bikes_avg_full` and `bikes_avg_short` are fully correlated ($r = 1.00$).
- `wind_speed_max` and `wind_speed_avg` are highly correlated ($r = 0.96$).

Hence, the second of each of these pairs of features was manually excluded.

1: Describe the distributions of the fraction of available bikes in terms of temporal features and the possible derived features.

[1] `sklearn.compose.TransformedTargetRegressor`
[2] `sklearn.model_selection.TimeSeriesSplit`
[3] `sklearn.feature_selection.VarianceThreshold`

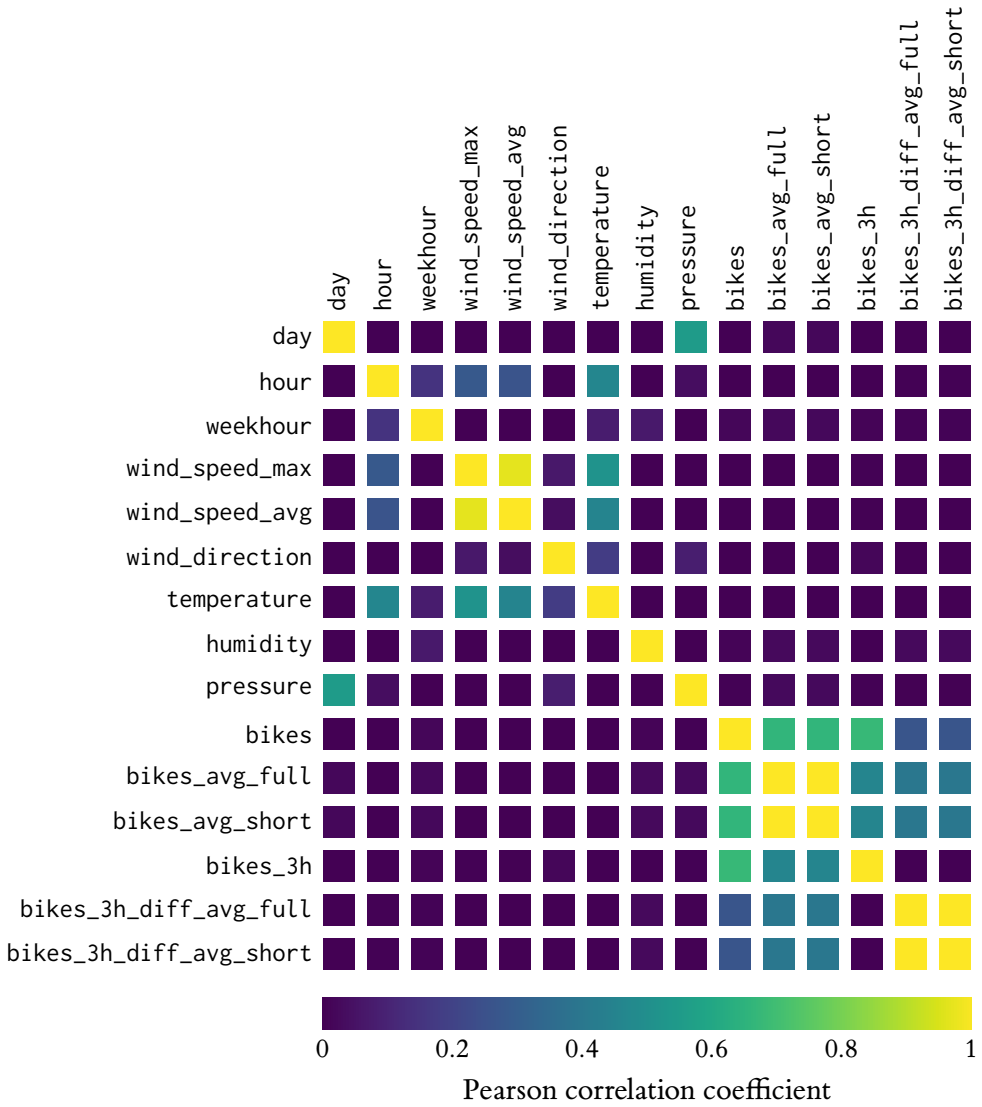

Figure 3: The Pearson correlation coefficients between pairs of quantitative features. The ordering is the same as in fig. 1. Most of the temporal features are excluded, because they are obviously correlated. The zero-variance 'station' and `precipitation` features are also excluded (section 3.1).

3.2 Model selection

- Baseline 4.434 / 5.449
- Decision tree

References

Bergmeir, Christoph, Rob J. Hyndman, and Bonsoo Koo (2018). "A Note on the Validity of Cross-Validation for Evaluating Autoregressive Time Series Prediction". In: *Computational Statistics & Data Analysis* 120, pp. 70–83.

Buitinck, Lars, Gilles Louppe, and Mathieu Blondel (2013). "API Design for Machine Learning Software: Experiences from the Scikit-Learn Project". In: *ECML/PKDD 2013 Workshop: Languages for Data Mining and Machine Learning*.

Pedregosa, Fabian et al. (2011). "Scikit-Learn: Machine Learning in Python". In: *The Journal of Machine Learning Research* 12, pp. 2825–2830.