Coursework: Machine Learning Models for Rented Bikes Dataset

2208458

I. Introduction

In the last decade, Machine Learning models have become a very important tool for most online companies. Machine learning is a type of Artificial Intelligence that can go over a set of data and predict future outcomes of the data [4]. "Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products" [4]. This enables machines to learn about the costumers and try to make decisions based on the input data, without any human interaction needed [5]. Utilizing this models can also bring benefits to the companies in respect to their competition, "...can be the key to unlocking the value of corporate and customer data and enacting decisions that keep a company ahead of the competition" [5].

For this project, a data set for rented bikes is provided. This data set consists of the number of bikes rented at each hour together with other features such as the date, temperature, humidity, wind speed, etc. The goal is to create two machine learning models, along with a baseline model to compare them, that can predict the number of rented bikes based on all the other features. To do this, we first have to establish which type of problem this is so our machine learning models can be as robust as possible on their predictions.

In machine learning, there are various types of learning algorithms, and most of them fall in one of two categories. Supervised Learning algorithms is one of them. These algorithms are used when we have a data set that is labeled, this means that we have data that is already tagged with the correct answer [3]. The second category is the Unsupervised learning algorithm. With these algorithms, the data is not labeled and the algorithm has to work on it's own to find patterns on the data [3]. Since our data is labeled and has a lot of features, our problem can be solved with a supervised learning algorithm. However, these algorithms are separated into classification and regression algorithms, which are distinguished by their output.

- **Classification:** These types of algorithms group their output inside a class [3]. This means that they have a certain categorization for the output.
- **Regression:** These types of algorithms just have one output value [3]. For instance, it's output could be the price of an item.

As mentioned before, our algorithm has a lot of features and all of them are labeled, so we can assume that the algorithm that best fits the data would be a Supervised Learning algorithm. Moreover, the value that we want to predict is the number of rental bikes, so it is one value that can't be categorized in a group, thus a Regression algorithm would be best for this problem.

II. Methods

A. Models and Metrics

For the project, the models that are going to be used are Polynomial regression and a Decision Tree regressor. Before defining how the performance of the algorithms is going to be tested, it's always better to visualize the data and see how the features relate to our target (number of rented bikes), also to get a sense of how the data is distributed to select a good metric for our algorithms. Keep in mind that some features (such as seasons and holiday) were manipulated so the values correspond to a number instead of a word.

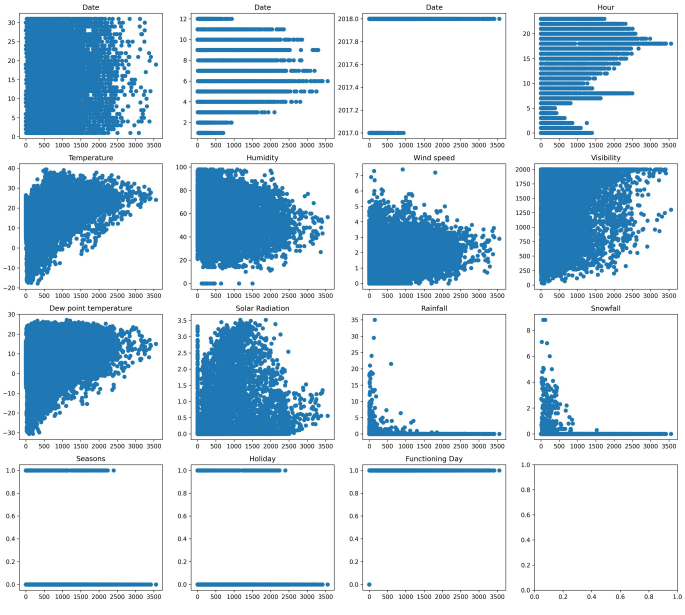

Fig. 1. Visualization of the features against the target value.

The metrics used to find the level of error in machine learning models are designed specifically for the type of algorithm. Since the algorithms selected are regression algorithms, the normal metrics for this algorithms are mean squared error, mean absolute error (MAE) and r-squared (R^2). On this project the MAE and the R^2 are going to be used.

- **Mean Absolute Error:** This metric is defined by the following formula:

$$E = \frac{1}{N} \sum_{(\vec{x},y)} |y - f(\vec{x})|^2$$

Where N is the total number of values, y is the real value and $f(\vec{x})$ is the predicted value. What this metric does is calculate the MAE between the real point and the predicted point and return the mean of all mean absolute errors.

This metric was selected because of how the data is distributed, it is normally used when there are outlier points, due to "MSE and RMSE punish larger errors more than smaller errors, inflating or magnifying the mean error score. This is due to the square of the error value. The MAE does not give more or less weight to different types of errors and instead the scores increase linearly with increases in error" [1]. As we can see in Figure 1, some plots, such as temperature, humidity, wind speed, the data is very close together, but there are a lot of outlier points which can result in a big error that may not represent the true performance of the models.

- **R-squared:** This metric is described by the following formula:

$$R^2 = 1 - \frac{\sum_{(\vec{x},y)} (y - f(\vec{x}))^2}{\sum_{(\vec{x},y)} (y - \bar{y})^2}$$

Where y is the real value, $f(\vec{x})$ is the predicted value and $\bar{y}$ is the mean of all the target values y. This metric is very common when using linear regression models, because "R-squared measures the strength of the relationship between your model and the dependent variable on a convenient 0 – 100% scale" [2]. Since one of the models being used in this project has to do with linear regression, based on theory, this is the best metric to measure the performance of the model.

Both of these metrics allow for good evaluation of the models when presented to common cases, for instance, on Figure 1, the common cases would be where the points are closer together, the uncommon cases would be the outlier point.

Now that the models and metrics have been established, we need to have a baseline model to compare our models to. The baseline model is done with the help of the library `sklearn.dummy`, which has a Dummy regressor model. The Dummy regressor predicts values based on mathematical properties of the data, such as, median, mean or quantile. In this case, the best measure for the data would be the median of the data, the prediction for the model is the median of the data. Since we have some outlier points, the median would be more representative than the mean, because, just as the metrics, the mean would inflate the prediction. "It is best to use the median when the distribution is either skewed or there are outliers present" [7].

B. Hyperparameter Selection

Before training and testing the models, a good hyperparameter selection is needed. Hyperparameters are the parameters of the model that can be set on creation of the model. For both models, the process was equal, some variables were created, from which, the values for the hyperparameters were set. Cross validation was done on the training data with 5 splits (more than 5 made the program run very slow) to iterate through all the variables, also the MAE was calculated for all the iterations. The hyperparameters with the lowest values for our metrics were saved, also plots were made based on the mean of the calculated MAE to visualize the effects of the different hyperparameters on the models.

1) Decision Tree Regressor: For this model, three hyperparameters were taken into a count to measure the best performance. The criterion, the splitter and the max depth. The criterion corresponds to the metric we want to use to split an internal node, such as MSE or MAE; the splitter is "The strategy used to choose the split at each node" [6]; the max depth is the max depth wanted from the root to the leaves of the tree. Each combination was plotted to visualize the effects of the hyperparameters on predicting the data.

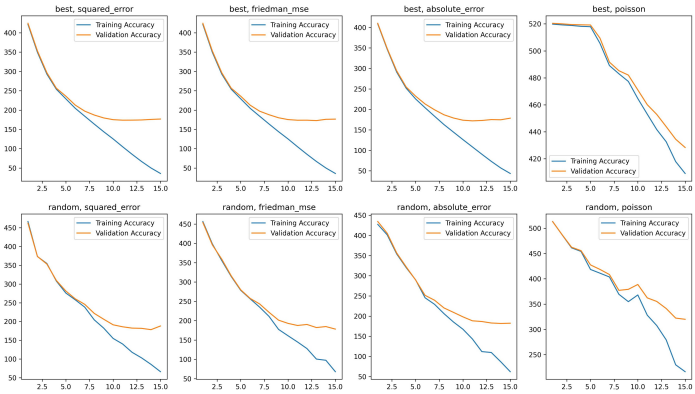

Fig. 2. Visualization of the different hyperparameter selection for a Decision Tree regressor.

Looking at the plots above, we can see that the first three plots that have the *best* splitter, have the lowest values for the MAE, which means that they could be the best fit for the data. Since looking at a plot may not be too accurate, the hyperparameters for the lowest MAE were saved, to have the exact hyperparameters at which the result was gotten. On Figure 2, we can see that, for the splitter *best*, two criterion were very close, *friedman_mse* and *absolute_error*. When running the code, the result for the hyperparameters that gave the lowest MAE alternates between these two, with values for max depth of 12 and 11 respectively (also very close values). This means that in both cases the predictions are very close and as a result we can safely say that both of these metrics work for the data set. Hence, the hyperparameters that were selected for the tree were *best* for the splitter, *friedman_mse* for the criterion and *12* for the max depth.

2) Polynomial Regression: This model finds the polynomial function that best fits the data. For this model, only one hyperparameter was used, due to the other ones not having any effect on the model's performance. The hyperparameter tested was the degree of the polynomial function. In this case, cross-validation was not used because the program ran very slowly. For each iteration, the result for the r^2 score was saved and plotted to look at how the degree adapted to the data. The results are shown in the below figure.

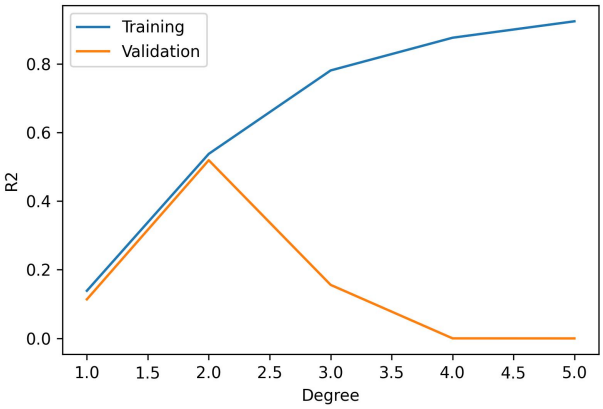

Fig. 3. Visualization of the hyperparameter selection for Polynomial Regression.

Looking at Figure 3 it is very noticeable that the value for the degree, that best adapts to the data, is 2, where the r^2 score is maximum.

III. Results

After selecting the hyperparameters for the models, the next step is to train the models, test the models and look how well they could predict the data. Keep in mind the results can vary with each program execution, but the difference between the results is very slim.

A. Polynomial Regression

The results obtained for the prediction using polynomial regression are summarized in Figure 4, where the blue dots represent the real value and the orange dots represent the predicted value.

For Polynomial Regression, as mentioned before, the best metric to use is r^2 because it uses a linear regression model to predict the data. The value for r^2 was 0.678, which means that, based on this metric, the model could predict 67% of the values correctly. Looking at Figure 4 we can see that the predictions were more focused on the parts were the points are closer together. These results are expected because the model tries to fit a polynomial line on the data, the predictions will be then based on that line and since the data is very close together on most cases, the line will take part where the data is closer together, leaving bad predictions for the outlier points.

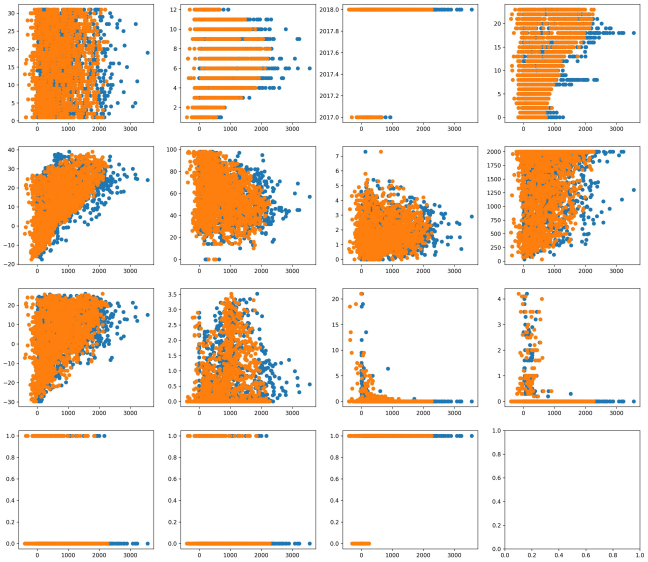

Fig. 4. Prediction results for Polynomial Regression.

Fig. 5. Prediction results for Decision Tree Regressor.

B. Decision Tree Regressor

The results obtained for the predictions using this model are shown in Figure 5, where the blue dots represent the real value and the orange dots represent the predicted value.

The metric used for the Decision Tree Regressor was the mean squared error and the result was 155.141, this means that when predicting values the average difference between the real value and the predicted value was 155.141. Again these results are expected due to how the data is distributed. Decision Trees, are normally known for over-fitting the data. Because of the hyperparameters that were chosen, we can see on Figure 5 that this was not the case for the data set. We can see that most of the outlier points are not well predicted, however, the predicted values are not far from the real value, this can be

seen with the closeness of the orange dots in respect to the blue dots.

C. Results in Comparison

Now that the results for both of the models have been analyzed, we can compare the results between them and also with the Dummy Regressor to really understand how robust the predictions are for each one.

TABLE I
PREDICTION RESULTS FOR ALL MODELS USING DIFFERENT METRICS.

	Dummy Regressor	Polynomial Regression	Decision Tree Regressor
MAE	496.136	260.957	154.973
R^2	-0.060	0.678	0.819

Looking at Table I, we can see that the Dummy regressor did a good job setting the baseline for the models. The Dummy regressor had a mean squared error almost double compared to the Polynomial Regression and more than triple for the Decision Tree Regressor. This means that both of the selected models could learn from the training data set and make good predictions based on this data. When comparing the two models that were selected, we can see that on the MAE the difference is very big. We can visualize this difference when looking at Figure 4 and Figure 5, on Figure 4, as mentioned before, the predicted points are more focused on the dots that are close together, on the contrary, we can see that the Decision Tree (Figure 5) could predict values not only where the points are closer together, but also on the outlier points. This means that the Polynomial Regression did not adapt to the randomness on the dataset as well as the Decision Tree did. We can also see that for the R^2 metric the Decision Tree also had a better performance, which means that this model could relate more to the dataset. These results are expected, due to how the models interpret the data. Polynomial Regression only takes into a count the points that are close to the polynomial line used to fit the data, whereas the Decision Tree, can split it's nodes with more precision to better understand the data.

After looking at the results, we can see that in general both of the models could generalize to unseen data. This is true for Polynomial regression only when the data being predicted is close to the modeled polynomial line. As a result, the Decision tree can generalize more to unseen data than the Polynomial Regression model.

REFERENCES

[1] J. Brownlee. Regression Metrics for Machine Learning, 2021. URL: https://machinelearningmastery.com/regression-metrics-for-machine-learning/.

[2] J. Frost. How To Interpret R-squared in Regression Analysis - Statistics By Jim, 2017. URL: https://statisticsbyjim.com/regression/interpret-r-squared-regression/HowToInterpretR-squaredinRegressionAnalysis.

[3] D. Johnson. Supervised vs Unsupervised Learning: Key Differences, 2022. URL: https://www.guru99.com/supervised-vs-unsupervised-learning.html.

[4] Kent State University. What is Experiential Learning and Why Is It Important?, 2022. URL: https://www.techtarget.com/searchenterpriseai/definition/machine-learning-MLhttps://www.kent.edu/community/what-experiential-learning-and-why-it-important.

[5] Netapp. What Is Machine Learning (ML) and Why Is It Important? — NetApp, 2020. URL: https://www.netapp.com/artificial-intelligence/what-is-machine-learning/.

[6] Sklearn. sklearn.tree.DecisionTreeRegressor — scikit-learn 1.0.2 documentation, 2022. URL: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html.

[7] Zach. When to Use Mean vs. Median (With Examples), 2021. URL: https://www.statology.org/when-to-use-mean-vs-median/https://www.statology.org/when-to-use-mean-vs-median/#:$\sim$:text=It'sbesttousethemeanwhenthedistributionof,whentherearecleareoutliers.