Unreliable Sensors

Jannik Sheikh, 3873496
/graded

Submission by 15th March 2021

1 Introduction

The second part of the final project of the course «Introduction to Machine Learning»was analyzing and cleaning a messy data set of ordered sensor measurements, which were collected over time. Since our final goal was to predict the target variable for unseen data, we can categorize this machine learning task as supervised learning, regression. The used model is Linear Regression. R^2 and MSE were chosen as evaluation metrics as well as cross validation for model selection and hyperparameter tuning. Furthermore different data cleaning and preproccessing steps had to be conducted.

2 Data Set

The original data set contains of 7.432 rows and 12 features, as well as the target variable y for each 7.432 observations. The columns labels are numbers from 0 to 11, and no further information were given, see Figure 1. Figure 2 is a visualization of each feature in orderly time and showing that Feature 0 is an outlier, due to the fact that it probably represents the index, with abrupt changes in between. Figure 4 shows a gap right before index 3000 for feature 1. The reason for that are missing values. Further investigation revealed, that Feature 5 and 7 have missing values as well, but much less than Feature 1 and thereby not visible on any plot. All features beside feature 8 are right skewed which we have to address during our preprocessing, see Figure 3 as an example. Moreover feature 10 has 12 observations with a negative value. Lastly no duplicated entries where found.

3 Data Cleaning and Preprocessing

We first decided to drop the feature 0 of the mentioned reasons in section 2. Since we only had so few observations with a negative value for feature 10, we decided to drop those, but left the order intact. To address the skewness we applied a cube root transformation, which resolved that our features, besides feature 8, were then between -0.5 or 0.5 and thereby fairly symmetrical. To calculate the skew value, we used the *skew()* method from *pandas*. Feature 8 was already normal distributed. For the missing values we used the method *interpolate* from *pandas*. Here, we fill the missing values by considering data points based on our index. Afterwards we standardized our features, by extracting the mean and dividing by the standard deviation. Features transformed this way have approximate 0 mean and variance of 1.

4 Model

We used Linear Regression to solve this regression task. Linear regression is a statistical method to fit a line (or a hyperplane in multiple dimensions) to the input data X and our target y. Usually, it is modeled as

$$y_i = \beta_0 + \beta_1 \cdot X + \beta_2 \cdot X + \cdots + \beta_n \cdot X,$$

where β_0 is the intercept and all other β's are the coefficients for the slope of the line/plane.

5 Evaluation

For the evaluation of our models, we used 10-fold cross-validation. Together with the R^2 and the MSE metrics.

- R^2 describes how much of the variance in the data is explained by the model. Values are between $[-\infty, 1]$, where 1 means we explain the variance in our data perfectly and negative values imply a bad fit of the model. The formula is
$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i}y)^2}{\sum_{i=1}^{n}(y_i - \bar{y_i})^2},$$
where
$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$
and
$$\sum_{i=1}^{n}(y_i - \hat{y_i})^2 = \sum_{i=1}^{n}\epsilon_i^2,$$
the sum of squared residuals.

- MSE is the mean squared error of our model and the true data points. Lower values imply a better fit of the model. The formula is
$$MSE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y_i})^2.$$
$\hat{y}$ is the predicted value, y the true value.

6 Feature Selection

With feature selection we want to detect the most relevant features from our dataset and remove all features that don't contribute to the model performance. This can help to increase the computational efficiency of the models and reduce training times. Lower feature count can also help with generalization of the model and reduce overfitting. For Linear Regression the feature selection can be considered as hyperparameter tuning. We used the wrapper method *Recursive feature elimination*, which selects features by recursively considering smaller and smaller sets of features [1]. We first train on the whole set of features, and then remove the feature with the lowest impact from the current set. After that we get a feature ranking. With the help of the *sklearn* implementation of recursive feature elimination cross validation (RFECV), we can embed cross validation in this process of selecting the best number of features. We used 10-fold cross validation for evaluation.

7 Experiments

7.1 Set-Up

After applying our data cleaning and preprocessing techniques, described in 3, we splitted our data into training and testing sets. 30% were hold out for validation. Subsequently we fitted and predicted, using 10-fold cross validation, the performance of our model without any feature selection. Afterwards we applied our mentioned feature selection method from 6 and evaluated again.

7.2 Results

Table 1 and Table 2 show the evaluation of our model, with and without the use of feature selection on our preprocessed data set. For both train and test set, our R2 score is the same. Our MSE score though is a bit lower on the train set, for the model with feature selection. On our validation set, the MSE is 1 point higher for the model with feature selection, than without.

8 Analysis and Discussion

Overall the results are sufficient. Our model fits the training and validation set pretty good, since we see no big difference between the training and test score. Feature selection had no significant impact on our model. This is probably due to the fact that we only dropped one feature, feature 1, after our feature selection step. This means our model without feature selection uses all 11 features and the model after feature selection uses 10 features. Only the MSE score differ slightly from each other but it is insignificant. Further research on larger data sets, with more features should be considered, to get a greater effect of feature selection. For comparison other regression models could be tested, with and without feature selection.

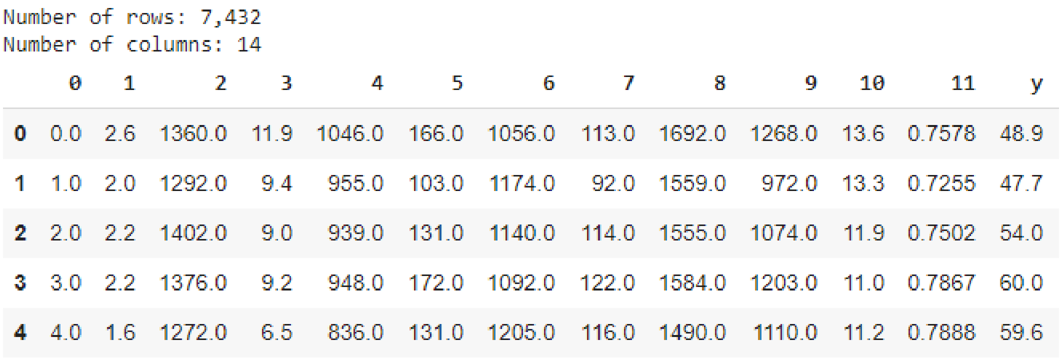
9 Appendix

Model	R2 Score	MSE
without Feature Selection	0.897	31.844
with Feature Selection	0.897	31.682

Table 1: Evaluation on Train Set

Model	R2 Score	MSE
without Feature Selection	0.881	35.798
with Feature Selection	0.881	35.799

Table 2: Evaluation on Test Set



```
Number of rows: 7,432
Number of columns: 14
```

	0	1	2	3	4	5	6	7	8	9	10	11	y
0	0.0	2.6	1360.0	11.9	1046.0	166.0	1056.0	113.0	1692.0	1268.0	13.6	0.7578	48.9
1	1.0	2.0	1292.0	9.4	955.0	103.0	1174.0	92.0	1559.0	972.0	13.3	0.7255	47.7
2	2.0	2.2	1402.0	9.0	939.0	131.0	1140.0	114.0	1555.0	1074.0	11.9	0.7502	54.0
3	3.0	2.2	1376.0	9.2	948.0	172.0	1092.0	122.0	1584.0	1203.0	11.0	0.7867	60.0
4	4.0	1.6	1272.0	6.5	836.0	131.0	1205.0	116.0	1490.0	1110.0	11.2	0.7888	59.6

Figure 1: Inital Data Set

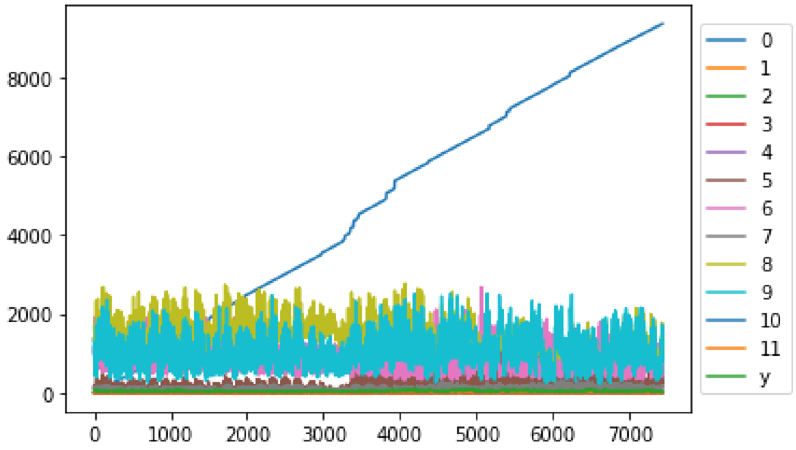


Figure 2: Plot of Inital Data Set

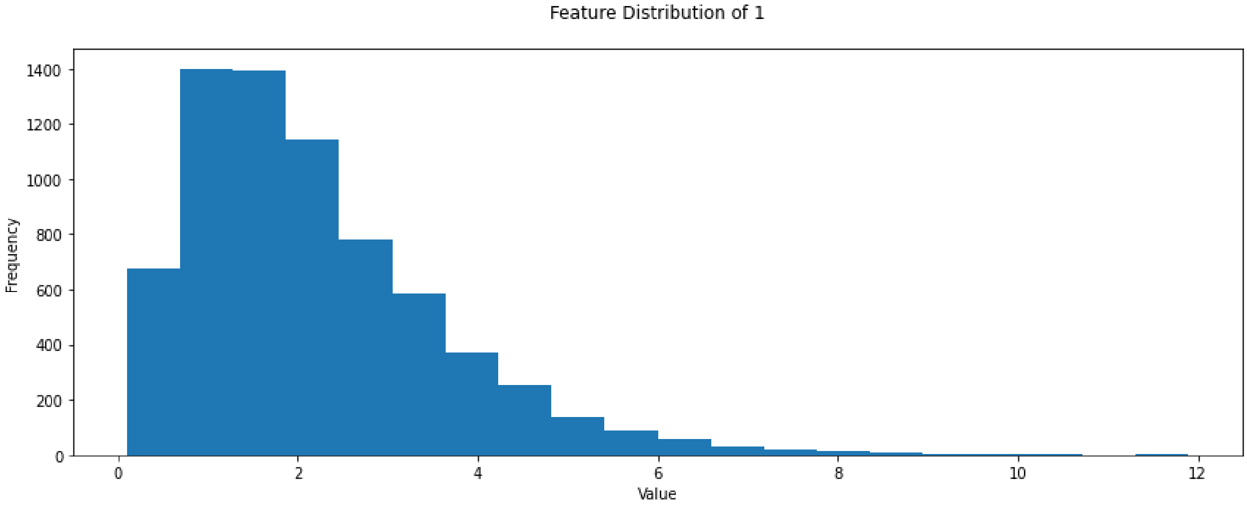


Figure 3: Right skewed distribution Feature 1

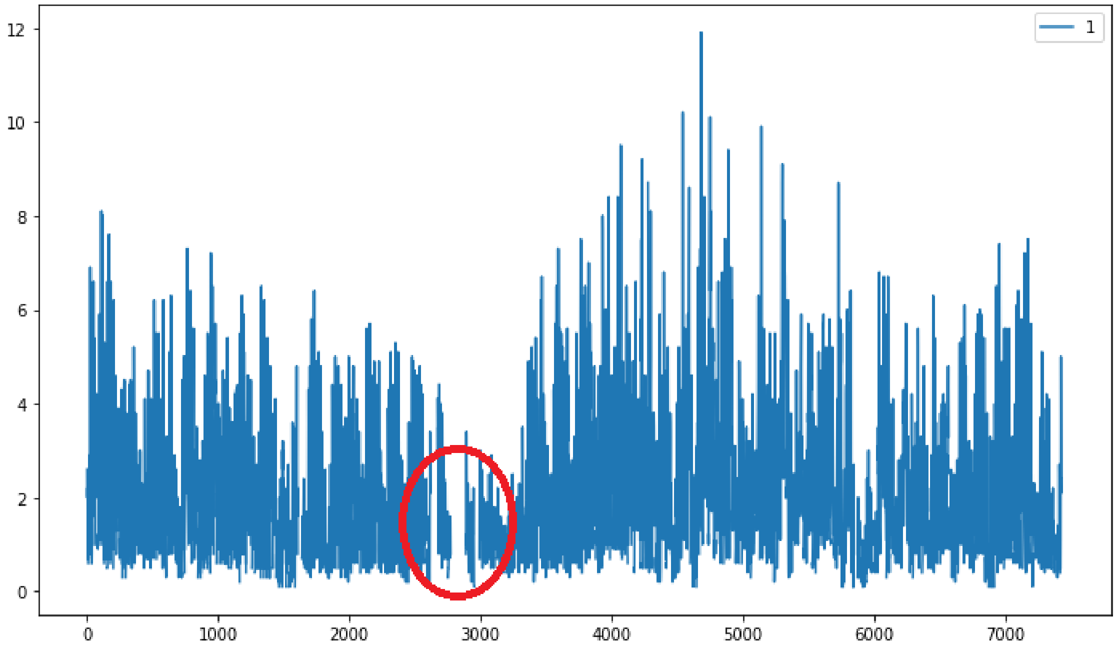


Figure 4: Plot of Feature 1 with gap

References

[1] “sklearn.feature$_{s}election.rfe.''$. accessed March 12, 2021.