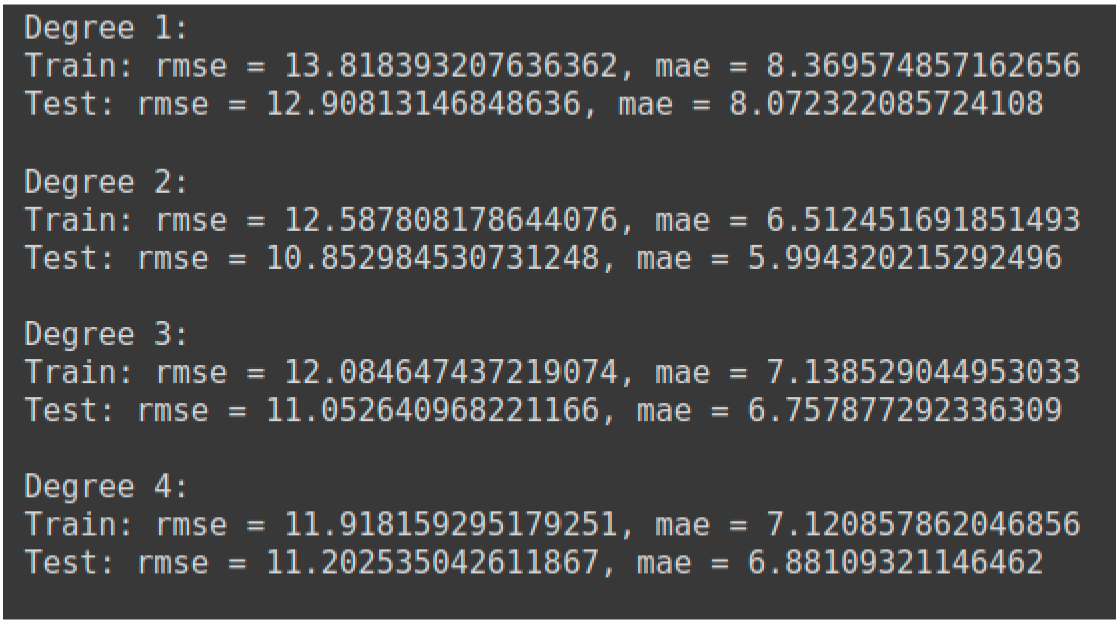
Andrey Vagin
a.vagin@innopolis.university

Regarding the Preprocessing

Which regression model was the most effective for the missing values, and why?

- Multinomial regression of 2-nd degree.



```
Degree 1:
Train: rmse = 13.818393207636362, mae = 8.369574857162656
Test: rmse = 12.90813146848636, mae = 8.072322085724108

Degree 2:
Train: rmse = 12.587808178644076, mae = 6.512451691851493
Test: rmse = 10.852984530731248, mae = 5.994320215292496

Degree 3:
Train: rmse = 12.084647437219074, mae = 7.138529044953033
Test: rmse = 11.052640968221166, mae = 6.757877292336309

Degree 4:
Train: rmse = 11.918159295179251, mae = 7.120857862046856
Test: rmse = 11.202535042611867, mae = 6.88109321146462
```

As we can see 1-st degree (usual linear regression) is underfitted in comparison with the 2-nd degree. 3-rd and 4-th are overfiited in comparison with the 2-nd degree.

What encoding technique did you use for encoding the categorical features, and why?

var3:

We have 236 different categories in var3 columns. Almost half of them has 1, 2 or 3 occurances in the table. We may have not enough data for these classes to extract some information for our model.

Ordinal encoding technique will not give us meaningful represenation for our data. We have 236 different categories. Such ecncoding will create big numeric difference between first and last classes but there is no such difference in location names. Ordinal encoding is not suitable for this column.

One-hot encoding techique in our case will give big growth in number of columns where information will be very sparse because we have a lot of classes and not that very big amount of data.

My choise in this case try to use one-hot encoding and try to drop var3 column and compare the results.

var6:

In var6 we are having only two categories. There is a good practice for linear regression usage to drop one of the categories to avoid collinearity in data columns. If we do so with One-hot encoding on binary data it will be equal to ordinal encoding.

I encoded ‘yes’ with 1 and ‘no’ with 0 in one column.

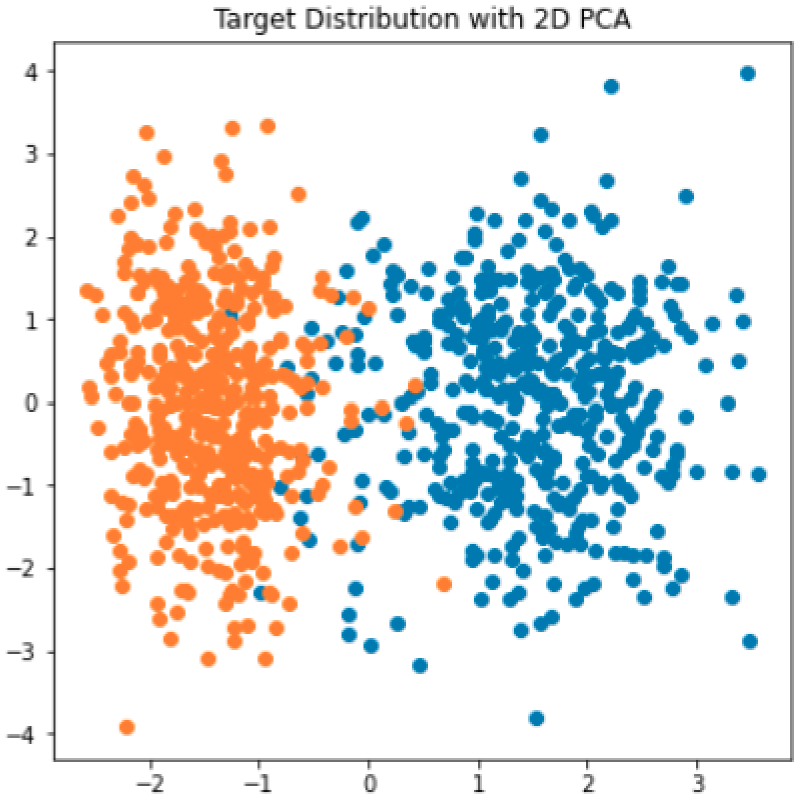
var7:

In var7 we are having dates from 2019 year. I decided to convert all this dates into number of seconds which passed from the beginning of 2019. Such transformation do not break meaning inhereted in the date data.

Regarding the training process

Which classification model performed best, and why?

- The differnce in accuracy between Logistic Regression (0.983) and KNN (0.978) is very small. I can not make clear conclusion which of this models is the best.



Here we can see our data reduced to 2D with PCA. Even reduced data is linearly separable. That means that such task is suitable for Logisitc Regression. This data is also grouped very well that is why KNN performs very well either.

Linear Regression performed the best with l1 regularization, without PCA usage.
KNN performed best with cosine similarity, n_neghbors=10 without PCA usage.

What were the most critical features with regards to the classification, and why?

I standardized the data before feeding it to the model, that why Linear Regression coeffisients are interpretable. The most critical feature is var1 column. Absolute value of coefficient for var1 is at least 8 times bigger than any other coefficient. var7, var4 and some one-hot encoded columns also noticeably affect the prediction.

Some one-hot encoded labels has zero coefficients, some of them are comparable with var7 column coefficient. This means that var3 column information is valuable and we have zero coefficients because of lack of data: big number of different locations and small number of rows in the table.

What features might be redundant or are not useful, and why?

var2 corresponding coefficient is zero. This column do not affect to the prediction at all. Coefficients of var5 and var6 are very close to zero in comparison to columns from the previous question.

var2 is redundant and there is a big possibility to be redundant for var5 and var6 either.

Did the dimensionality reduction by the PCA improve the model performance, and why?

Both Logistic Regression and KNN showed best results on data without dimensionality reduction.

Logistic Regression has comparatively big coefficients for specific locations from var3 column. Columns of such locations are needed for prediction. I think informaton about this locations is too damaged after PCA applying. One-hot encoded columns are very sparse that is why they have little variance and thier information almost reduced with PCA.

As for KNN, it may be that rows with the same location in var3 column are likely to have the same target value. PCA reduction deletes such information.

Naive Bayes showed best result (accuracy = 0.94) with reducing to 6 companents. Accuracy without reduction was about 0.6, nearly to be random. Naive Bayes is conditional probability model. As we saw in the critical features section we have lack of data for some of locations in var3 column. Naive Byes may not learn probabilities var3 column properly and works bad if this var3 column is present (it’s one-hot encoded version). We can notice that best result is obtained with reduction to 6. 6 is number of feature columns without var3 column.