Report

Problem: Create an ML algorithm for predicting the prices of the cars given the car's characteristics

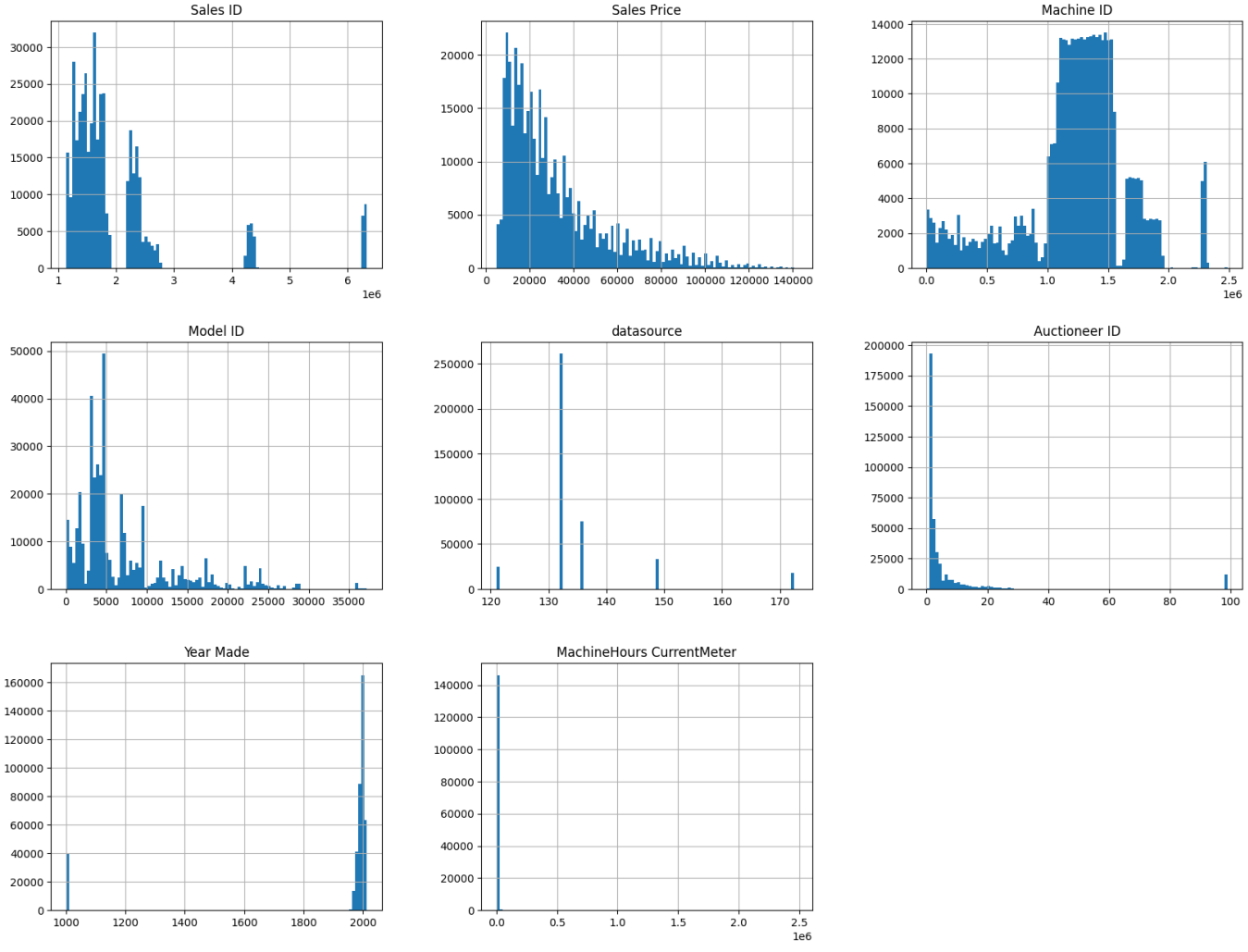
Problem specifics:
- Supervised task (the label are given in the dataset)
- Regression task (more precisely: multiple regression, univariate regression)
- Batch learning

Performance measure: Root Mean Square Error

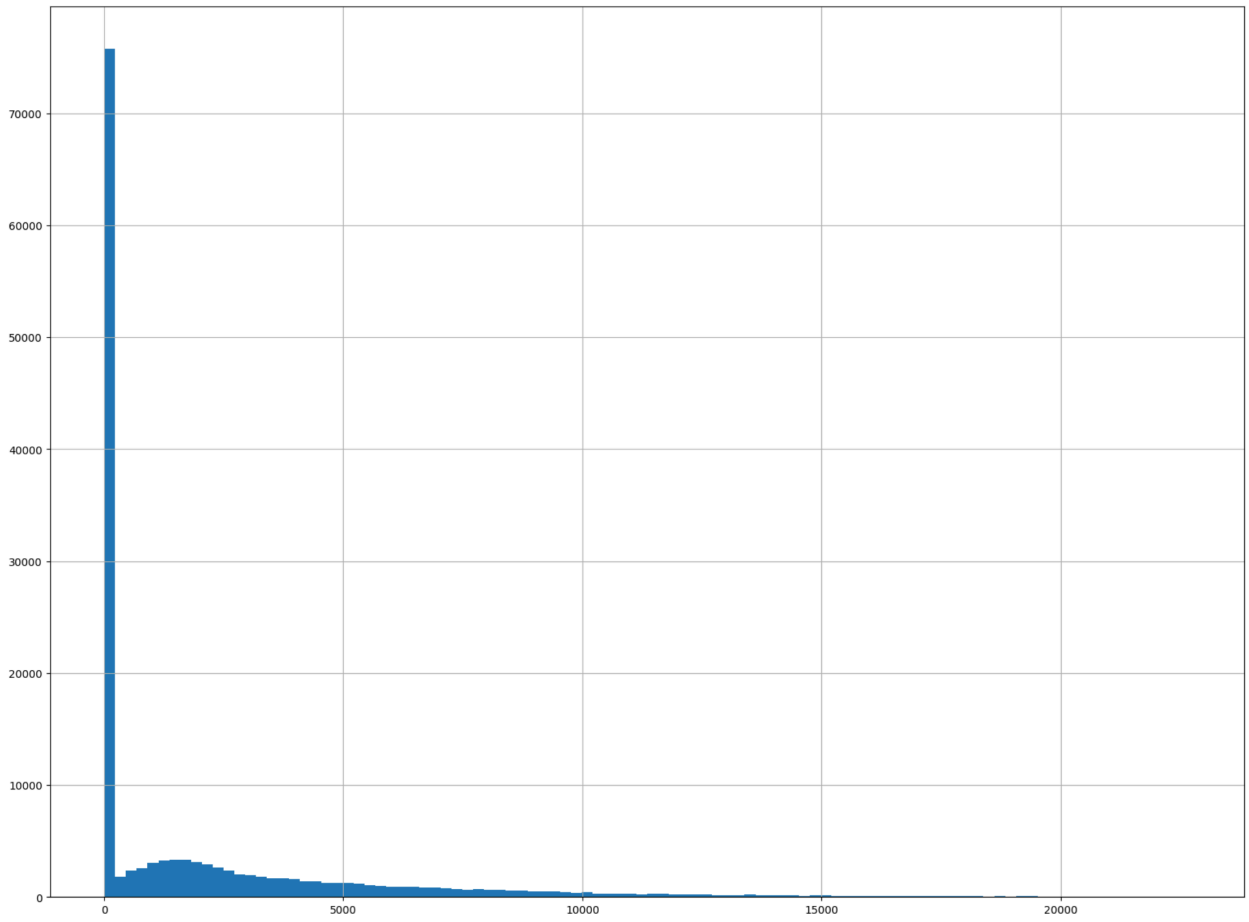
Visualisation

While discovering the dataset given, it's easy to spot that there are a lot of categorical attributes and only a few numerical ones. You may find the list of all attributes in the appendix1.pdf (pages 1-2).

There is the visualization of the numerical attributes (x-axis - value itself, y-axis - number of instances):



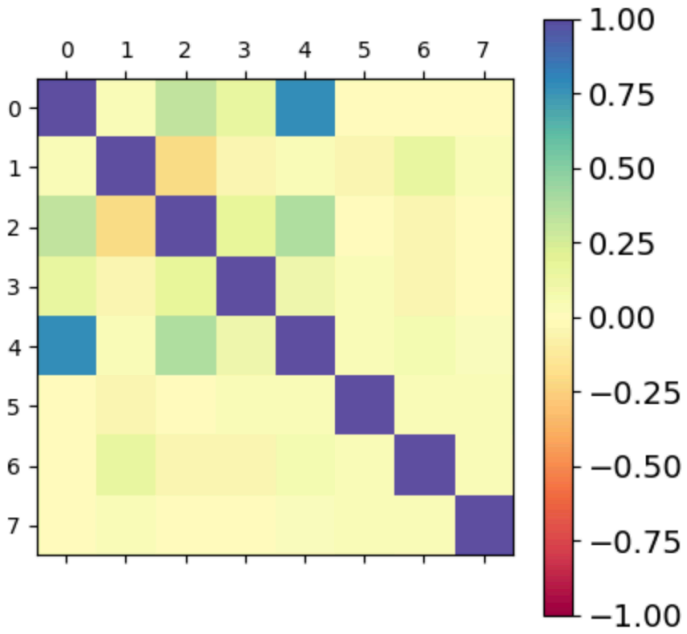
‘MacineHours CurrentMeter’ attribute has a weird histogram because its max value is much bigger than its mean value. Here is the histogram without such a gap:



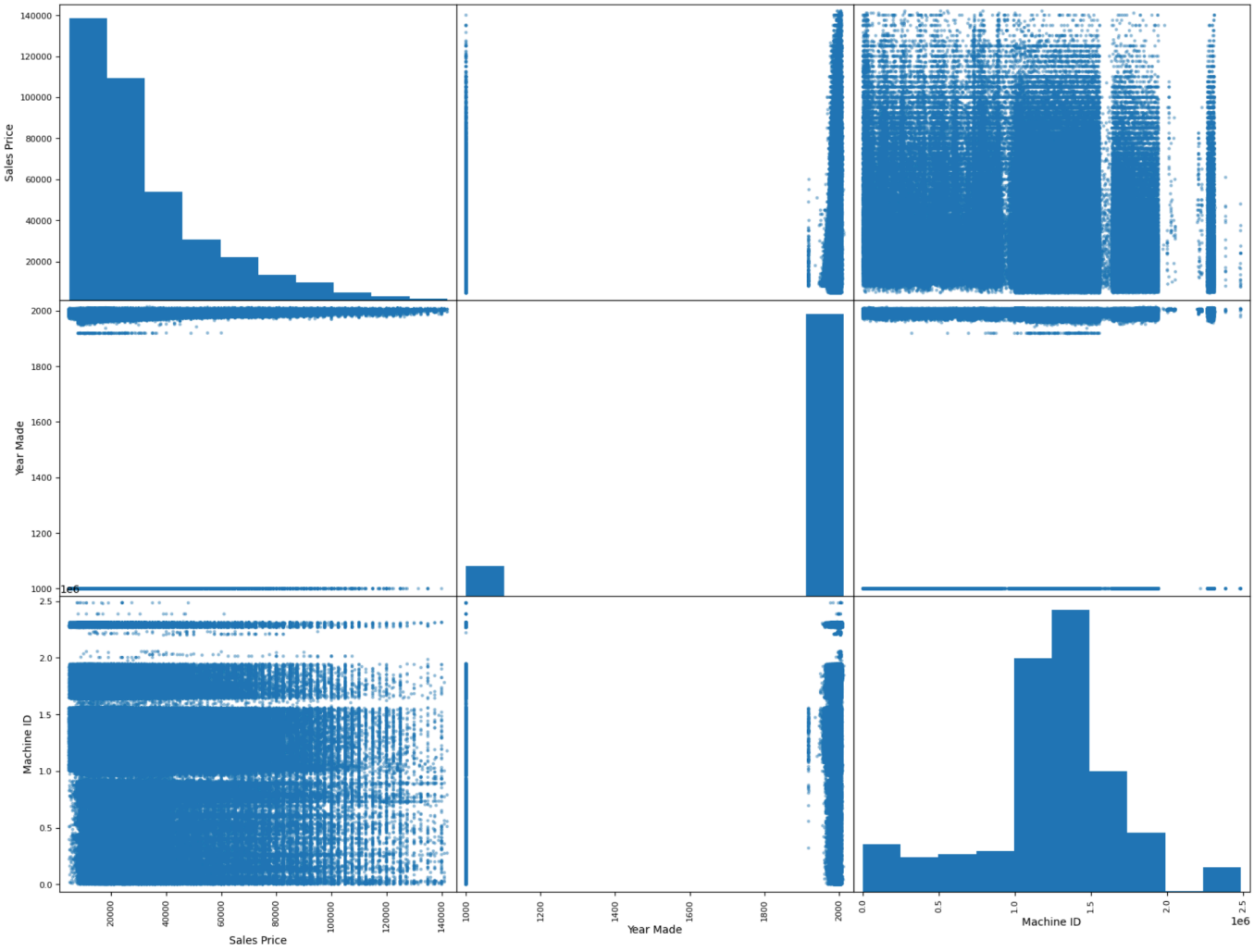
Also, we have a correlation matrix for the numerical attributes:

	Sales ID	Sales Price	Machine ID	Model ID	datasource	Auctioneer ID	Year Made	Machine Hours Current Meter
Sales ID	1.000000	0.021225	0.310083	0.147657	0.769096	-0.00336	0.004409	0.007511
Sales Price	0.021225	1.000000	-0.218688	-0.039675	0.021114	-0.04810	0.155058	0.018688
Machine ID	0.310083	-0.218688	1.000000	0.164715	0.374642	0.006563	-0.053225	-0.002421
Model ID	0.147657	-0.039675	0.164715	1.000000	0.096362	-0.01788	-0.046277	-0.005416
datasource	0.769096	0.021114	0.374642	0.096362	1.000000	-0.01870	0.069903	0.010034
Auctioneer ID	-0.003361	-0.048102	0.006563	-0.017888	-0.018706	1.000000	-0.020317	0.032633
Year Made	0.004409	0.155058	-0.053225	-0.046277	0.069903	-0.02031	1.000000	-0.009124
MachineHours CurrentMeter	0.007511	0.018688	-0.002421	-0.005416	0.010034	0.032633	-0.009124	1.000000

In colour, it looks like this:



From this, it is obvious that only *'Machine ID'* and *'Year Made'* have some useful correlation for us. That's why we used only these two parameters for the ML algorithm. And for them the scatter matrix presented:



The correlation is barely seen, but there are the highest numbers we have. So let's keep it.

As for the categorical attributes - all histograms you may find in the appendix1.pdf. I only want to pay your attention that we have a few attributes, in which we have thousands of classes. For such instances, we used frequency encoding to create data for the ML algorithm

Model Selection and Tuning

As for the model types, the following algorithms were considered: Linear Regression, Decision Tree Regression, Random Forest Regression, Ridge Regression, Lasso Regression, and ElasticNet Regression.

The Linear Regression, Ridge Regression, Decision Tree, and Lasso Regression showed pretty much the same result with a mean RMSE score of around 14,100. The worst one was the Elastic Net with a score of 17,977, meanwhile, Forest Regression turned out to be the best model for this problem: 11,954 with the standard settings of scikit-learn. That is why it was chosen for the fine-tuning. More information about the results during the model testing you may find in the *results.rtf* file.

During the tuning, the result of the Forest Regression model decreased up to 10,972. Important to note, that the result occurred in the case of the highest values of the parameters checked. That is why the performance of the model may rise with the parameters' values increasing. However, further exploration was not conducted due to the lack of computational power. I stopped on a sufficient result for myself.

Results

Using the Forest Regression model we received the following result:

RMSE = 10616.4
with [10516.6, 10715.2] 95% confidence interval