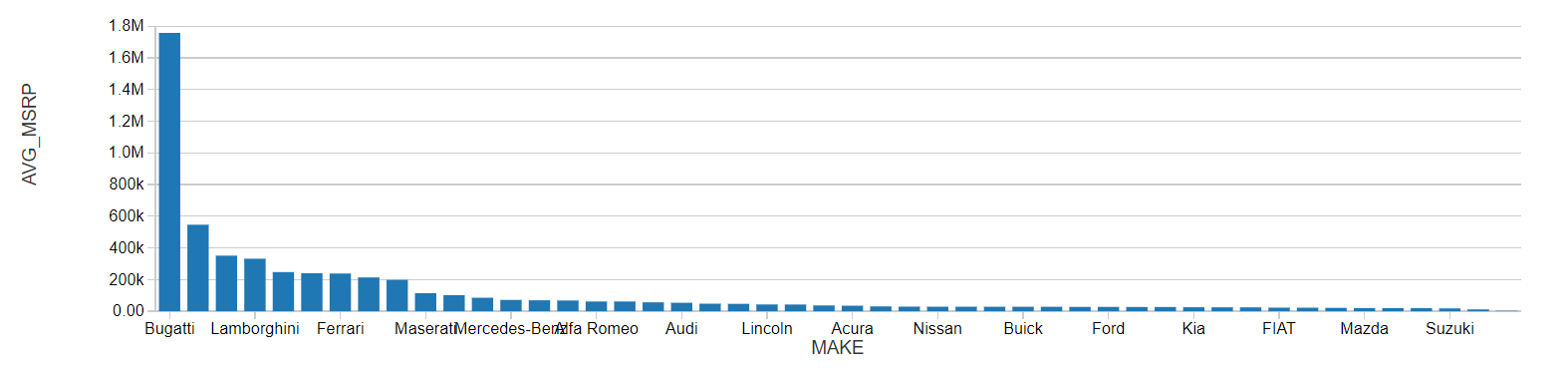
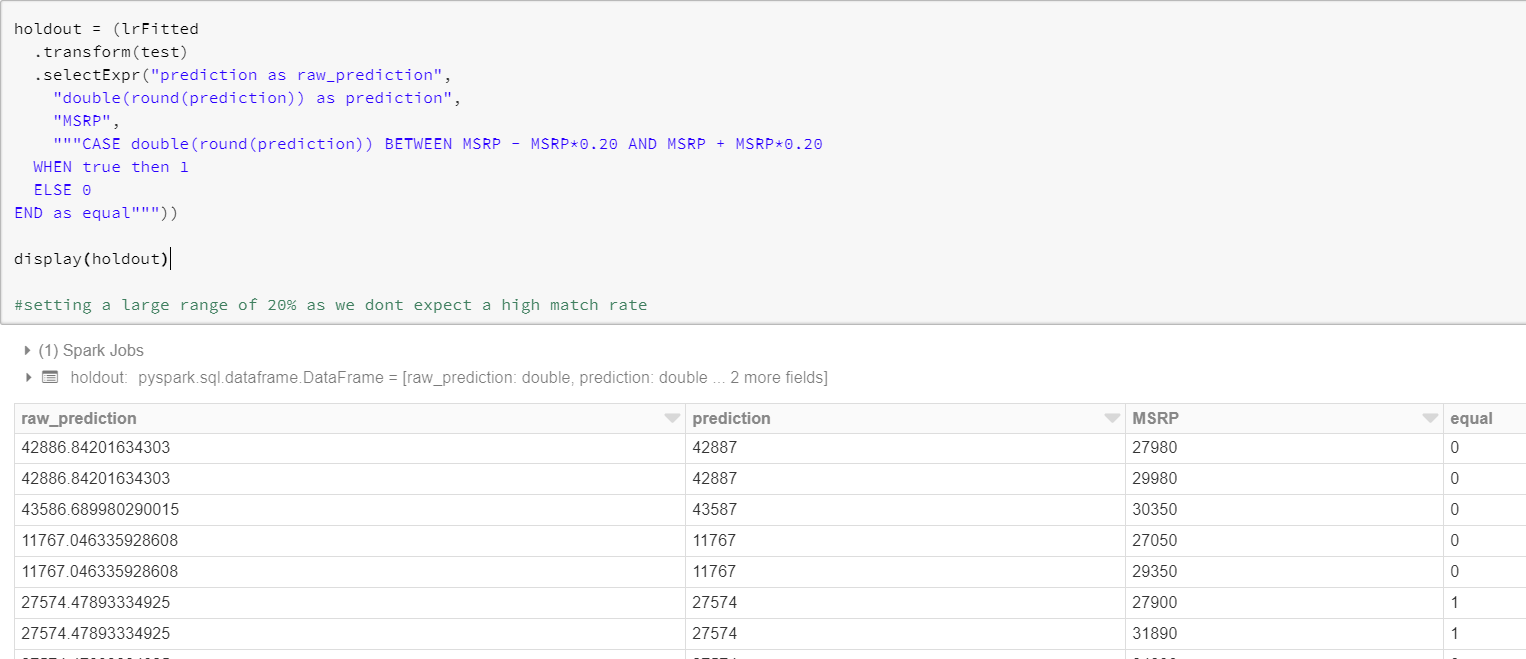
**Linear Regression**

Going through analysis of the data set, we can see that we have both numerical values and categorical values. We created two different models one with using only the numerical values other with converting the categorical into numerical using StringIndexer, and VectorAssembler.

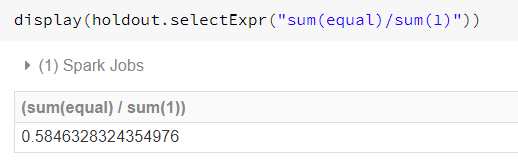
Starting with linear regression, we set the feature columns to be just the available numerical columns. Which were: 'Year', 'Engine HP', 'Engine Cylinders', 'Number of Doors', 'highway MPG', 'city mpg', 'Popularity'. Since we know that the Make of the vehicle has a huge impact on MSRP, we are expecting a bad model, and since they are categorical pieces of data we are not including them at this point. Point in case we know that Bugatti has an average MSRP of about 1.7M, Maybach with average of 546k, and in the low point we have Plymouth at 3k. So, it is not hard to see that the linear regression without usage of the categorical values will give us a bad model. See below for graph.



After running through the model, we set ranges against the predicted value and the actual MSRP, since getting a MSRP to be exact will be very difficult. We tried to use a 20% range, If the predicted value fell in between MSRP – MSRP\*0.2 and MSRP + MSRP\*0.2 then the model was successful in prediction. See below for code.Even with a large range such a 20% of the MSRP we still get a model which is only accurate for about 23%.

Since this model was not very useful we decided to try using the stringIndexer to convert categorical data to usable points of data in a linear model. The full set of columns we are using now are: Make, Model, Year, Engine Fuel Type, Engine HP, Engine Cylinders, Transmission Type, Driven\_Wheels, Number of Doors, Market Category, Vehicle Size, Vehicle Style, highway MPG, city mpg, Popularity. We expect a much better model this time as we are including all columns provided. We know the biggest contributors to the model should be Model, Make, Engine HP, which are all now included.

Again, running through the same steps, we fit the data and tested against the 20% range of MSRP. We saw a big improvement. We see our model is now about 82% accurate in the predictions of MSRP using all available columns. Since we know 20% range is not useful for us in use in real life. Setting the range to be 10% is a little more feasible, IE. MSRP Between MSRP - MSRP\*0.10 AND MSRP + MSRP\*0.10. After running the below. We can see the model accuracy has decreased to about 58% which is still better than what we had before.



Dropping the range to only %5 which is more feasible IE MSRP between MSRP - MSRP\*0.05 AND MSRP + MSRP\*0.05, we see a huge drop in accuracy, we get about 34%. If we want to improve the model we can try removing columns which may not be as useful for us, like cylinders if we already have HP it may not be needed, city MPG probably does not impact price as much. We will try other modeling methods to improve model accuracy.

