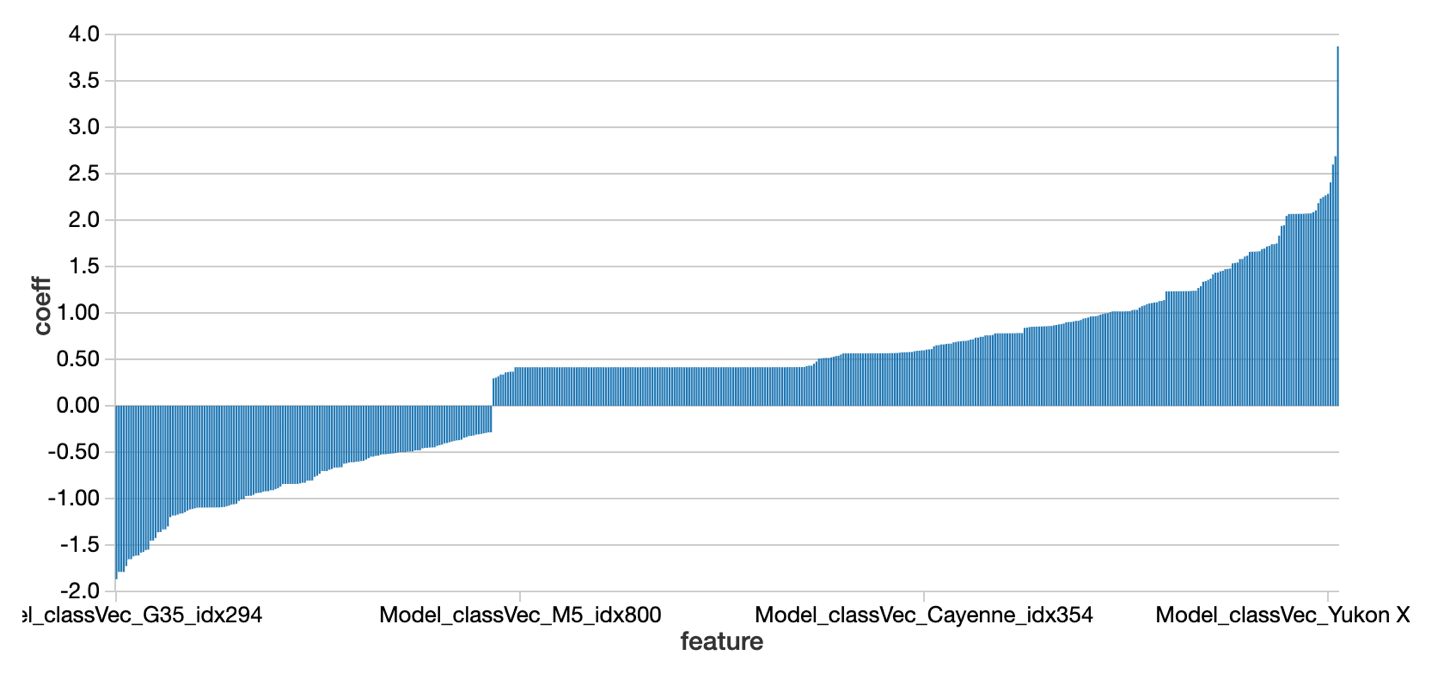
**Support Vector Machine Classification**

Support Vector Machines (SVM) can be used for a variety of machine learning tasks. In this case, a linear classifier (LinearSCV from the pyspark machine learning classification library) was chosen to predict the price class of a vehicle given the input features in the dataset. This linear classifier attempts to draw a hyperplane through the dimensions of features in the set, which will maximize the division in the data (value vs. expensive). The mean of the MSRP was chosen as the classification point. It should be noted that the distribution of the MSRP was not normal; therefore, the mode of the MSRP was also used as the classification point for comparison, but it is not included in this report as the difference was negligible.

Initial tuning of the classifier with all features (1000+) enabled resulted in an ROC AUC of 0.99 and an F1 score of 0.89. The fitting of this model took approximately 4.5 minutes. Unfortunately, automated feature selection was not possible with the LinearSVC library, as it is an experimental library in pyspark and is not as rich in features as other, more mature, libraries. In addition, the evaluator for the LinearSVC model only outputs the area under the ROC; therefore, a confusion matrix and F1 score has to be calculated manually in order to properly asses the fit, especially given that the output variable, MSRP, was not normal.

In a linear SVM classifier, the weight of each coefficient from the resulting model can be used to determine the importance of the feature relative to the other features. Higher negative coefficients have more weight to classify the vehicle as value, and higher positive coefficients have more weight to classify the vehicle as expensive. In order to determine the most important features, the largest 500 coefficients (absolute value) were extracted and plotted.

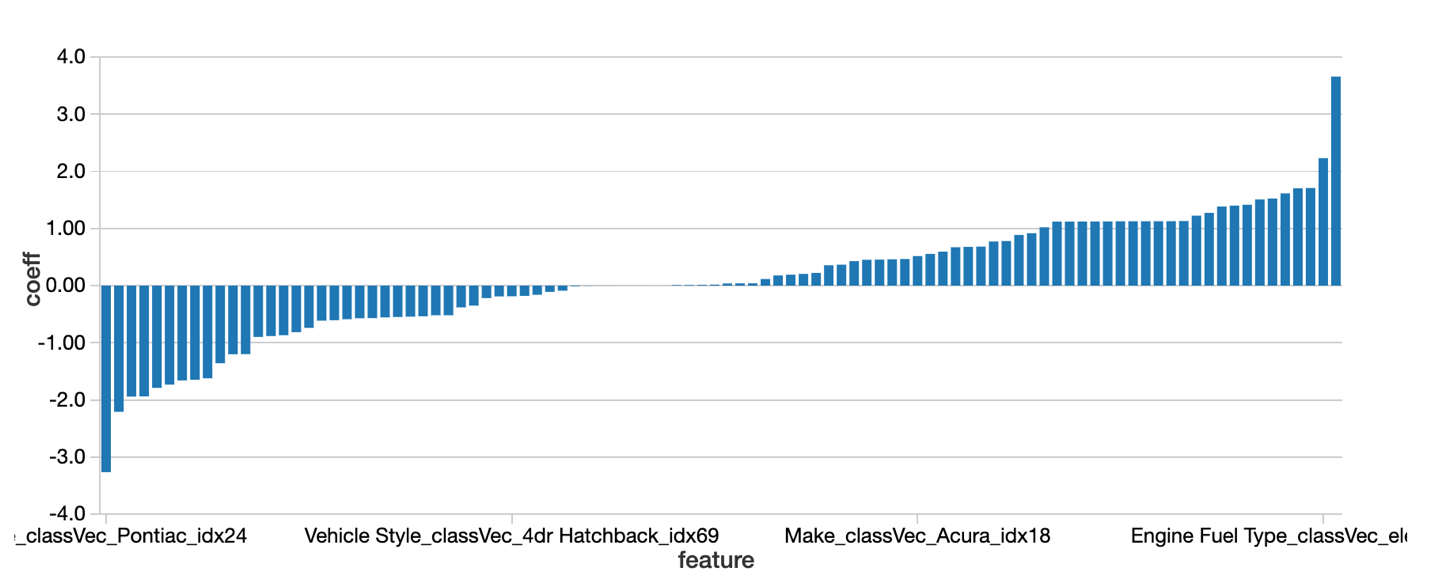
Top 500 Features from Initial SVM Fit



The above showed that the Vehicle Model feature for each vehicle was the most influential in determining the class. The pyspark feature library function, VectorSlicer, was used to select a subset of these most important features from the original feature list and run the fit again. The reduced feature set fit resulted in an ROC AUC of 0.98 and an F1 score of 0.89; however, the time to run the model, approximately 5 minutes, took longer than the original fit.

Since Vehicle Model was so prominent in the top features, a fit was attempted with this feature excluded from the inputs. This fit resulted in an ROC AUC of 0.98 and an F1 score of 0.89, and took approximately 2.5 minutes to run. The same methodology was then used to extract the top 150 features from this fit.

Top 150 Features from SVM Fit without Vehicle Model



With Vehicle Model removed, Vehicle Make was the most common type of feature that was influencing the classifier. Some of the other notable features that were important in classifying the vehicle as expensive were the following: Engine Fuel Type and Market Category of Exotic, High-Performance, or Luxury. Other important features to classify the vehicle as value were the following: Vehicle Style such as Midsize or Compact and Market Category of Hatchback.

A final fit was performed using the top 150 features (which already excluded the Vehicle Model features) above. The result was an ROC AUC of 0.98 and an F1 score of 0.88 and took approximately 2.6 minutes.

As can be seen, the Support Vector Machine classifier is a robust and versatile model that performs well with large or small features sets. Its only downside when used in pyspark is that it is not as feature rich as most other modelling tools and requires manual intervention to evaluate models and perform feature selection.