Notes:

* SVM: finds optimal hyperplane
* Optimizes margin / decision boundary bw two classes
* Hyperplane in 2 d is just a line, in 3-dim is a plane, in n-dim space, is an n-dim plane
* Kernel trick, projecting data to obtain a better decision boundary
* SVM, preferrable for classification
* Excels with high feature few rows data
* Untangle complex relationships, deals with outliers too
* Not good for lot of rows, few features.
* Not good for transparency, or predictor significance
* If constrained for time, or compute power, may not want to use
* Slow to train
* Kernel parameter and C hyperparameter (regularization parameter) are the most important
* High C, low penalty
* C: how willing the algorithm is to misclassify points in an effort to capture the general pattern of the data
* C determines the difference between overfitting and underfitting

Ok, in order to get the algorithm to run, I’ve selected only a sample of the dataframe for training the algorithm.

About 10,000 rows out of 38,000. And even this still takes 5 minutes to run.

Now, the classifier is still erroneously classifying all points as 0’s. Will need to look into this.

* Hyperparameter tuning: looking for the right kernel, and for the right C