# Introduction:

For this exercise I experimented with different preprocessing methods and their effect on various ML models as well as with and without hyperparameter optimization. I went with numerical preprocessors because every categorical preprocessor I tried had extremely long runtimes that I would just cancel after 15+ minutes of running.

# Dataset Description:

I used the wine quality dataset[1] for this and picked the white wine dataset because it has more entries than the red dataset. This data includes 11 features and a target value called “quality” that is between 0 and 10.

[1]: <https://archive-beta.ics.uci.edu/dataset/186/wine+quality>

# Experimental Setup:

For these experiments I picked a few models to test on, Random Forest because it has been the most accurate in every test on the wine dataset, K Nearest Neighbors because I wanted to how much preprocessing would improve it, and Support Vector Machine because I wanted a third model, and it was used in all the examples on scikit-learn’s documentation.

Then I picked a couple preprocessing methods, I decided to use two different scaler methods because I thought those would have the most impact and I used the basic scaler and a minmax scaler.

Then I ran some tests on all the models with default parameters and optimized parameters with and without preprocessing and compared the results.

I then output the results sorting by accuracy first and time second then labeling the best one.

# Results:

A screenshot of a computer screen

Description automatically generated

The results show some improvements for random forest with preprocessed data and huge improvements for the support vector and k nearest neighbors algorithms when the data is preprocessed.

# Code:

Main.py