Gabe Wilson

CS 4641

11/29/19

Final Project Free Response

**Introduction**

Data science, machine learning, and artificial intelligence are all buzzwords that have piqued the interest of statisticians, analysts, and scientists alike. However, these topics that form the intersection of computer science and statistics are accessible to more than your Jane Street quant. For this very reason, I sought to build a kernel that would be useful to the general populace by taking vast amounts of quantitative information and boiling it down into an evidence-based prediction. In order to achieve this goal, I first needed to choose a relevant dataset. I chose a Kaggle csv file that stored the features of more than 240,000 songs from Spotify. This data included features such as genre, acousticness, instrumentallness, loudness, danceability etc. Danceability was the feature that stood out the most to me, so I wrote three algorithms that sought to predict the danceability of songs from the dataset. Given that danceability was recorded in the csv as a decimal value between 0 and 1, I implemented the regression algorithms that facilitated making a non-discrete prediction. The success of this kernel has real world applications that include recommending songs for dance parties and playlists and determining what features of a piece factor into making it a danceable song. To measure performance, I will use scikit-learns “score” method which returns the coefficient R^2. This coefficient is given by 1 - u/v, where u is the residual sum of squares ((y\_true - y\_pred) \*\* 2).sum() and v is the total sum of squares ((y\_true - y\_true.mean()) \*\* 2).sum().

**Description of the Algorithm**

Briefly describe each of the algorithms you will be testing. Identify the hyperparameters that need to be tuned in the next section and explain how they are related to the complexity of the hypothesis class.

1. *K Neighbors Regression*

The first algorithm I implemented to predict the danceability was K Neighbors Regression. Unlike K Neighbors Classification that aggregates the classes of the k nearest neighbors to classify a query point, K Neighbors Regression uses supervised learning to make a prediction for features with continuous values. In our case, this feature was the danceability. In order to understand the functionality of this algorithm let us simplify our dataset by graphing it on a 2d cartesian coordinate system. This reduction of the dataset calls for us to simplify each data point to only have 2 features for example, energy and instrumentalness We take in k as a parameter that indicates how many points adjacent to the query point should be factored in to determine a value for our hypothesis class.

(add graph here)

1. *Kernel Ridge Regression*

The first algorithm I implemented to predict the danceability was K Neighbors Regression.

1. *Neural Network*

**Tuning hyperparameters**