Extra Credit Assignment: Kernel-based Naive Bayes Classification

Harsh Chaturvedi, Pratik Anand

Training Data

The training data consists of **7352 observations**. Each observation has **561 input features** and a **single target variable** belonging to one of **6 classes**. As seen in the figure below, the distribution of observations among target classes is quite even i.e. **the distribution is not skewed towards any subset of classes**. This allows us to use **classification accuracy** as a reliable metric of model performance.

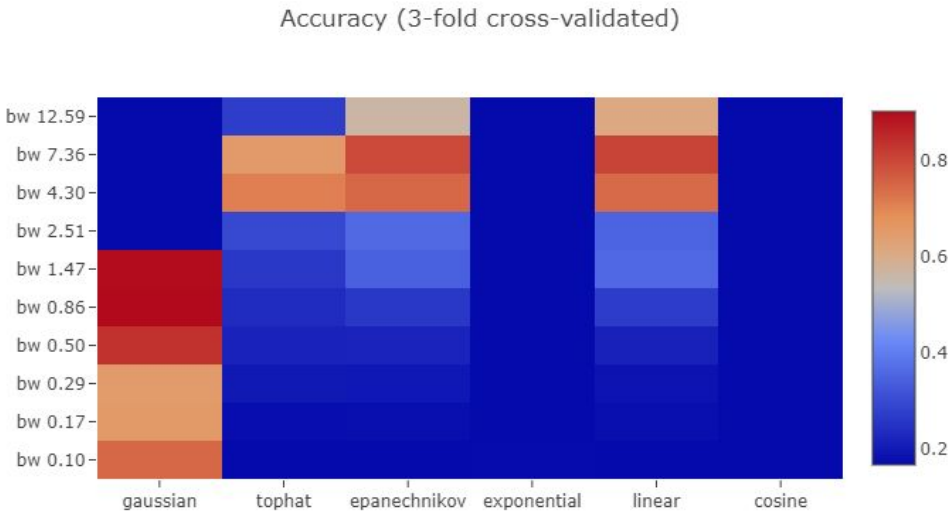
The training data (as well as test data) were imported into memory for manipulation as Dataframe objects as implemented in the Pandas Python library.

Model Implementation & Tuning

We implemented a **kernel-based Naive Bayes classifier** using the KernelDensity, BaseEstimator and ClassificationMixin functions from the Sickit-Learn machine learning library for Python.

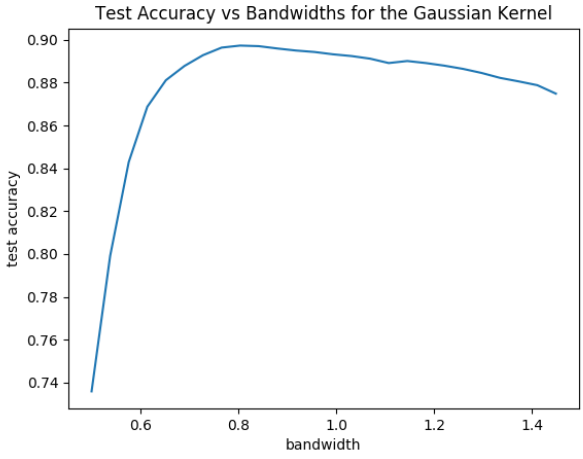
Before tuning the parameters for the model, we tried standardizing the features. However, this resulted in poorer model-performance compared with un-standardized features. We therefore decided to **forego feature scaling** in our final model

In order to **rapidly test** several hyperparameter combinations, we initially worked with a **10% stratified random sample** of the full training dataset. We performed **3-fold cross validation** for several kernels viz. **gaussian, tophat, epanechnikov, exponential, linear** and **cosine** over a range of bandwidths using Sickit-Learn's GridSearchCV function, the results of which are summarized in the adjoining heatmap.



The **gaussian kernel outperforms** all others with a test accuracy of 0.9 when used with a smoothing bandwidth of 0.86.

After having identified the best kernel and the broad bandwidth range for optimal model performance, we proceeded to do a more **fine-grained grid-search** over the relevant bandwidth range, this time **utilizing the full training dataset**, results of which are summarized in the adjoining line-plot. The **optimal bandwidth found was 0.858** which produced a cross-validated **test accuracy of 0.897**.



Predictions

Having tuned our model, we **applied it to our test data** (consisting of 2947 observations), and generated **our predictions** for the same: https://goo.gl/BQyH6b.

Remarks: In contrast to most other classification algorithms, the kernel-based Naive Bayes classifier is fast to train, but slow to predict outputs for new inputs. This is likely because the underlying PDFs are cumulatively computed at training time, but class probabilities for each observation are individually computed at time of prediction.