

## HW2 Implementation Code

ECE: 523  
David Akre

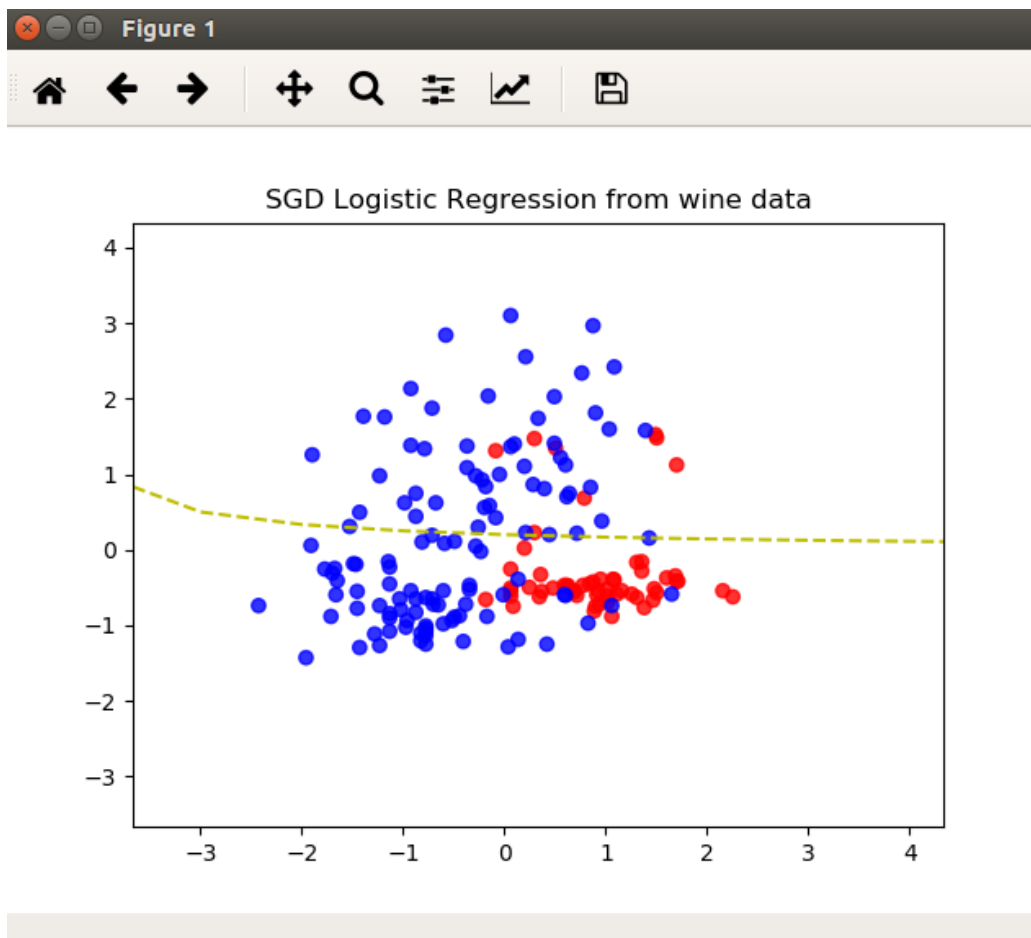
### Logistic Regression utilizing SGD Algorithm

#### Process:

- Matplotlib will generate a graph with the logistic regression to linearly separate the two classes generated by data pulled from the 523 github (i.e. wine.csv)
- The logistic regression algorithm employs the SGD algorithm to minimize the function on the data and find the maximum margin between two classes of data points. The SGD algorithm will return a classifier used to help fit the linear model to the data at the end. Once the model is fit to the classifier value, I plot the scatter plot from the wine.csv to the general linear model in the form  $(1 / (1 + e^{wt \cdot x}))$

#### Results:

Results captured in sgd\_logreg\_wine.png in the results directory.



### Dimensionality Reduction

#### Process:

- The code will read in a dataset provided the ECE523 repository which will be in the format of set of csv files.

- Iteratively, once a file is read into program memory the first part will apply the naive bayes classifier and assess its overall accuracy based on the data and output.
- The second part will then apply PCA (i.e. principal component analysis) on the data as a preprocessor to reduce the dimensions down according to the MLE protocol in sklearn (e.g. I set the n\_components value to 0.5 which will allow for sklearn to attempt to reduce the dimensionality to an ideal setting for classification).
- After the preprocessing finishes, I apply the naive bayes classification algorithm provided by sklearn followed by assessing the overall accuracy of the fit.
- In the final step I create a data frame with a dictionary of the results with the key value being the data file read in (i.e. the csv file). After which I write the data frame to a results.csv file which is a 10x2 table format.

## Results:

Results captured in results.csv in the results directory.

	Naive Bayes Classifier	Naive Bayes Classifier with PCA preprocessing
lung-cancer.csv	0.87097	0.83871
magic.csv	0.72685	0.71712
wine.csv	0.9887	0.9661
lymphography.csv	0.85034	0.85714
car.csv	0.6989	0.71743
wine-quality-white.csv	0.44864	0.47989
yeast.csv	0.13621	0.51585
wine-quality-red.csv	0.56133	0.56446
zoo.csv	1	0.87
waveform.csv	0.80916	0.82977

## Density Estimation

### Process:

- This code will generate a checkerboard of data from two classes and employ a density estimator (i.e. kNN) to classify the data. After the classification the code will plot the probability of 'x given y' or  $P(X|Y)$  using the density estimator

### Results:

Results captured in knn\_checkerboard.png in the results directory.

