ECE 156B
HW # 1

Vicki Chen

Classification Methods

These methods below scored the highest in accuracy on the test data after splitting the original dataset into test & train data.

K nearest neighbor

The K-nearest neighbor algorithm essentially forms a majority vote between the K most similar instances to a given unseen observation. Similarity is defined according to a distance metric between two data points. It uses Euclidean distance given by

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \ldots + (x_n - x'_n)^2}$$

given a positive integer **K**, an unseen observation **x** and a similarity metric **d**, KNN classifier performs the following two steps:

1. Runs through the whole dataset computing **d** between **x** and each training observation. We'll call the **K** points in the training data that are closest to **x** the set **A**
2. It then estimates the conditional probability for each class, that is, the fraction of points in **A** with that given class label

Finally, input x gets assigned to the class (in our case, the Pokémon) with the largest probability.

$$P(y = j|X = x) = \frac{1}{K} \sum_{i \in \mathcal{A}} I(y^{(i)} = j)$$

KNN searches the memorized training observations for the K instances that most closely resemble the new instance and assigns to it the their most common class.

LDA (linear discriminant analysis)

LDA makes predictions by estimating the probability that a new set of inputs belongs to each class (Pokémon). The class that gets the highest probability is the output class and a prediction is made.

The model uses Bayes Theorem to estimate the probabilities. Briefly Bayes' Theorem can be used to estimate the probability of the output class **k** given the input **x** using the probability of each class and the probability of the data belonging to each class:

$$\hat{P}(Y = k \mid X = x) = \frac{\hat{P}(X = x \mid Y = k)\hat{P}(Y = k)}{\sum_j \hat{P}(X = x \mid Y = j)\hat{P}(Y = j)}$$

Naïve Bayes

Naive Bayes classifier is a probabilistic model based on the Bayes theorem.

Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.
The variable **y** is the class, which represents the Pokémon. Variable **X** represent the features.

X is given as,

$$X = (x_1, x_2, x_3, ..., x_n)$$

Here x1,x2…. represent the features, i.e. height, weight, ATK. By substituting for **X** and expanding using the chain rule we get:

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Now, obtain the values for each by looking at the dataset and substitute them into the equation. For all entries in the dataset, the denominator does not change. Therefore, the denominator can be removed and a proportionality can be introduced.

$$P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

Last, find the class **y** with maximum probability.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

SVM (BEST MODEL)

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space N (features) that classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Classification Comparison

After getting the results from different classification methods, we calculated the accuracy of each method and plotted a comparison graph:
Accuracy of Logistic regression classifier on training set: 0.79
Accuracy of Logistic regression classifier on test set: 0.69
Accuracy of Decision Tree classifier on training set: 1.00
Accuracy of Decision Tree classifier on test set: 0.77
Accuracy of K-NN classifier on training set: 1.00
Accuracy of K-NN classifier on test set: 0.91

Accuracy of LDA classifier on training set: 0.98
Accuracy of LDA classifier on test set: 0.98
Accuracy of GNB classifier on training set: 0.83
Accuracy of GNB classifier on test set: 0.80
Accuracy of SVM classifier on training set: 1.00
Accuracy of SVM classifier on test set: 0.98

From these results, we can see that **SVM** has the highest accuracy. Once I confirmed which algorithm has the highest accuracy, I used this method to predict the Pokémon based on the data from the unknown set.

Results

The results of the predicted Pokémon based on the unlabeled data set according to **SVM** is:

['Charmander' 'Charmander' 'Bulbasaur' 'Bulbasaur' 'Jynx' 'Jynx' 'Jynx'
'Jigglypuff' 'Bulbasaur' 'Jynx' 'Pidgey' 'Jynx' 'Pidgey' 'Charmander'
'Charmander' 'Squirtle' 'Pikachu' 'Jynx' 'Pikachu' 'Jynx' 'Charmander'
'Jynx' 'Squirtle' 'Jynx' 'Pidgey' 'Bulbasaur' 'Pikachu' 'Shedinja'
'Pidgey' 'Squirtle' 'Bulbasaur' 'Shedinja' 'Pikachu' 'Charmander'
'Squirtle' 'Squirtle' 'Pidgey' 'Charmander' 'Shedinja' 'Pidgey'
'Shedinja' 'Squirtle' 'Charmander' 'Charmander' 'Jynx' 'Shedinja'
'Shedinja' 'Charmander' 'Charmander' 'Shedinja' 'Shedinja' 'Charmander'
'Squirtle' 'Jynx' 'Jigglypuff' 'Bulbasaur' 'Pikachu' 'Pidgey' 'Shedinja'
'Jigglypuff' 'Pikachu' 'Jynx' 'Bulbasaur' 'Bulbasaur' 'Jynx' 'Pikachu'
'Pikachu' 'Bulbasaur' 'Bulbasaur' 'Shedinja' 'Pikachu' 'Jigglypuff'
'Bulbasaur' 'Bulbasaur' 'Squirtle' 'Bulbasaur' 'Bulbasaur' 'Jigglypuff'
'Jigglypuff' 'Shedinja' 'Shedinja' 'Jigglypuff' 'Squirtle' 'Charmander'
'Shedinja' 'Squirtle' 'Jigglypuff' 'Jynx' 'Shedinja' 'Jigglypuff'
'Charmander' 'Pidgey' 'Pikachu' 'Charmander' 'Squirtle' 'Shedinja'
'Squirtle' 'Pikachu' 'Pikachu' 'Pidgey' 'Squirtle' 'Jynx' 'Charmander'
'Charmander' 'Charmander' 'Bulbasaur' 'Pidgey' 'Bulbasaur' 'Squirtle'
'Jigglypuff' 'Shedinja' 'Pidgey' 'Pidgey' 'Squirtle' 'Pikachu' 'Pikachu'
'Squirtle' 'Pikachu' 'Jigglypuff' 'Jigglypuff' 'Jynx' 'Charmander'
'Pikachu' 'Squirtle' 'Shedinja' 'Jigglypuff' 'Jigglypuff' 'Pikachu'
'Jigglypuff' 'Pikachu' 'Bulbasaur' 'Shedinja' 'Shedinja' 'Pikachu' 'Jynx'
'Pikachu' 'Shedinja' 'Shedinja' 'Bulbasaur' 'Shedinja' 'Jigglypuff'
'Pikachu' 'Jynx' 'Pidgey' 'Shedinja' 'Bulbasaur' 'Pidgey' 'Bulbasaur'
'Jynx' 'Charmander' 'Jigglypuff' 'Jigglypuff' 'Pidgey' 'Squirtle' 'Jynx'
'Shedinja' 'Pidgey' 'Pikachu' 'Squirtle' 'Jynx' 'Bulbasaur' 'Pidgey'
'Shedinja' 'Jynx' 'Shedinja' 'Jigglypuff' 'Pikachu' 'Jigglypuff' 'Pidgey'
'Shedinja' 'Jynx' 'Charmander' 'Bulbasaur' 'Pidgey' 'Bulbasaur'
'Bulbasaur' 'Shedinja' 'Squirtle' 'Pikachu' 'Charmander' 'Charmander'
'Shedinja' 'Bulbasaur' 'Squirtle' 'Pikachu' 'Pidgey' 'Bulbasaur' 'Pidgey'

'Pikachu' 'Jigglypuff' 'Bulbasaur' 'Jynx' 'Pikachu' 'Charmander'
'Squirtle' 'Shedinja' 'Jigglypuff' 'Pikachu' 'Pidgey' 'Jynx' 'Bulbasaur'
'Jynx' 'Jynx' 'Jynx' 'Bulbasaur' 'Charmander' 'Pikachu' 'Jynx'
'Charmander' 'Bulbasaur' 'Pidgey' 'Jigglypuff' 'Jigglypuff' 'Pikachu'
'Pikachu' 'Pidgey' 'Squirtle' 'Squirtle' 'Charmander' 'Jynx' 'Charmander'
'Charmander' 'Jynx' 'Bulbasaur' 'Pikachu' 'Pidgey' 'Jynx' 'Bulbasaur'
'Pikachu' 'Jigglypuff' 'Squirtle' 'Jigglypuff' 'Jynx' 'Bulbasaur'
'Squirtle' 'Jigglypuff' 'Squirtle' 'Pidgey' 'Squirtle' 'Charmander'
'Shedinja' 'Jynx' 'Pikachu' 'Pidgey' 'Pikachu' 'Bulbasaur' 'Charmander'
'Pidgey' 'Jynx' 'Squirtle' 'Jigglypuff' 'Pikachu' 'Squirtle' 'Pidgey'
'Pidgey' 'Pikachu' 'Pidgey' 'Charmander' 'Charmander' 'Charmander'
'Charmander' 'Shedinja' 'Charmander' 'Jynx' 'Jigglypuff' 'Pikachu'
'Pidgey' 'Bulbasaur' 'Bulbasaur' 'Jigglypuff' 'Pidgey' 'Pikachu'
'Squirtle' 'Shedinja' 'Jigglypuff' 'Pidgey' 'Bulbasaur' 'Squirtle'
'Squirtle' 'Pidgey' 'Jigglypuff' 'Charmander' 'Pidgey' 'Jigglypuff'
'Pidgey' 'Shedinja' 'Squirtle' 'Jigglypuff' 'Pikachu' 'Jigglypuff' 'Jynx'
'Shedinja' 'Charmander' 'Jigglypuff' 'Pidgey' 'Shedinja' 'Squirtle'
'Shedinja' 'Squirtle' 'Bulbasaur' 'Jynx' 'Charmander' 'Jigglypuff'
'Pikachu' 'Bulbasaur' 'Jynx' 'Jigglypuff' 'Jynx' 'Squirtle' 'Squirtle'
'Jigglypuff' 'Charmander' 'Shedinja' 'Pikachu' 'Pidgey' 'Shedinja'
'Charmander' 'Squirtle' 'Squirtle' 'Shedinja']

The file *comparison.py* attached in the folder runs the top 4 algorithm on the unknown dataset and classifies it. All results from the top 4 algorithms are saved as *npy* files with the algorithm's name as filename.

The file *classifier.py* attached in the folder runs different the classification methods and calculates the accuracy.

Resources used:
https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html?fbclid=IwAR28VyOJB-_ya6qDMJ4lD2l320LVjCMP2AMngAkdR9bhi-3ijUEgjsSWapA#sphx-glr-download-auto-examples-classification-plot-classifier-comparison-py
https://towardsdatascience.com/solving-a-simple-classification-problem-with-python-fruits-lovers-edition-d20ab6b071d2