ENGS 108 Fall 2020 Assignment 3b

Due October 7, 2020 at 11:59PM on Canvas

Instructors: George Cybenko

TAs: Chase Yakaboski

Rules and Requirements

1. You are only allowed to use Python packages that are explicity imported in the assignment notebook or are standard (bultin) python libraries like random, os, sys, etc, (Standard Bultin Python libraries will have a Python.org documentation). For this assignment you may use:
    - numpy
    - pandas
    - scikit-learn
    - matplotlib
2. All code must be fit into the designated code or text blocks in the assignment notebook. They are indentified by a **TODO** qualifier.
3. For analytical questions that don't require code, type your answer cleanly in Markdown. For help, see the Google Colab Markdown Guide.

In [ ]:
```
''' Import Statements '''
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import copy
import pickle
```

Data Loading

Upload the red and synthetic datasets to your google colab session using Google Drive. Read the following tutorial for how to get setup.

In [ ]:
```
dataset_base_path = '/content/sample_data'

#TODO: Set your base datasets path. This is my base path, you will need to change to match yours.
dataset_github_path = '/content/drive/My Drive/git/ENGS_108_Fall_2020/datasets'
```

In [ ]:
```
#-- Load Cyrillic Data
with open(os.path.join(dataset_github_path, 'cyrillic_data.pk'), 'rb') as f_:
  cyrillic, letter_map = pickle.load(f_)
```

Problem 1: Cyrillic Dataset

In this problem, we will put many of the techniques we've learned together to compare different ways to classify handwritten Russia Cyrillic letters. The dataset has already been converted for you into a numpy array with the first column delineating the letter class as an integer value. A letter_map dictionary provides a simple mapping from these integers to the associated cyrillic character. >

> **(a)** Produce a distance matrix between characters using the Hamming distance, where the distance is smallest for characters that are *similar*.

In [ ]:
```
#TODO: Your code goes here.
```

> **(b)** Produce a distance matrix between characters using the Hamming distance, where the distance is smallest for characters that are *most different*.

In [ ]:
```
#TODO: Your code goes here.
```

> **(c)** Build a k-nn classifier for the cyrillic dataset by first shuffling the data and holding out 20% of the data for cross validation. Report your accuracy on this holdout set.

In [ ]:
```
#TODO: Build and train your KNN like we did in HW2.
```

> **(d)** Build a decision tree classifier for the cyrillic dataset using the same training and holdout dataset you used in (c). Report your accuracy on this holdout set.

In [ ]:
```
#TODO: Build and train your Decision Tree like we did in HW2.
```

> **(e)** Build a logistic classifier for the cyrillic dataset using the same training and holdout dataset you used in (c). Report your accuracy on this holdout set.

In [ ]:
```
#TODO: Build and train your logistic classifier like we did in Problem 2.
```

> **(f)** Compare all your different models and report which was best and why you think the best performing model worked better than the others.

TODO: Report on your findings.