CPSC 483 - Introduction to Machine Learning

Programming Assignment 1

CALIFORNIA STATE UNIVERSITY FULLERTON

Due: March 1 @ 11:59 pm on Canvas

Maximum Points: 100

The assignment is an individual submission for graduate students and undergraduates can work in pairs.

Programming Language: Python

In this project you will use Python and Jupyter. Chapter 2 of the textbook can be useful for Jupyter Installation.

Platforms

Like ML projects in general, for this project you will need a Jupyter notebook/Google Colab with Python 3. Jupyter allows you to create documents mixing text, equations, code, and visualizations.

Preferred Python version: 3.8

Libraries and Code

This project must be implemented in pure Python with the Python Standard Library, without the use of any third-party libraries. You can use code from a whirlwind tour of Python. All other code must be your own original work.

Updates to the assignment specifications

If any changes or clarifications are made to the project specification, these will be posted on Canvas.

Academic misconduct

You are welcome to collaborate with your peers regarding the conceptualization and framing of the problem. For example, we encourage you to discuss what the assignment specification is asking you to do or what you would need to implement to be able to respond to a question. However, discussing beyond that will be considered cheating. Your submissions will be run through a plagiarism detection software program. We will invoke University's Academic Misconduct policy (https://www.fullerton.edu/senate/publications_policies_resolutions/ups/UPS%20300/UPS%20300.021.pdf) wherein appropriate levels of plagiarism or collusion are deemed to have taken place.

Goal of this assignment:

- Implement a K-Nearest Neighbor classification algorithm from scratch
- Use the K-Nearest Neighbor algorithm offered by Scikit learn to compare the results
- Identify dependent and independent variables
- Split the data into train and test sets

Every five years, Fullerton city in California conducts a satisfaction survey for a random number of Fullerton residents. The survey asks residents to rate their satisfaction, happiness, and well-being with Fullerton services. The dataset provided to you (HappinessData-1.csv) includes the resident's survey response for the years 2015-2020. The surveys are attached as PDFs. Within the data, an NA indicates that the question was not asked during those years or a blank indicates a nonresponse (no response received) to a question that was asked.

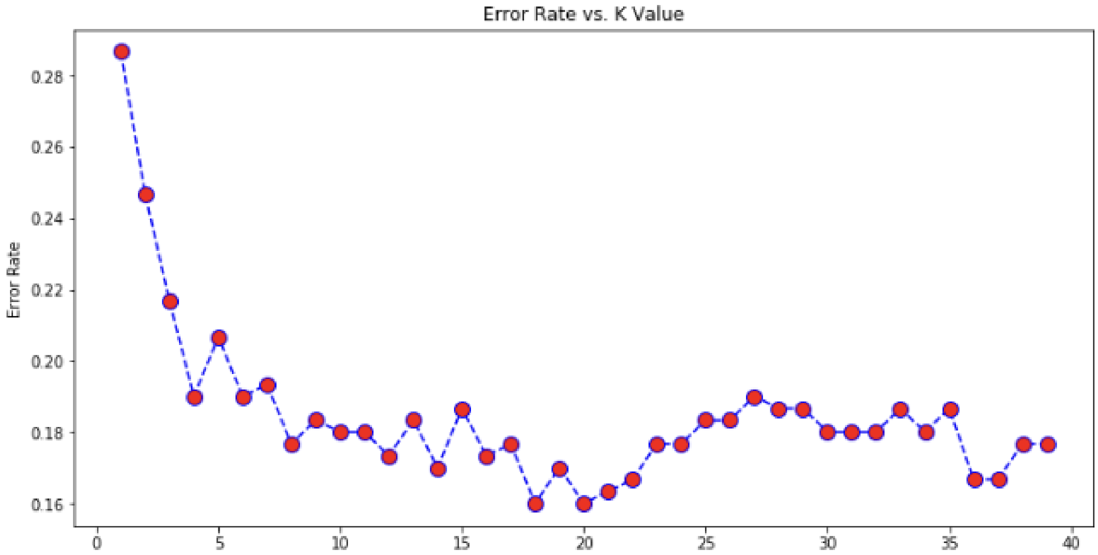
You are provided with a dataset with the following descriptive features:

1. City Services Availability
2. Housing Cost
3. Quality of schools
4. Community trust in local police
5. Community Maintenance
6. Availability of Community Room

All the features above have values 1 to 5. The first column in the data file has the target feature with a class label: 0 or 1; 0 for unhappiness, 1 for happiness (Note that before you apply any machine learning you will need to move the class label to be the last column in the csv file.

Tasks (10 points each):

1. Download the data file (.csv) provided on Canvas.
2. Move the class label column to be the last column in the csv file. Note the best way to do the task is to code it in your python script rather than manually moving the column.
3. During data preprocessing you should consider missing/NA values in the data.
4. Use Pearson Correlation for feature correlation (will be covered in class soon)
5. Implement k-nearest neighbor on your own without using any ML libraries. You can choose a distance metric of your choice. You are recommended to try out more than one metric covered in class to compare the performance and then choose the best.
6. Compare your results obtained in the previous step with the k-nearest neighbor offered by Scikit-learn. The textbook also walks you through how to use knn implementation from Scikit-learn.
7. For the above two tasks start by using the default number of neighbors(k=5) and then some random values for k, the number of neighbors and test its performance. Find the best k value by defining a function that Iterates over different values of k and show your results using an elbow plot that plots the error rate and the k-value. Below is a sample plot for your reference.



8. Build and fit your model for each k on training data and evaluate the performance on test data. The plot error rate for different k values.
9. You should test your model performance based on the train-test split **and** the n-fold cross-validation approach described in class.
10. For testing, use all the appropriate metrics such as the confusion matrix as discussed in class.

You can refer to the ML workflow diagram described in the class to ensure you follow all the necessary steps involved in building your first ML model from scratch.

Submission:

What to turn in on Canvas? (File names without extensions: CPS483PA1Yourfullname)

- Turn in all your code as Jupyter Notebooks (.ipynb file(s))
- Turn in all your code as Python files (.py) files

If you are using Google Colab instead of Jupyter Notebook:

- **Also turn in the link for the Google colab file.**
  The link will start with : https://colab.research.google.com

Reference:

W.W. Koczkodaj, T. Kakiashvili, A. Szymanska, J. Montero-Marin, R. Araya, J. Garcia-Campayo, K. Rutkowski, D. Strzalka, How to reduce the number of rating scale items without predictability loss? Scientometrics, 111(2): 581-593, 2017.