Data Science Project

Welcome! Your job here is to take the data sets below and build one or more functional predictive models to address the USE CASE (below). The basic evaluative principle is how well you do navigating, understanding, and modeling an unseen data set. We place a very high priority on code cleanliness and testing. To this end, please take the time to refactor your code to make it understandable to someone that has not worked on this problem before.

USE CASE:
A poll was run for a municipal runoff election in a metro area; 1577 voters were sampled of an estimated 400k registered voter population. We are tasked with building a machine learning model to predict candidate support (for labels *candidate_a* and *candidate_b*) probabilities with the highest accuracy. Given that voter turnout and intent is often low, these models would be used to identify and turn out potential supporters of our candidate (*candidate_a*).

DATA SETS:
voters.csv contains a subset raw client data we frequently receive from clients. It *does not contain normalized engineered features from our social graph profiles*.

data_dictionary.csv contains a description of the features in the voter file and a key to the values that need further explanation.

support.csv contains the labeled data we received from a poll as described in the use case.

graph.csv contains the connections from our social graph. If a relationship exists between two people, source and sink are the prim_ids of the individuals that match back to voters.csv and support.csv. Relationship score is the strength of the relationship.

DELIVERABLE:

- **support_probabilities.csv** consisting of columns prim_id and candidate_a_support_proba representing the primary id of the voter and the probability of support for *candidate_a*.
- **readme.pdf** a write up detailing your methods, procedures, evaluations and conclusions you arrived at. Specifically please include:
  - The evaluation of your machine learning model
  - Machine learning algorithms you attempt and why
  - Analysis of the results
  - Explanation and reasoning behind any encodings, normalizations, composites or other transformations on the original features
  - How you leveraged the graph data
  - What you would do with this data if you worked with it daily (as opposed to this brief time).
- **prediction.py** all the code (Python *highly* preferred, R, Scala etc) you used in this assignment. Using what you have given us and the source data, we must be able to replicate your results.

GROUND RULES:

- It is encouraged that you do not work on this project for more than 8 hours and should be delivered 3 days after the send date via email with attachments or appropriate Google Drive links.
- You may not have any person help you. You may, however, consult the Internet, your books, your notes, your parrot, etc.
- The appropriateness and quality of your models is an important factor in our evaluation (though we understand you had limited time). Provide evidence you have wrestled with the data and pertinent algorithms to squeeze out what knowledge you can in the time you have.
- You must write up your results and submit all the code you used to build your models.
- You are *highly encouraged* to create “graph based features” and measure the performance of your models before and after the addition of these features and/or any advanced ML improvements.

SUGGESTIONS:

– Spend 30 minutes understanding the data and setting up a modeling plan, 3-5 hours on feature engineering and modeling and 30 minutes writing your results.

– You should try 2-3 different modeling algorithms.

HINTS:

1) Consider a variety of models that support classification; we suggest starting with simpler techniques (don’t jump to using neural networks!).

2) Do not regard this as a classification problem; instead, treat it as a probability problem.