Data Science with Python
Assignment 1
Task 2: Short Answer Question

Name: Varun Ramesh

Student ID: s3793675

I certify that this is all my own original work. If I took any parts from elsewhere, then they were non-essential parts of the assignment, and they are clearly attributed in my submission. I will show I agree to this honor code by typing "Yes": *Yes*.

Table of Contents

1. Feature Engineering

a) Feature Columns

- Have you seen any of the 6 films in the Star Wars franchise? – Nominal
- Have you seen any of the 6 films in the Star Wars franchise? – Nominal.
- Please rank the Star Wars films in order of preference with 1 being your favourite film in the franchise and 6 being your least favourite film. – Ordinal.
- Do you consider yourself to be a fan of the Star Wars film franchise? – Nominal.
- Which of the following Star Wars films have you seen? Please select all that apply. – Nominal.
- Please state whether you view the following characters favourably, unfavourably, or are unfamiliar with him/her. – Ordinal.
- Which character shot first? – Nominal.
- Are you familiar with the Expanded Universe? – Nominal.
- Do you consider yourself to be a fan of the Expanded Universe? – Nominal.
- Do you consider yourself to be a fan of the Star Trek franchise? – Nominal.

b) Target Columns

- Gender - Nominal
- Age – Nominal
- Household Income – Nominal
- Education – Nominal
- Location (Census Region) – Nominal

c) ID-like columns

ID like columns are irrelevant in machine learning. So we are dropping the first column called Respondent ID.

d) Checking for Missing Values

Missing values are to be dropped or filled in with mean/medium/mode.

e) Making Categorical Features Numeric

Since all the features must be numeric, we deal with Nominal and Ordinal Features differently.

f) Encoding Nominal Features

As we can see above and looking at the Nominal columns, we can One-Hot encode all those columns to make it numeric. This way we are not looking at any arithmetic relationship between those columns.

g) Encoding Ordinal Features

- ### Ranks columns:

  Ranks of Star war movies has already been label encoded but there is further encoding needed because, 1 denotes best ranked Movie while 6 denotes worst ranked Movie. Here there is an arithmetic relationship between them which says 6 is greater than 1. To prevent this, we denote 5 as best ranked movie while 0 as the worst ranked movie

- ### Character Favourable column:
  Here there are parameters which has Highly Favourable, Somewhat Favourable, Neutral, Somewhat Unfavourable and Highly Unfavourable. This cannot be integer encoded because this would give arithmetic superiority between the column classes which is not needed. So, the best way to deal this would be one-hot encoding.

h) Target Feature:

Now target feature is the only one which is not numeric. To make this numeric first we subset the data with each of the demographics. Not all demographics have the same number of classes. It can be noticed that for Gender we can use a binary classifier but the rest of them we need to use Multi-class classifier because there are more than one class present. For example, Education has 'Less than High School', 'High School', 'Bachelors Degree', 'Masters Degree'. This makes it have many classes and we should move to Multi class classifier.

Taking all the columns and except the target column into numpy array and storing it into a variable. Then we convert the label-encoded target feature into a numpy array.

j) Train – Test Split:

Splitting the data set into train and test data is next most important thing to do before fitting in any Models or selecting them. Normalization of the data set is also very vital before splitting the data into train and test. The train and test data should have each of the target and label column in them.

2. Model Validation

The Starwars data which is cleaned and is in the form of a Numpy array. We have also split the columns into label features and target feature. Now the next process is to split this further into test set and train set (each of the labels and target features). Train data is to fit the parameters and make the machine understand about the data set and its nuisances. Train set is to assess the predictive power of the model for future entries.

- Feature Selection and Ranking: We first use the filter methods to examine the relationship between the descriptive features and the target feature. Or we then use the wrapper method as it selects different set of features for each wrapper. However, the wrapper method is much slower. Finally we can finish our Feature selection using methods like F-Score, Mutual Information, Random Forest Importance or using SPSA.
- Choosing accuracy or recall as a performance matrix. After choosing one of them as a performance matrix, we can analyse them in
  - Confusion Matrix
  - Precision, Recall and F1 Measures
  - Profit Matrix
  - ROC curves

3. Model Selection

- Looking at our Target columns, we can see that all of our target columns are Multi-Class Classification except Gender. For Gender we can use Binomial Classification. To understand how to do Multi-Class Classification we can use either 'one vs one' classification or 'one vs rest' classification.
  - One vs One: Here we pick 2 classes at a time (other classes are ignored for the time being) and train these two classifiers. The number of classes we get is N(N-1)/2.

    - One vs Rest: Here we pick one class and train a two class classifier. Here the samples of the selected class are on one side and the other samples are on the other side. This will give us N classifiers.
- Now since we have a numpy array of all the label and target columns, we can create an array of different models. Now we can fit our data in the array of models and generate confusion matrix and accuracy prediction to compare which model is the best for any future data which is given. Some of the models that can be used for prediction would be:
  1) SVC
  2) Decision Tree
  3) Random Forest
  4) K-Nearest Neighbours
- After choosing each of the models above, we can get the accuracy score of the model in many ways. One of them is by importing accuracy_score and printing the test_labels and the prediction array. Comparing this accuracy and choosing the highest one, we can take that model for applying the trained model for Unseen future data.

4. Trained Model to future Data

Having selected the model to be used, we can now give the unseen future data to it for future prediction. For example; if our model is chosen as Logistic Regressing, we would have

- Normalised our Data
- Split into Training and Test
- Chosen Logistic Regression as the best model based on accuracy
- And now finally feeding in the new unseen data into the same model to predict the new values using the function called 'model.predict()'

These steps can be used for all the models whichever we choose to be the right one and our prediction can be made.