Electronic Assignment Cover sheet

Please fill out and attach as the first page of Assignment.

Student (s) Number as per your student card:

Ashwin Ramdas 10534162

Course Title: MSc in Data Analytics

Lecturer Name: Mr. Abhishek Kaushik

Module/Subject Title: B9DA104 Machine Learning

Assignment Title: CA2

No of Words: 1650

A1) a)

Data Pre-processing refers to the transformation of data in such a way that it is now in a state where the machine can easily read it. Then, the algorithm can easily interpret the data.

There are six steps of Data Pre-processing, namely:-

1. Import Libraries: In this step, we import the libraries that will be needed in the program. These include matplotlib, scikit-learn(sklearn), plotly etc.
2. Import the dataset: In this step, we import the dataset onto the program, either in the form of a file or from an external source. For this the most commonly used library is Pandas.
3. Taking care of Missing Data in the Dataset: In this step, we take care of any NA values that might be in the dataset. We can simply “drop” the rows which have NA value if there are only few NA values. Else, we can also replace all the NA values by the median of that particular column, this is done by a class in python called Imputer.
4. Encoding the categorical data- Since the ML models are based on mathematical equations and calculations, the machine has complications understanding text and categorical data. Therefore, it is important to correctly encode categorical data in the dataset for the model
5. Splitting the data set into train set and test set-In this step, we split the data into train set and test set. You can either create a transformer for this or use the train_test_split class of sklearn. It is important because after the model is created, we need to test it on real world data for which the labels are already known
6. Feature scaling- It is the process of standardizing the range of features of the data or independent variables. This is done because different scales can make our prediction less accurate especially when the difference is a huge difference that can potentially alter results.

b)

Decision Tree is a decision-making support tool that uses a structure that look like roots of trees in which each internal node represents a decision to be taken by the algorithm on an attribute (For example, Whether next digit of a binary code is zero or one), while each of the branch represents outcomes of that test and the leaf nodes describe the class label

Information gain refers to the ratio of new information gained to the existing information in the dataset after transforming a dataset. It is also sometimes described as the reduction in entropy as it is calculated by finding the entropy before and after the transformation. It was proposed as there was a bias towards multi valued attributes and it was removed by taking the number and size of the branches while choosing an attribute

Entropy refers to the measure of disorder/uncertainty in a dataset. A decision Tree splits the data according to entropy. By that logic, Entropy affects the drawing of its boundaries. It is basically the difference between the predicted and the actual values in that dataset on which Decision Tree was applied and is very essential to construct a decision tree.

c)

The Chinese restaurant algorithm is a way of describing the Dirichlet process. The creators of this algorithms made it after getting inspired by the large crowds at massive Chinese restaurants in San Francisco's Chinatown. Therefore, they named it so.

In this algorithm we assume that the Chinese restaurant in question has unlimited booths and each booth has unlimited seats. "Where will a new person entering the restaurant will sit?" that is the question we're trying to answer.

So, according to the algorithm, Customer 1 (the first customer) can sit anywhere, while Customer 2 will have the following probabilities:

Probability of sitting at booth 1: $1 / (1 + \alpha)$

Probability of sitting at a new booth: $\alpha / (1 + \alpha)$

Therefore, for example, if there are 8 customers in the restaurant, of which customer 1, 2 & 5 are sitting at the first table; customer 3 is sitting at the second table; customer 4, 6 & 8 are on the third table and customer 7 is sitting at the fourth table,

The probability of a new customers will be as follows:

Probability of sitting at booth 1: $3 / (8 + \alpha)$

Probability of sitting at booth 2: $1 / (8 + \alpha)$

Probability of sitting at booth 3: $3 / (8 + \alpha)$

Probability of sitting at booth 4: $1 / (8 + \alpha)$

Probability of sitting at new booth: $\alpha / (8 + \alpha)$

Where α is a positive scalar hyperparameter, which set before start

the numerator is the number of people already at the booth (n_k)

And the denominator is the number of people already at the restaurant + α

To summarize,

The probability of the *n*th customer sitting at an existing table is $n_k / (\alpha + n - 1)$,

The probability of the *n*th customer sitting at a new table is $\alpha / (\alpha + n - 1)$.

Those tables increase in popularity which already has more people in it. Therefore empty tables are least likely to filled by new customers

A2)

I have chosen the dataset 'Supply_Chain_Shipment_Pricing_Data' which is present in data.gov at the link "https://catalog.data.gov/dataset/supply-chain-shipment-pricing-data" for my regression task and I have chosen the dataset 'rdu-weather-history' which is present in data.gov at the link "https://catalog.data.gov/dataset/local-weather-archive" for my classification task.

a)

For regression, I took this dataset after looking through the archive as It seemed as the best dataset of all I had seen which had a continuous value as the target variable.

I scoured through that website in the hope of finding a workable dataset but few of the links didn't work, some were too large, while some simply didn't have enough data. So, I started working on this dataset as soon as the link worked and there was enough data to work with.

The first step was moving the file into the particular folder that contained the notebook. Then, I imported the libraries that were to be included in the code. This included the sci-kit learn library, the pandas, the numpy library etc.

Then as the steps of data pre processing goes, the dataset was imported with the help of pandas library, where the read_csv command was used by me.

The data was stored in a variable called 'dana'. Then the variable 'dana' was called by me to see if the data had been correctly read.

Then, the target variable was decided to be the 'line item price'. So, I checked it's correlations and found Its respective correlations. However, I found that the variable 'shipment mode' was missing from the list which could obviously have a lot of effect on the price of the commodities and, therefore, I used get_dummies func. from the pandas library to create one hot encoding for it .

After that, I split the dataset into train and test set. For which I used the sklearn library.

Then I defined a class known as 'DataFrameSelector' to select only the relevant columns from the dataset. Another class was created known as 'MyLabelBinarizer' after the fact as my pipeline wasn't working perfectly.

Then, I created a numerical pipeline known as 'num_pipeline' and another pipeline known as 'cat_pipeline' for the categorical columns/values.

The numerical and categorical attributes to be selected were defined

Both the pipelines were then connected using FeatureUnion.

The data was then loaded into the pipeline. Here, both the train and test set were separately loaded to prepare them for models.

Then, to work on this I chose the algorithm 'DecisionTreeRegressor' as it was the one that worked best on this data. The model was created and then the data was fit in it.

Then, predictions were done to check for errors. The first error we checked was mean_squared_error, to find RMSE, followed by mean_absolute_error. Then, the cross_val_score was fount out.

After which, I defined a function to show the various RMSEs. Then, the accuracy was checked. Then the errors were again checked on the test set, which brought us to the end of the task.

b)

For the classification task, I loaded the data to the folder the notebook was after which I read the data using the read_csv function in the pandas library.

The data was loaded into a variable called ‘d’, which was then called to check if there was any discrepancies in the data. Then, the target variable was chosen to be the hail column as the data seemed sufficient.

So, labelbinarizer was used to convert the values of that column into binary values as they were string values. After that the correlation with that particular column was checked and the suitable columns selected. I divide the data into features and labels. After which, the data was divided into train and test set.

A class named DataFrameSelector was made to facilitate the pipeline for this task. Therefore, the pipeline was constructed next and the data was loaded into it. Both the test set and the train set was prepared through it for the models.

Naturally, the next step was to create the model and DecisionTreeClassifier was chosen as it performed exceptionally on this data.

The cross_val_scores were found out next and a function was defined to display those scores along with the mean and standard deviation.

The mean_squared_error was found out to get the RMSE, and then mean_absolute error was found.

After which, the r2 score was found. This was followed by the checking all those error on the test set to ensure that the model doesn’t over or under fit the data.

And with that the classification task was completed.