Lab 1

Predict Survival Rate
Context
Abstract: Hepatocellular Carcinoma dataset (HCC dataset) was collected at a University Hospital in Portugal. It contains real clinical data of 165 patients diagnosed with HCC.
This dataset is a challenging case for feature engineering and is subject of active research.
Relevant Information: HCC dataset was obtained at a University Hospital in Portugal and contains several demographics, risk factors, laboratory and overall survival features of 165 real patients diagnosed with HCC. The dataset contains 49 features selected according to the EASL-EORTC (European Association for the Study of the Liver - European Organisation for Research and Treatment of Cancer) Clinical Practice Guidelines, which are the current state-of-the-art on the management of HCC.

This is an heterogeneous dataset, with 23 quantitative variables, and 26 qualitative variables. Overall, missing data represents 10.22% of the whole dataset and only eight patients have complete information in all fields (4.85%). The target variables is the survival at 1 year, and was encoded as a binary variable: 0 (dies) and 1 (lives). A certain degree of class-imbalance is also present (63 cases labeled as “dies” and 102 as “lives”).

A detailed description of the HCC dataset (feature’s type/scale, range, mean/mode and missing data percentages) is provided in Santos et al. “A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients”, Journal of biomedical informatics, 58, 49-59, 2015.(Attached in your Lab 1 doc.)

There are two dataset files that you will be using for hcc-data.csv and hcc-data-complete-balance.csv.

Task 1(Only use hcc-data.csv)

Data Visualization: In this step you will analysis the datasets and try to find relationship between the attributes. This is the most important step in any machine learning problem.

1. Identify the dataset columns into nominal, categorical, continues etc. categories
2. Use dataframe.info and dataframe.describe to get the insights about the data.
3. Find the number of null values for each columns

Exceute this:->: data.isnull().sum(axis=0)

4. Know about the patients (Example of analysis for ages)

a. Find the oldest person
b. Find the youngest person
c. Find average age group
d. Find median age

e. Find the relationship between the deaths and ages(class column is your prediction variable)

f. Find the age groups whose survival rate is the largest

g. Find similar relationships for atleast 3-4 columns that you think can play a role in prediction (For example, Sex, alcohol consumption etc.)

h. Get more visuals on data distributions

i. Use plotCorrelationMatrix

ii. plotScatterMatrix

iii. plotPerColumnDistribution

Use information from the plots to get an intuition for selecting feature variables

i. Find missing values

i. Get the count of missing values

ii. Plot a head map for missing values

j. Applying different technique to handle missing values (For each technique verify your prediction results)

i. Use dropna

ii. Use replace na with zero or max value

iii. Use replace na with mean

iv. Search for additional techniques to handle null values, excluding the above three and test. (Include the all the techniques that you used in your report.)

k. Applying feature scaling technique if you think it is required. (Optional)

l. Split the dataset in train and test samples

m. Applying the regression model that you think is most suited for this problem.

Task 2 (Only use hcc-data-complete-balance.csv.)
Repeat Steps I and m and compare the your prediction result with the first technique.

Comparisontechnique:

We will use comfusion matrix to evaluate the performance
Compute Precision, Recall and F1 score for both Task 1 and Task 2

Task 3

a. Apply feature transform on the features used in task 1

a. Does varying the polynomial degrees changes your accuracy?

b. Can you identify if you model is underfitting or overfitting? (Hint use cross validation error and in-sample error plot to identify high bias and high variance.) Plot the relationships.

Sample code for polynomial regression.

```
# pass the order of your polynomial here  degree is 2
poly = PolynomialFeatures(2)

# convert to be used further to linear regression

X_transform = poly.fit_transform(X_train)
```

Submission details:

Python code file

Detailed report of your analysis and finds. Add plots and describe your finding on data analysis and model prediction. Compare the results for Task 1,2 and 3.