Lab 1

Predict Survival Rate

Due date: March 19th before class

Task 1 (kaggle dataset)

Submission - Kaggle & canvas

https://www.kaggle.com/t/7f4e0a0e69c942edaf3ebc0d71f5d75e

Dataset - hcc_train.csv, hcc_test.csv

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

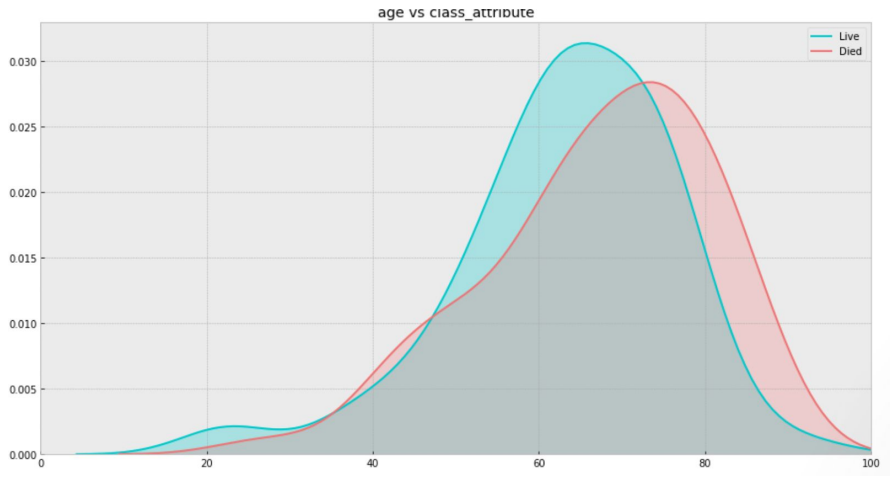
1. Identify the dataset columns into nominal, categorical, continues etc. categories
2. Use dataframe.info and dataframe.describe to get the insights about the data.
3. Find the number of null values for each columns
   Exceute this:->: data.isnull().sum(axis=0)
4. Know about the patients (Example of analysis for ages)
   a. Find the oldest person
   b. Find the youngest person
   c. Find the average age group
   d. Find median age
   e. Find the relationship between the deaths and ages(the class column is your prediction variable)

```
plt.figure(figsize=(15,8))

sns.kdeplot(
    data.age[data.class_attribute == 1],
    color="darkturquoise",
    shade=True
)

sns.kdeplot(
    data.age[data.class_attribute == 0],
    color="lightcoral",
    shade=True
)

plt.legend(['Live', 'Died'])
plt.title('age vs class_attribute')
plt.xlim(0,100)
plt.show()
```



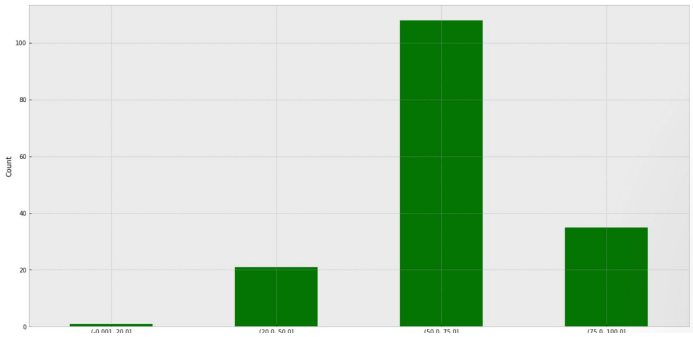
f. Find the age groups whose survival rate is the largest

```
bins = [0, 20, 50, 75, 100]

out = pd.cut(
    data.age,
    bins=bins,
    include_lowest=True
)

ax = out.value_counts(sort=False).plot.bar(
    rot=0,
    color="g",
    figsize=(20,10)
)

plt.xlabel('Age bins')
plt.ylabel('Count')
plt.show()
```



g. Find similar relationships for at least 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 heat map for missing values
j. Applying a 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 all the techniques that you used in your report.)
k. Applying the feature scaling technique if you think it is required. (Optional)
l. Applying the regression models that you think is most suited for this problem.
m. At least one of the models used to compute should be your own implementation using NumPy.
n. Upload your test data **predictions** to Kaggle competition in the correct submission format.

Task 2 (Use hcc-data-complete-balance.csv.)

Submission - canvas

1. Split the dataset in train and test samples
2. Applying the regression model that you think is most suited for this problem.
3. Compare your prediction result with the first technique.

Comparison technique:

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

Task 3

Submission - canvas

a. Apply feature transform on the features used in task 1
    a. Does varying the polynomial degree change your accuracy?
    b. Can you identify if your 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:

- Jupyter Note files (You can have one file to show task 1-3)
- Prediction file submitted to Kaggle
- A detailed report of your analysis and finds. Add plots and describe your findings on data analysis and model prediction. Compare the results for Tasks 1,2 and 3.

GRADING

Task 1 (50):

- Data visualization (hcc-train.csv): 20
- Own implementation of ML algorithm: 10
- Kaggle leaderboard standing: 20

Task 2 (20):

- Repeat steps to compare: 10
- Confusion matrix: 10

Task 3 (10)
Report (10)
QA (10)

NOTE: If your code and submission result on Kaggle does not match, no points will be awarded for LAB 1.