CSCI 184 Homework 1
Due Date: April 15th, 2022

1) One-Hot Encoding

For this part, you will focus on a house price detection dataset that comprises of 1460 rows and 81 columns. You will perform a comparative study between using one-hot encoded categorical columns as opposed to not using them at all.

What do you need to do?

1. Load the dataset from ‘train.csv’ into a pandas DataFrame and print it along with its shape.
2. For dealing with missing values, delete a column with any Null value (**Note: not recommended approach, but here focus is one-hot encoding**). Also delete the ‘ID’ column.
3. For simplicity, we will work with all the numerical columns but just a subset of the categorical columns. Among all the remaining categorical columns after step 2, choose the categorical columns that have less than 10 unique values.
4. Print the resultant dataframe and its shape after step 3.
5. Divide the data into X and Y, where X is the set of features and Y is the target variable. For this dataset the ‘SalePrice’ is the target variable.
6. For step 5, two approaches will be used (as discussed in class):
    a. Set of ONLY numerical features and dropping all Categorical features
    b. Set of features comprising both numerical and **one-hot encoded** categorical features.
7. Print the original dataframe joined with the one-hot encoded columns.
8. Once, the two sets of features are obtained from step 6, train a simple Linear Regression model from sklearn and obtain the Mean Absolute Error for both the cases:
    a. Mean Absolute Error when Dropping Categoricals:
    b. Mean Absolute Error with One-Hot Encoding
9. Write a report with screenshots of your results and the final results for step 8 a and b.
10. Submit your code as an .ipynb file and a document reporting your findings.

2) Feature Selection

For this part, you will focus on feature selection for the diabetes dataset.

You can download the data directly using this URL:
https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv

What do you need to do?

1. Load the dataset. The link does not have any names for headers. Add them with the following (in order: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, Outcome)
2. Print the DataFrame and its shape.
3. Convert the DataFrame object to NumPy array.
4. Segregate the data into X and Y. Here ‘Outcome’ is your target variable.
5. Perform Feature Extraction
    a. Using Filter-Based Method
        i. For numerical input and categorical output, refer to slides to figure out which method you want to use.
6. Set k value to 3,4 and 5 for step 5
7. For each value of k, specify the features that are extracted:

Method	Value of k	Feature Names
Filter-Based	3	:
	4	:
	5	:

8. After you have extracted the features, build a Logistic Regression Model from sklearn using two features:
    a. **Case 1:** Two of the features are from the list of extracted features using the Filter-based method for k = 3,4 and 5
        i. For example, if Pregnancies, Glucose, BMI and Age are the four features as a result of k = 4, choose any 2 of the features to train your model.

b. **Case 2:** Two of the features are columns that were not extracted in the feature extraction phase.

   i. For example, if Pregnancies, Glucose, BMI and Age are the four features as a result of k = 4, choose any 2 of the features from BloodPressure, SkinThickness, Insulin, DiabetesPedigreeFunction.

For each case, choose two combinations of features and train your model.

9. Evaluate your model. You may choose to evaluate the training data itself.
10. Report your findings in a table:

Case	Feature Names	Precision	Recall	F1-Score
1				
2				

Write a report with screenshots of your results and the final results for step 10. Submit your code as an .ipynb file and a document reporting your findings.

You may choose to use the same report for two of the parts, but please use two separate .ipynb files for the two parts.