INSTRUCTIONS

1. The assignment contains four questions. A few bonus questions are mentioned.
2. This assignment is due on **6th Feb, 23:59** (**No Further extensions**).
3. Assignment must be implemented in Python 3 only.
4. You are allowed to use libraries for data preprocessing (numpy, pandas etc) and for evaluation metrics, data visualization (matplotlib etc.).
5. You will be evaluated not just on the overall performance of the model and also on the experimentation with hyper parameters, data prepossessing techniques etc.
6. The report file must be a well documented jupyter notebook, explaining the experiments you have performed, evaluation metrics and corresponding code. The code must run and be able to reproduce the accuracies, figures/graphs etc.
7. For all the questions, you must create a train-validation data split and test the hyperparameter tuning on the validation set. Your jupyter notebook must reflect the same.
8. Any attempts at **plagiarism will be penalized heavily**.
9. Make su

1) REGRESSION

Please find the Diamond Price Prediction Data set https://drive.google.com/drive/folders/1qE1tm3Ke3uotTyv6SUqruI09t-AkcwRK?usp=sharing (https://drive.google.com/drive/folders/1qE1tm3Ke3uotTyv6SUqruI09t-AkcwRK?usp=sharing). "description.txt" contains the feature description of data, "diamonds.csv" has the data.

```
In [ ]: # To read data from diamonds.csv
        import pandas as pd
        headers = ["carat",     "cut","color","clarity","depth","table","
        price","x","y","z"]
        data = pd.read_csv('diamonds.csv', na_values='?',
                 header=None,  names = headers)
        data = data.reset_index(drop=True)
        data = data.iloc[1:]
        data.describe()
        #print(data)
```

```
# This is formatted as code
```

KNN Regression [Diamond Price Prediction Dataset]

1. a) Build a knn regression algorithm [using only python from scratch] to predict the price of diamonds.

```
In [ ]: # code for knn regression
```

1. b) Do we need to normalise data? [If so Does it make any difference?].

In [ ]:

```
# give proper explanation
```

1. Experiment with different distance measures[Euclidean distance, Manhattan distance, Hamming Distance] to handle categorical attributes.

In [ ]:

```
# show all the experiments
```

1. Report Mean Squared Error(MSE), Mean-Absolute-Error(MAE), R-squared (R2) score in a tabular form.

In [ ]:

```
# report a table
```

1. a) Choose different K values (k=2,3,5,7,11,16) and experiment. Plot a graph showing R2 score vs k.

In [ ]:

```
# plot
```

1. b) Are the R-squared scores the same? Why / Why not? How do we identify the best K? Suggest a computational procedure, with a logical explanation.

In [ ]:

```
# Explanation
```

1. a) Also, report the performance of scikit-learn's kNN regression algorithm.

In [ ]:

```
# scikit-learn KNN Regressor
```

1. b) Compare it with the algorithm you built. [you can use complexities, R2 score etc..]

In [ ]:

```
# Comparison
```

1. From the above experiments, what do you think are advantages and disadvantages of the knn regression algorithm?

In [ ]:

```
# report this  along with the experiments
```

2) Linear Regression

Dataset - same as above (Diamond Price Detection)

2a) Implement a Linear Regression model (from the scratch) taking suitable independent variables from the dataset.

Report and Calculate the error obtained.

```
In [ ]:
```

2b) What are the best suitable features you used to predict the price of the dataset and Why?

Idea: Use Correlation to get the suitable features and Report the values accordingly.

```
In [ ]: #code for Correlation between features and the Diamond Price.
```

Explanation for 2b) -

2c) Use the module Linear Regression from sklearn to predict the price of diamonds(considering the same attributes as before) and compare the result obtained with the above.

```
In [ ]: # import sklearn model
```

2d) Now, using the whole dataset, predict the price of the Diamonds using the module of Linear Regression from sklearn. Report the changes you have observed compared to before? Adding extra features did it make the prediction better or worse.Comment?

```
In [ ]:
```

2e) Now, compare the algorithms KNN regression and Linear Regression. What are the differences you have observed? Which is better and why. Your statements should be backed up with statistics.

Explanation -

2f) Plot the predicted values from KNN regression, Linear Regression and Actual Diamond Price.

```
In [ ]: #plot
```

KNN Classifier

In [ ]:

Decision Trees

The Wisconsin Breast Cancer Dataset(WBCD) can be found here(https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data (https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data))

This dataset describes the characteristics of the cell nuclei of various patients with and without breast cancer. The task is to classify a decision tree to predict if a patient has a benign or a malignant tumour based on these features.

Attribute Information:

```
#  Attribute                     Domain
   -- -----------------------------------------
   1. Sample code number            id number
   2. Clump Thickness               1 - 10
   3. Uniformity of Cell Size       1 - 10
   4. Uniformity of Cell Shape      1 - 10
   5. Marginal Adhesion             1 - 10
   6. Single Epithelial Cell Size   1 - 10
   7. Bare Nuclei                   1 - 10
   8. Bland Chromatin               1 - 10
   9. Normal Nucleoli               1 - 10
  10. Mitoses                       1 - 10
  11. Class:                        (2 for benign, 4 for malignant)
```

In [ ]:

```
import pandas as pd
headers = ["ID","CT","UCSize","UCShape","MA","SECSize","BN","BC","NN","Mitoses","Diagnosis"]
data = pd.read_csv('breast-cancer-wisconsin.data', na_values='?',
         header=None, index_col=['ID'], names = headers)
data = data.reset_index(drop=True)
data = data.fillna(0)
data.describe()
```

1. a) Implement a decision tree(from scratch using only python data structures) as a class.

In [ ]:

1. b) Train a decision tree object of the above class on the WBC dataset using misclassification rate, entropy and Gini as the splitting metrics.

In [ ]:

1. c) Report the accuracies in each of the above splitting metrics and give the best result.

In [ ]:

1. d) Experiment with different approaches to decide when to terminate the tree(number of layers, purity measure, etc). Report and give explanations for all approaches.

In [ ]:

1. e) Does standardisation and normalisation help? Report

Answer:

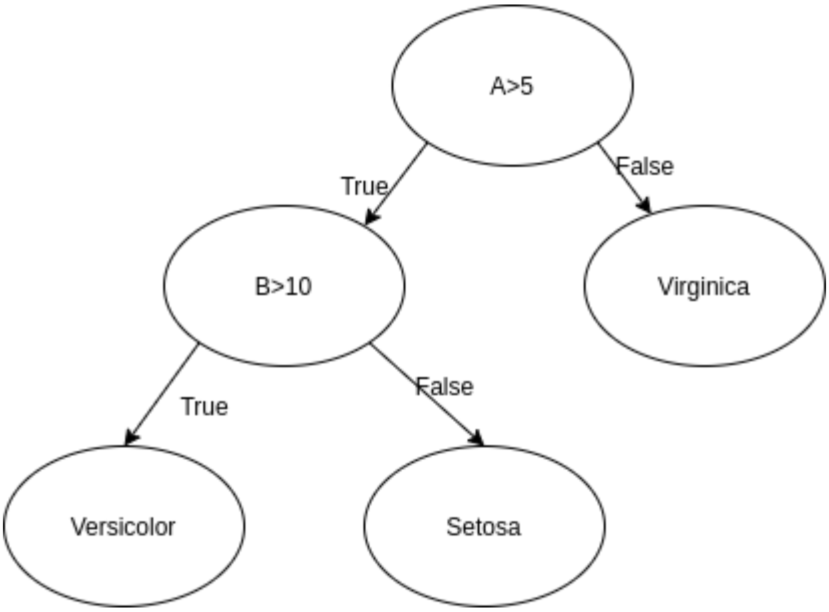
1. Compare your trained model with a model trained by the scikit-learn DecisionTreeClassifier module. Compare accuracies.

In [ ]:

1. Output a depth first traversal of both your decision tree and the one generated by scikit-learnin a file named `outputimp.txt` and `outputsck.txt` respectively in the following format and submit it along with the assignment.

```
<Node classification criteria in words.>
<Branch label>
  ... And so on, recursively.
```

For example, a depth first search traversal for the below decision tree would be:



```
Is A>5?
True Branch
  Is B>10?
  True Branch
    Versicolor
  Is B>10?
  False Branch
    Setosa
Is A>5?
False Branch
    Virginica
```

In [ ]:

1. Experiment with removing features that are redundant, highly correlated with other features and report accuracies of the resulting model. Explain your approach.

In [ ]:

1. Report the advantages and disadvantages of decision trees based on the above question.

Answer: