Project 2

Jinting Zhang

**Question 1:** Show the correlation among the attributes in the dataset. Then, find the **four** features with the highest correlations to the target.

Answer:

dataset.corr()

Here using the below code to find the four features with the highest correlations to the target .

#Corr\_matrix = dataset.corr()

#corr\_matrix["target"].sort\_values(ascending=False)

There are

worst\_radius

mean\_concave\_points

worst\_perimeter

worst\_concave\_points

**Question 2:** Split the dataset into the input features (X\_dataset) and the output target (y\_dataset).

Answer:

y\_dataset = dataset.pop('target')

X\_dataset = dataset[['mean\_concave\_points', 'worst\_radius', 'worst\_perimeter', 'worst\_concave\_points']]

**Question 3:** Find the normalized input features.

Answer:

X\_dataset\_mean = X\_dataset.mean()

X\_dataset\_std = X\_dataset.std()

X\_dataset\_norm = (X\_dataset - X\_dataset\_mean)/X\_dataset\_std

**Question 4:** Split the dataset into the training set (80%) and validation/test set (20%). Make sure that the random state is set to 100 or random\_state=100 for reproducibility.

Answer:

X\_train\_norm, X\_test\_norm, y\_train, y\_test = train\_test\_split(X\_dataset\_norm, y\_dataset, test\_size=0.2,

random\_state=100)

**Question 5:** Train the initialized model using the training data.

Answer:

model\_LR.fit(X\_train\_norm, y\_train)

**Question 6:** After training, evaluate the performance of the logistic regression model against the test set. To do so, run the following cell and report the accuracy.

Answer:

Evaluate the logistic regression model against the test set:

0.9473684210526315

**Question 7:** Complete the code and define a neural network with two more hidden layers with 8 and 6 nodes. Set the activation function for the hidden layers as relu and the kernel\_regularizer as l2 with l=0.001.

Answer:

layers.Dense(16, activation='relu',

kernel\_regularizer=tf.keras.regularizers.l2(l=0.001)),

layers.Dense(8, activation='relu',

kernel\_regularizer=tf.keras.regularizers.l2(l=0.001)),

**Question 8:** Train the neural network model (set epochs to 100).

Answer:

model\_NN.fit(X\_train\_norm, y\_train, epochs=100)

**Question 9:** Evaluate the performance of the neural network model against the test set. To do so, run the following cell and report the accuracy.

Answer:

- loss: 0.0908 - binary\_accuracy: 0.9737

**Question 10:** Which model did perform better against the test set in predicting the output. Moreover, is there any overfitting observed for the neural network? Finally, is there any significant improvement using the more complex model? Motivate your answer.

Answer: The comparison between performance of the two models (i.e. the linear

regression and the neural network) is simply done based on the evaluations on

the test set. Here the neural network model binary accuracy 0.9473684210526315 performs better against the test set in predicting the output compared to the linear regression model

0.9736842105263158. Here the neural network model can be possibly overfitting by checking validation metrics such as accuracy and loss. The validation metrics usually increase until a point where they stagnate or start declining when the model is affected by overfitting. During an upward trend, the model seeks a good fit, which, when achieved, causes the trend to start declining or stagnate. Here question 8 shows the training model of accuracy and loss.