Data Science for Mechanical Systems Assignment 4 (December/7/2021)

SACHIN FRANCIS DSOUZA – sfd2121

Problem 1.

1) **The Gini Index for the data set: 0.4675300607546925.**

2) **The value of splitting point: texture_mean ≤ 19.47**

   Range: 9.71 < 19.47 ≤ 39.28

3) **The value of splitting point (entropy = 0.953): texture_mean ≤ 18.635**

   Range: 9.71 < 18.635 ≤ 39.28

Problem 2.

1) **The Number of parameters in this NN: 26.**

2) **The value of the average prediction: 0.715533218076125.**

3) **The value of this error metric: 0.6983866220951158.**

Problem (3):-

P.T: In Lloyd's algorithm for K-means clustering, for a given cluster C with n observations, the averaged pairwise within cluster squared L2 norm equals to the sum of squared L2 norm of each point (in the cluster) to its center. That is:-

$$\frac{1}{n} \sum_{i,j \in C} \|x_i - x_j\|_2^2 = 2 \sum_{i \in C} \|x_i - \mu\|_2^2$$

where $\mu = \frac{1}{n} \sum_i x_i$, is the centroid of the cluster.

Proof:-

$\cancel{\forall C}$, $\frac{1}{n} \sum_{i,j \in C} \|x_i - x_j\|_2^2 = 2 \sum_{i \in C} \|x_i - \mu\|_2^2$

. We know that $\frac{1}{n} \sum_i x_i$, is the centroid of the cluster

K-means:-

a) $C_1, C_2, \ldots, C_K = \arg\min\left(\sum_{k=1}^{K} W(C_k)\right)$ where $W(C_k)$ measures within cluster variation

b) $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, 2, \ldots, n\}$ → very important

c) $C_k \cap C_{k'} = \emptyset$, for all $k \neq k'$

b) $C_1 \cup C_2 \cup \dots \cup C_K = \{1, 2, \dots, n\}$ → very important

c) $C_k \cap C_{k'} = \emptyset$, for all $k \neq k'$

Here $C_1, \dots, C_k$ denote sets containing the indices of the observations of each cluster.

K-means:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{d} (x_{ij} - x_{i'j})^2 \quad \text{— equation (2)}$$

K-means is based on minimizing the pairwise distance of data points within the same cluster.

Formally given $x_1, \dots, x_n \in R^d$ we partition these points into k clusters $C_1, \dots, C_k$ based on objective

Scanned with CamScanner

Lloyd's algorithm :

$$C_1, C_2, \ldots C_k = \arg\min \left(\sum_{k=1}^{K} W(C_k) \right) \quad -①$$

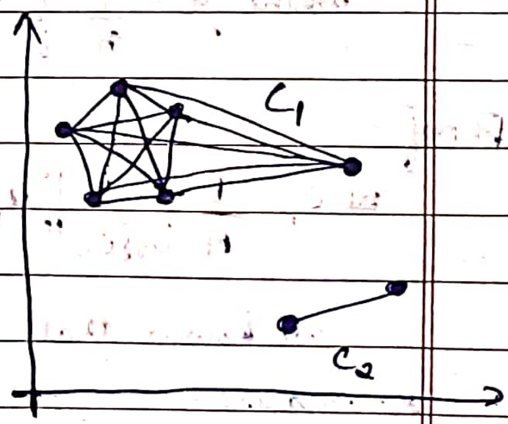
$$W(C_k) = \frac{1}{|C_k|} * \sum_{i,i' \in C_k} \sum_{j=1}^{d} (x_{ij} - x_{i'j})^2 \quad -②$$

Equate ② in ①

$$\underset{C_1 \ldots C_k}{\arg\min} \left(\sum_{i,l=1}^{k} \frac{1}{|C_k|} \sum_{i,j \in C_k} \| x_{ij}, x_{i'j} \|^2 \right) \rightarrow ③$$

Rewriting in simple terms

$$\min_{C_1, \ldots C_k} \sum_{l=1}^{k} \frac{1}{|C_l|} \sum_{i,j \in C_l} \|x_i - x_j\|^2$$



k-means → Lloyds algorithm

$$\boxed{\min_{C_1, \ldots C_k} \sum_{l=1}^{k} \frac{1}{|C_l|} \sum_{i,j \in C_l} \|x_i - x_j\|^2} \rightarrow \text{Rewritten ③ in simpler way} \rightarrow ④$$

Here $C_1, C_2, \ldots, C_k \Rightarrow l = 1, \ldots, k$. hence written as C_l

k-means → Lloyds algorithm

$$\boxed{\min_{C_1,\dots,C_k} \sum_{l=1}^{k} \frac{1}{|C_l|} \sum_{i,j\in C_l} \|x_i - x_j\|^2} \rightarrow \text{Rewritten (3) in simpler way} \rightarrow (4)$$

Here $C_1, C_2, \dots, C_k \Rightarrow l = 1, \dots, k$. hence written as C_l

Equation (4) is ~~Eta~~ K-means in a simpler way.

The cost function is a weighted average of the cluster variances with weights proportional to cluster size in terms of number of points $|C_l| \approx |C_k|$. we derive a more traceable form and tractable form.

Let → $\boxed{\mu = \frac{1}{|C_l|} \sum_{i\in C_l}^{n} x_i} \rightarrow (5)$

For derivation we derive both RHS & L.H.S

$$\sum_{l=1}^{k} \frac{1}{|C_l|} * \sum_{i,j\in C_l} \|x_i - x_j\|^2 = 2 \sum_{i\in C_l} \|x_i - \mu\|^2 \rightarrow (*)$$

This is equation (*)

Now splitting (★)

L.H.S :-

$$\sum_{i,j \in C_l} \|x_i - x_j\|^2 = \sum_{i,j \in C_l} \left(\|x_i\|^2 + \|x_j\|^2 - 2 < x_i, x_j > \right) \quad - (6)$$

$$\mu = \frac{1}{|C_l|}$$

Splitting (6) and applying the clusters and equation (5)

$$\sum_{i,j \in C_l} \|x_i - x_j\|^2 = \sum_{i \in C_l} \left(|C_l| \|x_i\|^2 + \sum_{j \in C_l} \|x_j\|^2 - 2 |C_l| < x_i, \mu > \right)$$

$$= 2|C_l| \sum_{i \in C_l} \|x_i\|^2 - 2|C_l|^2 \|\mu\|^2 \quad - (7)$$

$$\Rightarrow 2|C_l| \left[\sum_{i \in C_l} \|x_i\|^2 - |C_l| \|\mu\|^2 \right] \quad - (8)$$

~~Furthermore~~

Further more : R.H.S :-

$$\sum_{i \in C_l} \|x_i - \mu\|^2 = \sum_{i \in C_l} \left(\|x_i\|^2 + \|\mu\|^2 - 2 < x_i, \mu > \right)$$

$$\Rightarrow \sum_{i \in C_l} \left(\|x_i\|^2 + \|C_l\| \|\mu\|^2 - 2|C_l| \|\mu\|^2 \right)$$

$$\sum_{i \in C_l} \|x_i - \mu\|^2 \Rightarrow \sum_{i \in C_l} \|x_i\|^2 - |C_l| \|\mu\|^2 \rightarrow (9)$$

Scanned with CamScanner

$$\Rightarrow \sum_{i \in C_l} \left(\|x_i\|^2 + \|C_l\| \|\mu\|^2 - 2|C_l| \|\mu\|^2 \right)$$

$$\sum_{i \in C_l} \|x_i - \mu\|^2 \Rightarrow \sum_{i \in C_l} \|x_i\|^2 - |C_l| \|\mu\|^2 \rightarrow (9)$$

This is equivalent to equation (9) in equation (8)

From equation (8) $\Rightarrow$ we get $\Rightarrow$ (9)

$$\sum_{i,j \in C_l} \|x_i - x_j\|^2 = 2|C_l| \left[\sum_{i \in C_l} \|x_i\|^2 - |C_l| \|\mu\|^2 \right]$$

$$\boxed{\sum_{i,j \in C_l} \|x_i - x_j\|^2 = 2|C_l| \sum_{i \in C_l} \|x_i - \mu\|^2}$$

Hence rearranging

$$\boxed{\frac{1}{|C_l|} \sum_{i,j \in C_l} \|x_i - x_j\|_2^2 = 2 \sum_{i \in C_l} \|x_i - \mu\|_2^2} \Rightarrow (10)$$

Date
Page
classmate

where $\Rightarrow C_l = C_k \Rightarrow C_1 \cup C_2 \cup C_3 \cup C_k = n$

Thus from proof we get $\Rightarrow$

$$\sum_{l=1}^{k} \frac{1}{C_l} \sum_{i,j \in C_l} \left[\|x_i - x_j\|^2\right] = 2 \sum_{i \in C_l} \left[\|x_i - \mu\|^2\right]$$

here $\Rightarrow$ $\frac{1}{C_l} = \frac{1}{C_k}$ $\Rightarrow$ where $l \Rightarrow \{1, 2, 3, \ldots k\}$

and $C_1 \cup C_2 \cup C_3 \cup C_4 \cdots \cup C_k = \{1, 2, 3, 4, \ldots n\}$

$$\boxed{\therefore \frac{1}{n} \Rightarrow \sum_{i=1}^{l} \frac{1}{|C_l|} = \sum_{i=1}^{k} \frac{1}{|C_k|}} \quad - (11)$$

From (10)

$\therefore$

$$\boxed{\frac{1}{|C_k|} \sum_{i,j \in C_k} \|x_i - x_j\|_2^2 = 2 \sum_{i \in C_k} \|x_i - \mu\|_2^2}$$

$\therefore$ replacing C_k with n from (11)

$$\frac{1}{n} \sum_{i,j \in C} \|x_i - x_j\|_2^2 = 2 \sum_{i \in C} \|x_i - \mu\|_2^2$$

$$\boxed{\frac{1}{n} \sum_{i,j \in C} \|x_i - x_j\|_2^2 = 2 \sum_{i \in C} \|x_i - \mu\|_2^2} \Rightarrow \text{Proved}$$

where $\Rightarrow \mu = \frac{1}{} \sum^{n} x_i = \frac{1}{} \sum x_i$

From (10)

$$\boxed{\frac{1}{|C_k|} \sum_{i,j \in C_k} \|x_i - x_j\|_2^2 = 2 \sum_{i \in C_k} \|x_i - \mu\|_2^2}$$

$\therefore$ replacing C_k with n from (11)

$$\frac{1}{n} \sum_{i,j \in C} \|x_i - x_j\|_2^2 = 2 \sum_{i \in C} \|x_i - \mu\|_2^2$$

$$\boxed{\frac{1}{n} \sum_{i,j \in C} \|x_i - x_j\|_2^2 = 2 \sum_{i \in C} \|x_i - \mu\|_2^2} \Rightarrow \text{Proved}$$

where $\Rightarrow \mu = \frac{1}{|C_k|} \sum_{i \in C_k}^{n} x_i = \frac{1}{n} \sum_{i \in C} x_i$

Hence proved

and therefore minimizing $\sum_{l=1}^{k} \frac{1}{C_l} \sum_{i,j \in C_l} \|x_i - x_j\|_2^2$ is equivalent

to solving $\min_{\substack{C_1, \dots, C_k \\ \mu_1 \dots \mu_k}} \sum_{j=1}^{k} \sum_{i \in C_j} \|x_i - \mu\|_2^2$.

Scanned with CamScanner

```
%matplotlib notebook
import matplotlib.pyplot as plt
import numpy as np
from sklearn import tree
import pandas as pd
import seaborn as sns
import graphviz
from typing import Tuple
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Dense, Activation
```

```
sns.set(font_scale=1.5)
sns.set_style("whitegrid", {'grid.linestyle':'--'})
```

PROBLEM 1

```
cancer = pd.read_csv("/breast_cancer_data.csv")
cancer["label"] = cancer["diagnosis"].apply(lambda x: 0 if x == "B" else 1)
len(cancer['id'])
```

```
569
```

```
cancer.head()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoo
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	

```
column = cancer["texture_mean"]
maximum = column.max()
minimum = column.min()
print(maximum, minimum)
```

```
39.28 9.71
```

```
cancer.diagnosis = [1 if i=="M" else 0 for i in cancer.diagnosis]
cancer.head()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smo
0	842302	1	17.99	10.38	122.80	1001.0	
1	842517	1	20.57	17.77	132.90	1326.0	
2	84300903	1	19.69	21.25	130.00	1203.0	
3	84348301	1	11.42	20.38	77.58	386.1	
4	84358402	1	20.29	14.34	135.10	1297.0	

```
dt_model = tree.DecisionTreeClassifier(
    criterion="gini",
    max_depth=3,
)

features=["texture_mean"]
label = "label"
dt_model.fit(X=cancer[features], y=cancer[label])
```

```
DecisionTreeClassifier(max_depth=3)
```

```
def gini_index(p: float):
    """Gini index for a given binary class ratio."""
    return 2 * p * (1 - p)
```

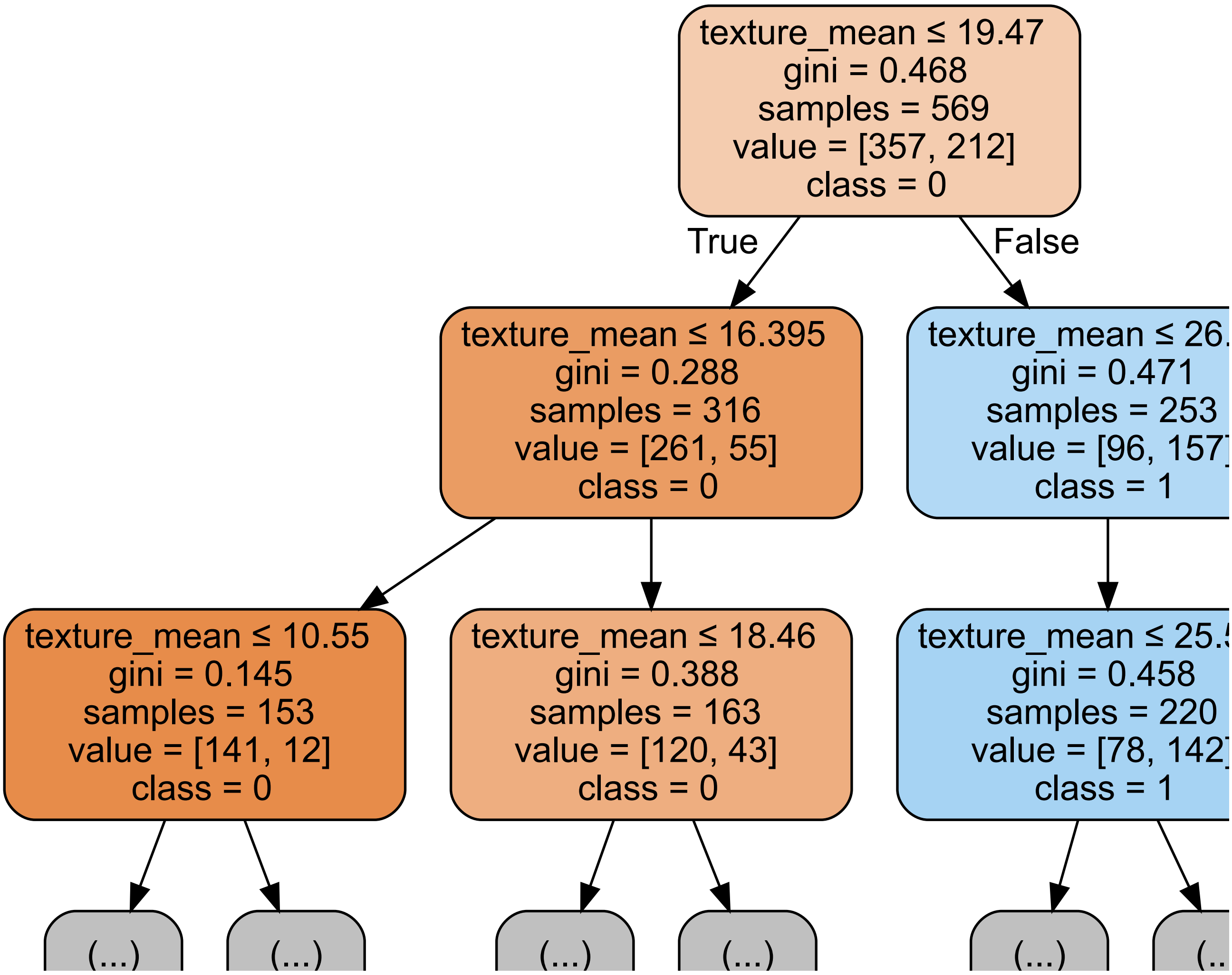
PROBLEM 1.1

```
print("The Gini Index for the dataset is : {}".format(gini_index(len(cancer[cancer.diagnos
```

```
The Gini Index for the dataset is : 0.4675300607546925
```

PROBLEM 1.2

```
dot_data = tree.export_graphviz(
    decision_tree=dt_model,
    out_file=None,
    feature_names=features,
    class_names=["0", "1"],
    filled=True,
    rounded=True,
    special_characters=True,
    max_depth=2,
)
graph = graphviz.Source(dot_data)
graph.render("cancer_tree")
graph
```



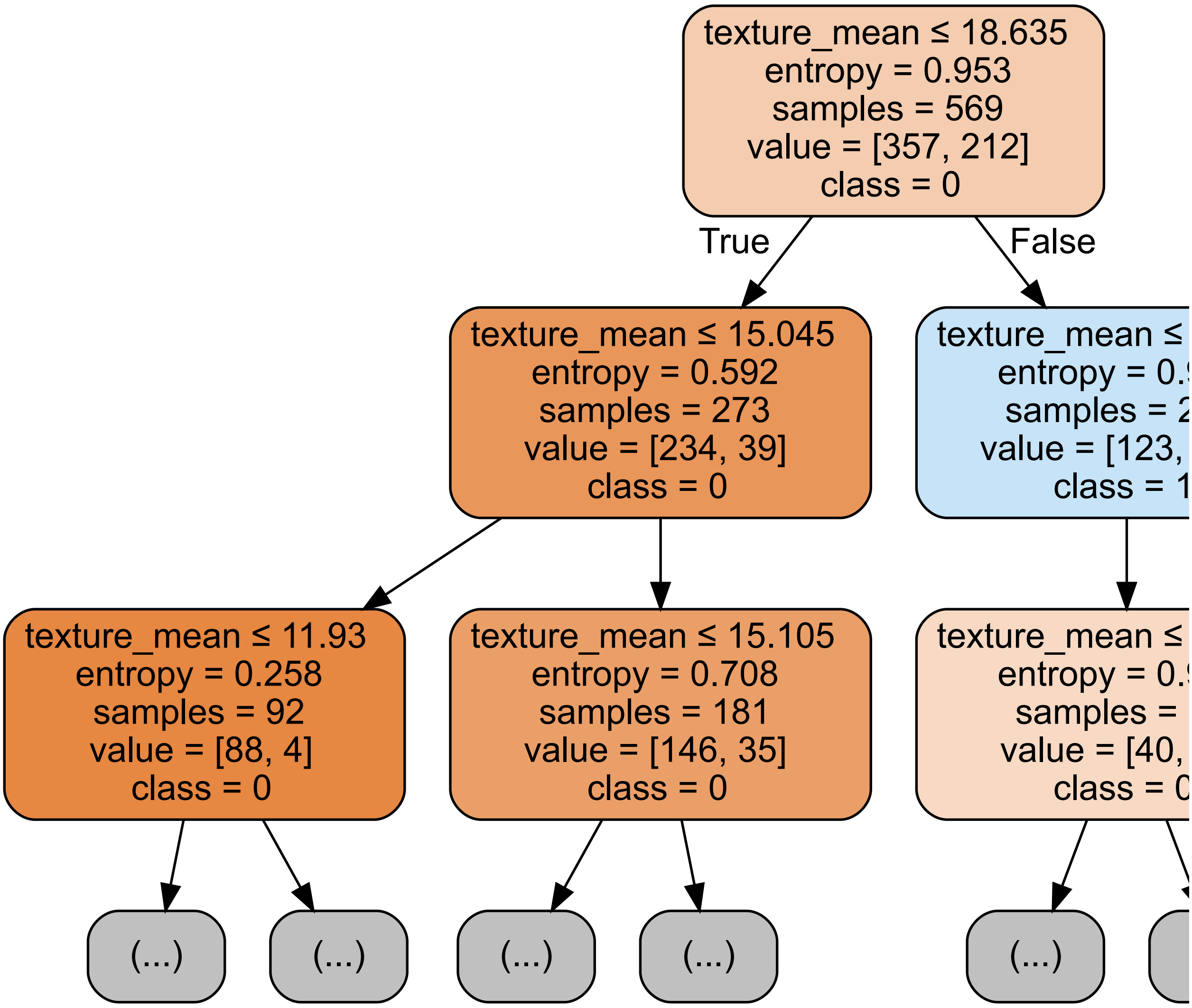

```
dt_model = tree.DecisionTreeClassifier(
    criterion="entropy",
    max_depth=3,
)

features=["texture_mean"]
label = "label"
dt_model.fit(X=cancer[features], y=cancer[label])
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=3)
```

PROBLEM 1.3

```
dot_data = tree.export_graphviz(
    decision_tree=dt_model,
    out_file=None,
    feature_names=features,
    class_names=["0", "1"],
    filled=True,
    rounded=True,
    special_characters=True,
    max_depth=2,
)
graph = graphviz.Source(dot_data)
graph.render("cancer_tree")
graph
```



PROBLEM 2

```
features = [
    "radius_extreme",
    "texture_extreme",
    "perimeter_extreme",

]
label = "label"

# train test split
X_raw, X_raw_test, Y, Y_test = train_test_split(cancer[features].values, cancer[label].val

# Standardize the input
scaler = StandardScaler()
scaler.fit(X_raw)
X = scaler.transform(X_raw)
X_test = scaler.transform(X_raw_test)

# formatting
Y = Y.reshape((-1, 1))
Y_test = Y_test.reshape((-1, 1))
```

```
def sigmoid(x):
    """Calculates sigmoid function."""
    return 1. / (1 + np.exp(-x))

def reLU(x):
    return np.maximum(0.0,x)

# parameters for the first layer
W_1 = np.ones((5, X.shape[1]))
print(f"Shape of W_1 is {W_1.shape}")

b_1 = np.ones((5, 1))*0.1
print(f"Shape of b_1 is {b_1.shape}")

# parameters for the second layer
W_2 = np.ones((1, 5))
print(f"Shape of W_2 is {W_2.shape}")

b_2 = np.ones((1, 1))*0.1
print(f"Shape of b_2 is {b_2.shape}")

# calculate the forward propagation
Z_1 = X @ W_1.T
print(f"\nShape of Z_1 is {Z_1.shape}")
print("Samples for Z_1:")
print(Z_1[:5])

A_1 = reLU(Z_1 + b_1.T)
print(f"Shape of A_1 is {A_1.shape}")
print("Samples for A_1:")
print(A_1[:5])

Z_2 = A_1 @ W_2.T
print(f"\nShape of Z_2 is {Z_2.shape}")
print("Samples for Z_2:")
print(Z_1[:5])

A_2 = Y_hat = sigmoid(Z_2 + b_2.T)
print(f"Shape of A_2 is {A_2.shape}")
print("Samples for A_2:")
print(A_2[:5])
```

```
Shape of W_1 is (5, 3)
Shape of b_1 is (5, 1)
Shape of W_2 is (1, 5)
Shape of b_2 is (1, 1)

Shape of Z_1 is (455, 5)
Samples for Z_1:
[[-2.95709089 -2.95709089 -2.95709089 -2.95709089 -2.95709089]
 [ 5.56621025  5.56621025  5.56621025  5.56621025  5.56621025]
 [-3.58131533 -3.58131533 -3.58131533 -3.58131533 -3.58131533]
 [-0.10923906 -0.10923906 -0.10923906 -0.10923906 -0.10923906]
 [-3.53688319 -3.53688319 -3.53688319 -3.53688319 -3.53688319]]
Shape of A_1 is (455, 5)
```

```
Samples for A_1:
[[0.         0.         0.         0.         0.        ]
 [5.66621025 5.66621025 5.66621025 5.66621025 5.66621025]
 [0.         0.         0.         0.         0.        ]
 [0.         0.         0.         0.         0.        ]
 [0.         0.         0.         0.         0.        ]]

Shape of Z_2 is (455, 1)
Samples for Z_2:
[[-2.95709089 -2.95709089 -2.95709089 -2.95709089 -2.95709089]
 [ 5.56621025  5.56621025  5.56621025  5.56621025  5.56621025]
 [-3.58131533 -3.58131533 -3.58131533 -3.58131533 -3.58131533]
 [-0.10923906 -0.10923906 -0.10923906 -0.10923906 -0.10923906]
 [-3.53688319 -3.53688319 -3.53688319 -3.53688319 -3.53688319]]
Shape of A_2 is (455, 1)
Samples for A_2:
[[0.52497919]
 [1.        ]
 [0.52497919]
 [0.52497919]
 [0.52497919]]
```

```
model = Sequential()
model.add(Dense(5, input_shape = (569,3), activation = 'relu'))
model.add(Dense(1, activation = 'sigmoid'))
```

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

PROBLEM 2.1

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 569, 5)            20

 dense_1 (Dense)             (None, 569, 1)            6

=================================================================
Total params: 26
Trainable params: 26
Non-trainable params: 0
_________________________________________________________________
```

PROBLEM 2.2

```
print(np.mean(Y_hat))
```

```
0.7155332180761279
```

```
avg_prediction = sum((A_2)/A_2.shape[0])[0]
print(avg_prediction)
```

```
0.715533218076125
```

PROBLEM 2.3

```
loss = -np.mean(np.multiply(Y, np.log(Y_hat)) + np.multiply(1 - Y, np.log(1 - Y_hat+1E-16)
print(loss)
```

```
0.6983866220951158
```

✓ 0s completed at 19:58