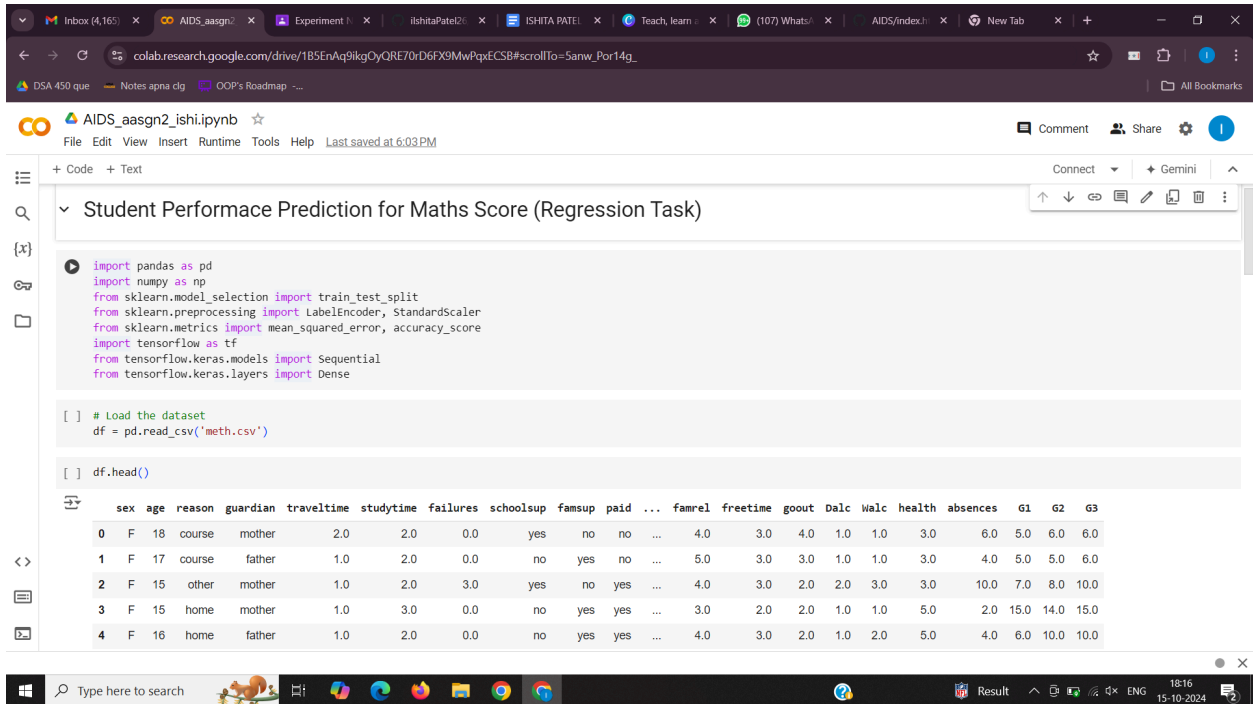


Assignment No: 2

Title : Develop cognitive application for Education using Neural Network.



colab.research.google.com/drive/1B5EnAq9ikgOyQRE70rD6FX9MwPqxECSB#scrollTo=5anw_Por14g

AIDS_aasgn2_ishi.ipynb

File Edit View Insert Runtime Tools Help Last saved at 6:03 PM

+ Code + Text

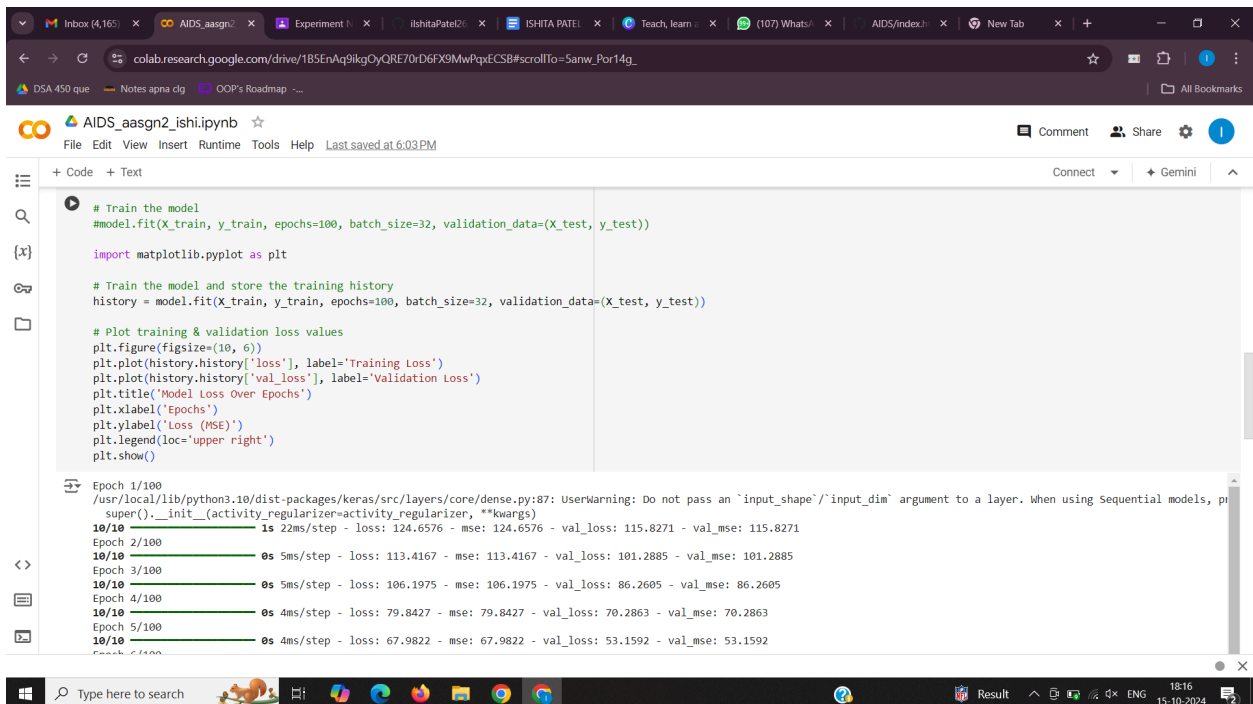
Student Performance Prediction for Maths Score (Regression Task)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import mean_squared_error, accuracy_score
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
[ ] # Load the dataset
df = pd.read_csv('meth.csv')
```

```
[ ] df.head()
```

	sex	age	reason	guardian	traveltime	studytime	failures	schoolsup	famsup	paid	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	F	18	course	mother	2.0	2.0	0.0	yes	no	no	...	4.0	3.0	4.0	1.0	1.0	3.0	6.0	5.0	6.0	6.0
1	F	17	course	father	1.0	2.0	0.0	no	yes	no	...	5.0	3.0	3.0	1.0	1.0	3.0	4.0	5.0	5.0	6.0
2	F	15	other	mother	1.0	2.0	3.0	yes	no	yes	...	4.0	3.0	2.0	2.0	3.0	3.0	10.0	7.0	8.0	10.0
3	F	15	home	mother	1.0	3.0	0.0	no	yes	yes	...	3.0	2.0	2.0	1.0	1.0	5.0	2.0	15.0	14.0	15.0
4	F	16	home	father	1.0	2.0	0.0	no	yes	yes	...	4.0	3.0	2.0	1.0	2.0	5.0	4.0	6.0	10.0	10.0



colab.research.google.com/drive/1B5EnAq9ikgOyQRE70rD6FX9MwPqxECSB#scrollTo=5anw_Por14g

AIDS_aasgn2_ishi.ipynb

File Edit View Insert Runtime Tools Help Last saved at 6:03 PM

+ Code + Text

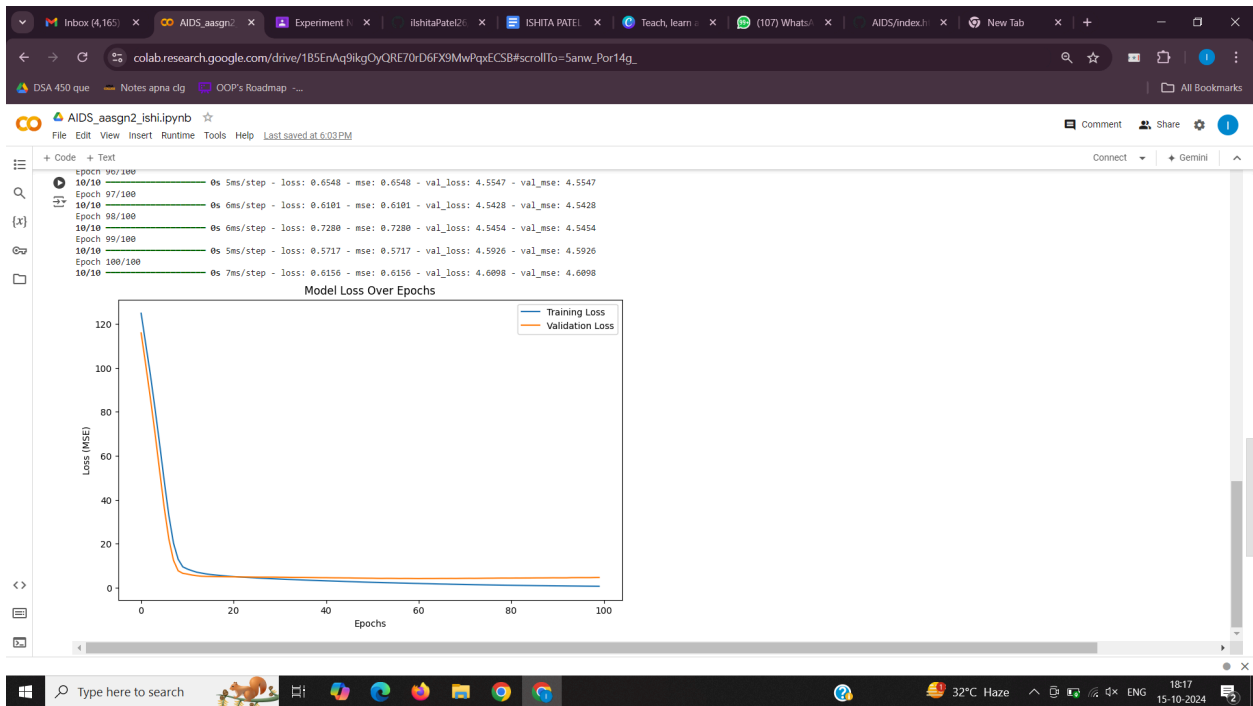
```
# Train the model
#model.fit(X_train, y_train, epochs=100, batch_size=32, validation_data=(X_test, y_test))

import matplotlib.pyplot as plt

# Train the model and store the training history
history = model.fit(X_train, y_train, epochs=100, batch_size=32, validation_data=(X_test, y_test))

# Plot training & validation loss values
plt.figure(figsize=(10, 6))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss Over Epochs')
plt.xlabel('Epochs')
plt.ylabel('Loss (MSE)')
plt.legend(loc='upper right')
plt.show()
```

```
Epoch 1/100
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an 'input_shape'/'input_dim' argument to a layer. When using Sequential models, p
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
10/10 1s 22ms/step - loss: 124.6576 - mse: 124.6576 - val_loss: 115.8271 - val_mse: 115.8271
Epoch 2/100
10/10 0s 5ms/step - loss: 113.4167 - mse: 113.4167 - val_loss: 101.2885 - val_mse: 101.2885
Epoch 3/100
10/10 0s 5ms/step - loss: 106.1975 - mse: 106.1975 - val_loss: 86.2605 - val_mse: 86.2605
Epoch 4/100
10/10 0s 4ms/step - loss: 79.8427 - mse: 79.8427 - val_loss: 70.2863 - val_mse: 70.2863
Epoch 5/100
10/10 0s 4ms/step - loss: 67.9822 - mse: 67.9822 - val_loss: 53.1592 - val_mse: 53.1592
```



colab.research.google.com/drive/1B5EnAq9ikgOyQRE70rD6FX9MwPqxECSB#scrollTo=5anw_Por14g_

AIDS_aasgn2_ishi.ipynb

File Edit View Insert Runtime Tools Help Last saved at 6:03 PM

+ Code + Text

```
[ ] # Make predictions and calculate metrics
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')
```

3/3 0s 3ms/step
Mean Squared Error: 5.081585334543468