NNDL: ICP3

Jyothi Kiran Boddeda
700769023

GITHUB LINK : https://github.com/jyothikiranboddeda/Neural-Network-Deep-Learning.git

VIDEO LINK :
https://drive.google.com/file/d/1mxDSuX_u88JfdGzQjACyc6bl98Bi0I3B/view?usp=sharing

1.Data Manipulation

a) Read the provided CSV file 'data.csv'.
b) https://drive.google.com/drive/folders/1h8C3mLsso-R-sIOLsvoYwPLzy2fJ4IOF?usp=sharing
c) Show the basic statistical description about the data.
d) Check if the data has null values.
e) Replace the null values with the mean
f) Select at least two columns and aggregate the data using: min, max, count, mean.
g) Filter the dataframe to select the rows with calories values between 500 and 1000.
h) Filter the dataframe to select the rows with calories values > 500 and pulse < 100.
i) Create a new "df_modified" dataframe that contains all the columns from df except for "Maxpulse".
j) Delete the "Maxpulse" column from the main df dataframe
k) Convert the datatype of Calories column to int datatype.
l) Using pandas create a scatter plot for the two columns (Duration and Calories).

ICP3.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('data.csv')

# Basic statistical description
print("Basic Statistical Description:")
print(data.describe())

# Check for null values
print("\nNull values before replacement:")
print(data.isnull().sum())

# Replace null values with the mean
data.fillna(data.mean(), inplace=True)
print("\nNull values after replacement:")
print(data.isnull().sum())

# Aggregation for 'Duration' and 'Calories'
aggregated_data = data[['Duration', 'Calories']].agg(['min', 'max', 'count', 'mean'])
print("\nAggregated Data:")
print(aggregated_data)
```

ICP3.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

[10]

```python
# Filter rows with Calories between 500 and 1000
filtered_calories = data[(data['Calories'] >= 500) & (data['Calories'] <= 1000)]
print("\nRows with Calories between 500 and 1000:")
print(filtered_calories)

# Filter rows with Calories > 500 and Pulse < 100
filtered_pulse = data[(data['Calories'] > 500) & (data['Pulse'] < 100)]
print("\nRows with Calories > 500 and Pulse < 100:")
print(filtered_pulse)

# Create a new dataframe excluding the 'Maxpulse' column
df_modified = data.drop(columns=['Maxpulse'])
print("\nModified DataFrame without 'Maxpulse':")
print(df_modified.head())

# Delete 'Maxpulse' from the main dataframe
data.drop(columns=['Maxpulse'], inplace=True)

# Convert the Calories column to int
data['Calories'] = data['Calories'].astype(int)

# Scatter plot for 'Duration' and 'Calories'
plt.scatter(data['Duration'], data['Calories'], color='green')
```

ICP3.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

[10]

```python
# Scatter plot for 'Duration' and 'Calories'
plt.scatter(data['Duration'], data['Calories'], color='green')
plt.title('Scatter Plot of Duration vs Calories')
plt.xlabel('Duration')
plt.ylabel('Calories')
plt.show()
```

```
72         90    100       127     700.0
73        150     97       127     953.2
75         90     98       125     563.2
78        120    100       130     500.4
83        120    100       130     500.0
90        180    101       127     600.1
99         90     93       124     604.1
101        90     90       110     500.0
102        90     90       100     500.0
103        90     90       100     500.4
106       180     90       120     800.3
108        90     90       120     500.3

Rows with Calories > 500 and Pulse < 100:
     Duration  Pulse  Maxpulse  Calories
65        180     90       130     800.4
70        150     97       129    1115.0
73        150     97       127     953.2
```

ICP3.ipynb

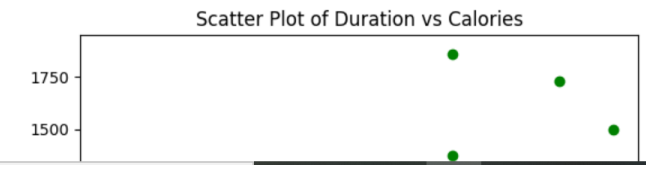
File Edit View Insert Runtime Tools Help

+ Code + Text

[10]

```
Rows with Calories > 500 and Pulse < 100:
     Duration  Pulse  Maxpulse  Calories
65        180     90       130     800.4
70        150     97       129    1115.0
73        150     97       127     953.2
75         90     98       125     563.2
99         90     93       124     604.1
103        90     90       100     500.4
106       180     90       120     800.3
108        90     90       120     500.3

Modified DataFrame without 'Maxpulse':
   Duration  Pulse  Calories
0        60    110     409.1
1        60    117     479.0
2        60    103     340.0
3        45    109     282.4
4        45    117     406.0
```

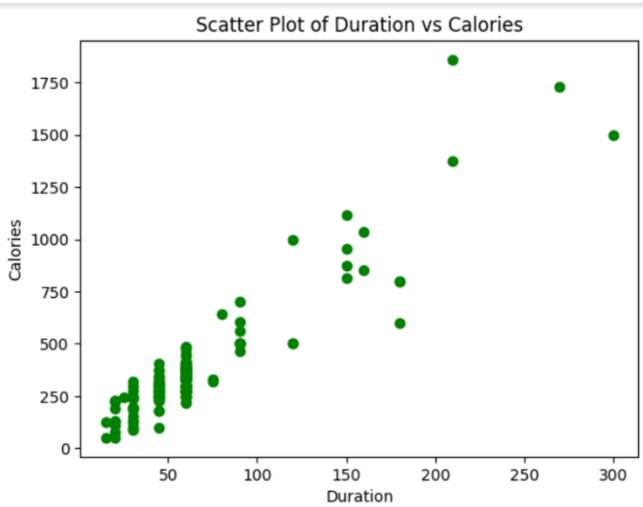


ICP3.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

[10]



2. Linear Regression

a) Import the given "Salary_Data.csv"
b) Split the data in train_test partitions, such that 1/3 of the data is reserved as test subset. c) Train and predict the model.
c) Calculate the mean_squared error
d) Visualize both train and test data using scatter plot

ICP3.ipynb

File Edit View Insert Runtime Tools Help

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Import the dataset
salary_data = pd.read_csv('Salary_Data.csv')

# Split the data into train and test partitions
X = salary_data[['YearsExperience']]
y = salary_data['Salary']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3, random_state=42)

# Train the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Calculate the Mean Squared Error
mse = mean_squared_error(y_test, y_pred)
```

ICP3.ipynb

File Edit View Insert Runtime Tools Help

```python
# Calculate the Mean Squared Error
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')

# Visualize the train and test data using scatter plot
plt.figure(figsize=(8, 4))

# Scatter plot for the training data
plt.scatter(X_train, y_train, color='blue', label='Train Data')

# Scatter plot for the test data
plt.scatter(X_test, y_test, color='red', label='Test Data')

# Plot the regression line
plt.plot(X_test, y_pred, color='black', linewidth=2, label='Regression Line')

plt.title('Salary vs Years of Experience')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.legend()
plt.show()
```

ICP3.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
Mean Squared Error: 35301898.887134895
```