Part 1: Data Exploration and Visualization (20 marks)

```
import pandas as pd

# Load CSV with comma delimiter
df = pd.read_csv('/content/sample_data/lifesat.csv', delimiter=',')

# Save as Excel file
df.to_excel('lifesat.xlsx', index=False)
```

Q1 (5 marks) Load the dataset lifesat.csv and display the first 5 rows.

```
import pandas as pd

# Load csv file
df = pd.read_csv('/content/sample_data/lifesat.csv')

# Display first 5 rows
display(df.head())
```

	Country	GDP per capita (USD)	Life satisfaction
0	Russia	26456.387938	5.8
1	Greece	27287.083401	5.4
2	Turkey	28384.987785	5.5
3	Latvia	29932.493910	5.9
4	Hungary	31007.768407	5.6

Q2 (5 marks) Print basic info and summary statistics.

```
# Print basic infos
display(df.info())

# Print summary statistics
display(df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27 entries, 0 to 26
Data columns (total 3 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Country               27 non-null     object
 1   GDP per capita (USD)  27 non-null     float64
 2   Life satisfaction     27 non-null     float64
dtypes: float64(2), object(1)
memory usage: 780.0+ bytes
None
```

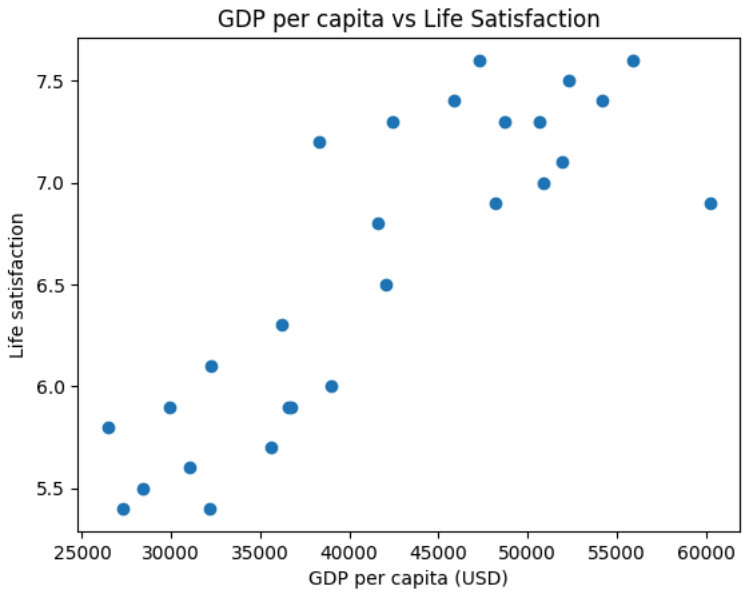
	GDP per capita (USD)	Life satisfaction
count	27.000000	27.000000
mean	41564.521771	6.566667
std	9631.452319	0.765607
min	26456.387938	5.400000
25%	33938.289305	5.900000
50%	41627.129269	6.800000
75%	49690.580269	7.300000
max	60235.728492	7.600000

Q3 (10 marks) Plot GDP per capita vs Life Satisfaction. Add labels and discuss the observed relationship.

```
import matplotlib.pyplot as plt

# Plot GDP between capita and Life Satisfaction
plt.scatter(df['GDP per capita (USD)'], df['Life satisf  'on'])
plt.xlabel('GDP per capita (USD)')
```

```
plt.ylabel('Life satisfaction')
plt.title('GDP per capita vs Life Satisfaction')
plt.show()
```



Part 2: Linear Regression Model (30 marks)

Q4 (5 marks) Extract input (X) and target (y). Print their shapes.

```
# Extract input (X) and target (y)
X = df[['GDP per capita (USD)']]
y = df['Life satisfaction']

# Print their shapes
# number of columns and rows
print("Shape of X:", X.shape)
print("Shape of y:", y.shape)
```

```
Shape of X: (27, 1)
Shape of y: (27,)
```

Q5 (10 marks) Train a Linear Regression model. Display coefficient and intercept.

```
from sklearn.linear_model import LinearRegression

# Train a Linear Regression model
model = LinearRegression()
model.fit(X, y)

# Display coefficient and intercept
print("Coefficient:", model.coef_)
print("Intercept:", model.intercept_)
```

```
Coefficient: [6.77889969e-05]
Intercept: 3.7490494273769093
```

Q6 (10 marks) Plot regression line with scatter plot of data.

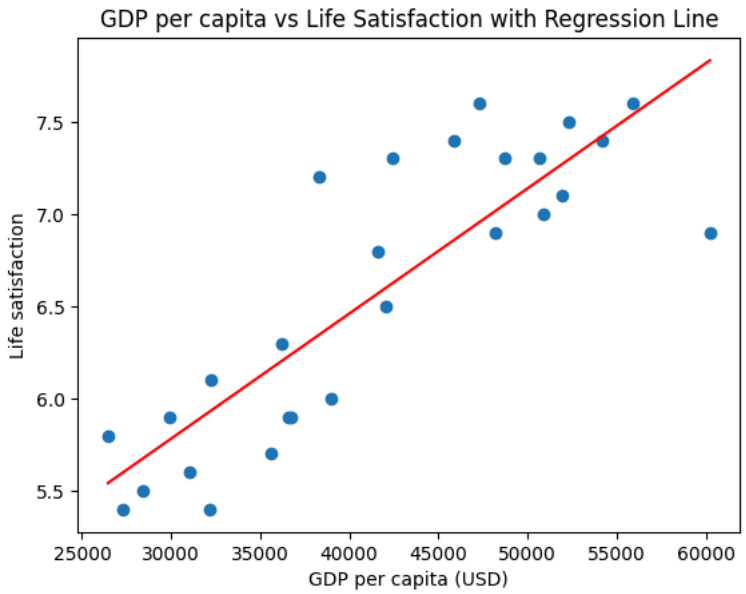
```
import matplotlib.pyplot as plt
import numpy as np
# removing warnings from display because nonreadable
warnings.filterwarnings("ignore", category=UserWarning, module="sklearn")


# Predict life satisfaction
X_new = np.linspace(X.min(), X.max(), 100).reshape(-1, 1)
y_predict = model.predict(X_new)

# Plot the original data points
plt.scatter(X, y)
```

```
# Plot the regression line
plt.plot(X_new, y_predict, color='red')

plt.xlabel('GDP per capita (USD)')
plt.ylabel('Life satisfaction')
plt.title('GDP per capita vs Life Satisfaction with Regression Line')
plt.show()
```




```
# Predict Life Satisfaction for GDP = 37,655.2 USD
gdp_new = [[37655.2]]  # Reshape the input as a 2D array
predicted_life_satisfaction = model.predict(gdp_new)

warnings.filterwarnings("ignore", category=UserWarning, module="sklearn")


print(f"Predicted Life Satisfaction for GDP = 37,655.2 USD: {predicted_life_satisfaction[0]:.2f}")

# Comment on the result
print("\nComment on the result:")
print(f"Based on the linear regression model, the predicted life satisfaction for a country with a GDP per capita of 37,655.2
print("This prediction is based on the linear relationship observed between GDP per capita and life satisfaction in the datas
```

```
Predicted Life Satisfaction for GDP = 37,655.2 USD: 6.30

Comment on the result:
Based on the linear regression model, the predicted life satisfaction for a country with a GDP per capita of 37,655.2 USD is ap
This prediction is based on the linear relationship observed between GDP per capita and life satisfaction in the dataset.
```

Part 3: K-Nearest Neighbors Regression (25 marks)

```
from sklearn.neighbors import KNeighborsRegressor

# Train a KNeighborsRegressor with n_neighbors=3
knn_model = KNeighborsRegressor(n_neighbors=3)
knn_model.fit(X, y)
```

```
KNeighborsRegressor
KNeighborsRegressor(n_neighbors=3)
```

```
# Predict Life Satisfaction for GDP = 37,655.2 USD using KNeighborsRegressor
gdp_new = [[37655.2]]  # Reshape the input as a 2D array
predicted_life_satisfaction_knn = knn_model.predict(gdp_new)
# removing warnings for more readability
warnings.filterwarnings("ignore", category=UserWarning, module="sklearn")


print(f"Predicted Life Satisfaction for GDP = 37,655.2 USD (KNN): {predicted_life_satisfaction_knn[0]:.2f}")
print(f"Predicted Life Satisfaction for GDP = 37,655.2 USD (Linear Regression): {predicted_life_satisfaction[0]:.2f}")

# Compare the results
```

```
print("\nComparison of results:")
print(f"The Linear Regression model predicted {predicted_life_satisfaction[0]:.2f}, while the KNeighborsRegressor predicted {pr
difference = predicted_life_satisfaction[0] - predicted_life_satisfaction_knn[0]
print(f"Difference: {difference:.2f} ")
print("The difference in predictions is due to the different approaches of the two models. Linear Regression finds a linear rel
```

```
Predicted Life Satisfaction for GDP = 37,655.2 USD (KNN): 6.37
Predicted Life Satisfaction for GDP = 37,655.2 USD (Linear Regression): 6.30

Comparison of results:
The Linear Regression model predicted 6.30, while the KNeighborsRegressor predicted 6.37.
Difference:  -0.07
The difference in predictions is due to the different approaches of the two models. Linear Regression finds a linear relationsh
```

```
#from sklearn.neighbors import KNeighborsRegressor
import matplotlib.pyplot as plt
import numpy as np

warnings.filterwarnings("ignore", category=UserWarning, module="sklearn")

# defining values as a Professor gave
n_neighbors_values = [1, 3, 5, 10]

# creating range for GPR values for PLOT
X_new = np.linspace(X.min(), X.max(), 100).reshape(-1, 1)

plt.figure(figsize=(10, 6))
plt.scatter(X, y, label='Original Data')

# Train and plot KNN models for each n_neighbors value
for n in n_neighbors_values:
    knn_model = KNeighborsRegressor(n_neighbors=n)
    knn_model.fit(X, y)
    y_predict_knn = knn_model.predict(X_new)
    plt.plot(X_new, y_predict_knn, label=f'KNN (n={n})')

plt.xlabel('GDP per capita (USD)')
plt.ylabel('Life satisfaction')
plt.title('GDP per capita vs Life Satisfaction with KNN Regression for different n_neighbors')
plt.legend()
plt.show()
```