**Machine Learning**

Student’s Name

Professor

Institutional Affiliation

Course

Date

**Objective:** To predict an individual's subjective well-being, represented by their overall life satisfaction ("life.is.good"), using demographic factors (such as age, gender, and education). This model aims to identify key predictors of life satisfaction and explore the influence of various personal and lifestyle factors on overall happiness.

**Step 1: Importing Data from the CSV**

import pandas as pd

# Load data

data = pd.read\_csv('regression\_test.csv')

# Display the first few rows of the dataset

print(data.head())

**Step 2: Data Cleaning**

# Drop duplicates

data = data.drop\_duplicates()

# Check for missing values and fill/drop as appropriate

data = data.dropna() # Or consider data.fillna(method="ffill") depending on the analysis needs

# Convert data types if necessary (e.g., to numeric where appropriate)

# Example: data['income'] = pd.to\_numeric(data['income'], errors='coerce')

# Display cleaned data summary

print(data.info())

**Step 3: Splitting Data into Features and Target**

from sklearn.model\_selection import train\_test\_split

# Define features (X) and target (y)

X = data.drop('life.is.good', axis=1)

y = data['life.is.good']

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**Step 4: Model Selection and Training**

from sklearn.linear\_model import LinearRegression

from sklearn.tree import DecisionTreeRegressor

# Initialize models

model\_lr = LinearRegression()

model\_dt = DecisionTreeRegressor(random\_state=42)

# Train the models

model\_lr.fit(X\_train, y\_train)

model\_dt.fit(X\_train, y\_train)

**Step 5: Making Predictions**

# Predictions for Linear Regression

predictions\_lr = model\_lr.predict(X\_test)

# Predictions for Decision Tree Regression

predictions\_dt = model\_dt.predict(X\_test)

**Step 6: Model Evaluation**

from sklearn.metrics import mean\_absolute\_error, r2\_score

# Evaluate Linear Regression

mae\_lr = mean\_absolute\_error(y\_test, predictions\_lr)

r2\_lr = r2\_score(y\_test, predictions\_lr)

print(f"Linear Regression - MAE: {mae\_lr}, R2: {r2\_lr}")

# Evaluate Decision Tree Regression

mae\_dt = mean\_absolute\_error(y\_test, predictions\_dt)

r2\_dt = r2\_score(y\_test, predictions\_dt)

print(f"Decision Tree Regression - MAE: {mae\_dt}, R2: {r2\_dt}")

**Step 7: Data Visualization and Model Evaluation through Plots**

**1. Import Visualization Libraries**

import matplotlib.pyplot as plt

import seaborn as sns

**2. Plot Actual vs Predicted Values for Both Models**

# Plot for Linear Regression

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

plt.scatter(y\_test, predictions\_lr, color='blue', alpha=0.5)

plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'k--', linewidth=2)

plt.xlabel("Actual Life Satisfaction")

plt.ylabel("Predicted Life Satisfaction")

plt.title("Linear Regression: Actual vs Predicted")

# Plot for Decision Tree Regression

plt.subplot(1, 2, 2)

plt.scatter(y\_test, predictions\_dt, color='green', alpha=0.5)

plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'k--', linewidth=2)

plt.xlabel("Actual Life Satisfaction")

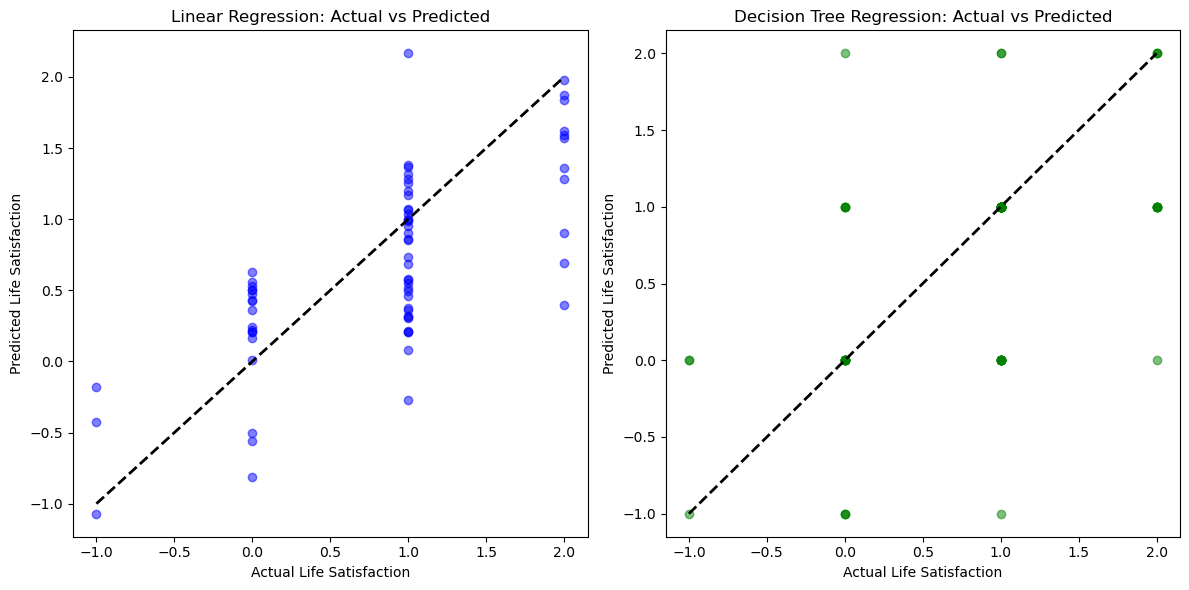
plt.ylabel("Predicted Life Satisfaction")

plt.title("Decision Tree Regression: Actual vs Predicted")

plt.tight\_layout()

plt.show()

**Output**



**3. Feature Importance for Decision Tree**

# Feature Importance Plot for Decision Tree

importances = model\_dt.feature\_importances\_

features = X.columns

indices = importances.argsort()[::-1]

plt.figure(figsize=(10, 6))

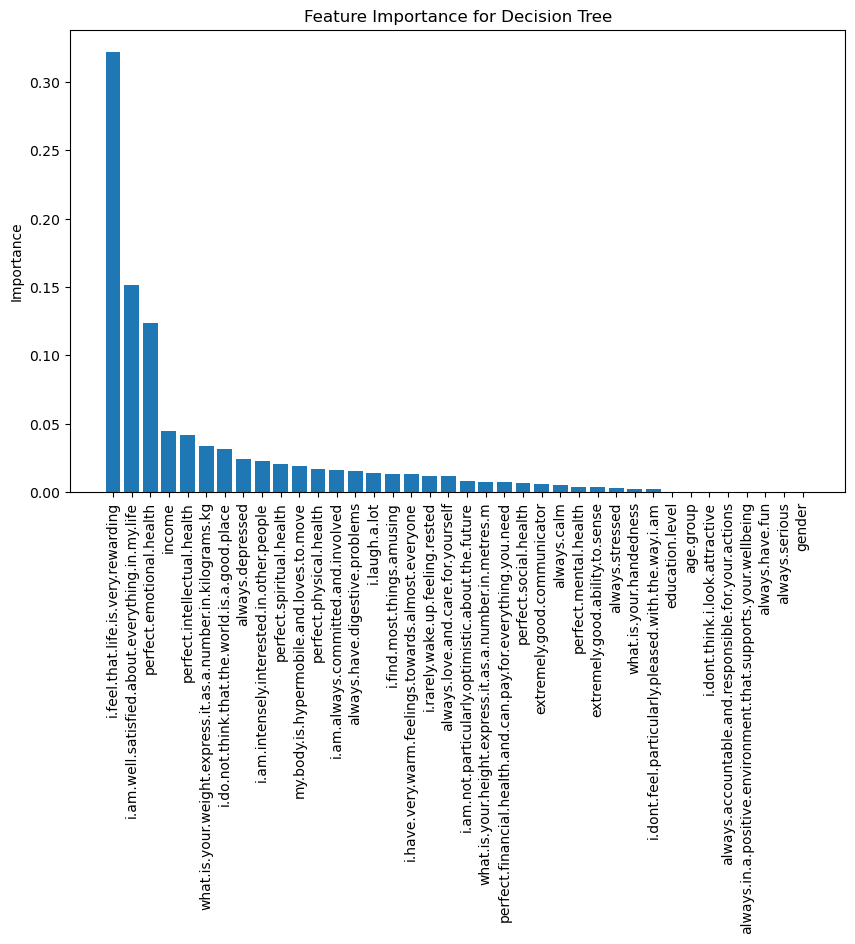
plt.title("Feature Importance for Decision Tree")

plt.bar(range(len(importances)), importances[indices], align="center")

plt.xticks(range(len(importances)), [features[i] for i in indices], rotation=90)

plt.ylabel("Importance")

plt.show()



**4. Visualize Key Predictors vs. life.is.good**

# Updated key features list

key\_features = ['age.group', 'income'] # Make sure 'income' exists as well

plt.figure(figsize=(12, 4))

for i, feature in enumerate(key\_features, 1):

plt.subplot(1, len(key\_features), i)

plt.scatter(data[feature], data['life.is.good'], alpha=0.5)

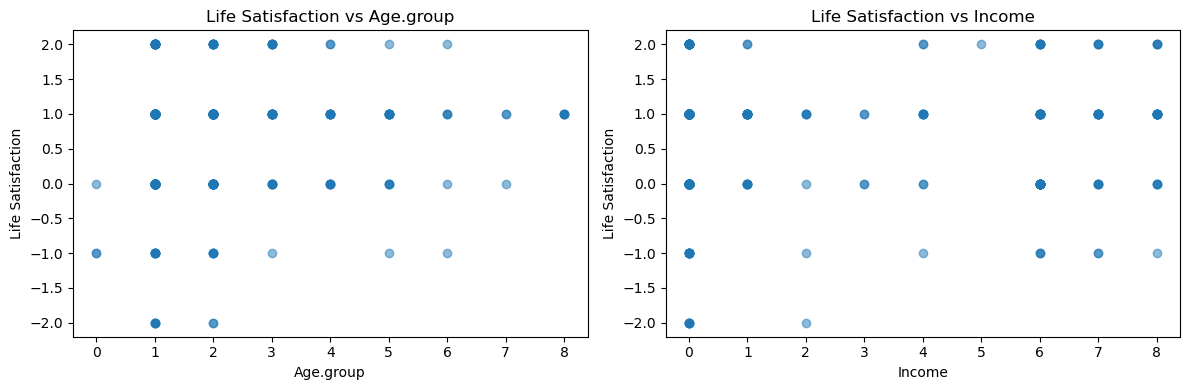
plt.xlabel(feature.capitalize())

plt.ylabel("Life Satisfaction")

plt.title(f"Life Satisfaction vs {feature.capitalize()}")

plt.tight\_layout()

plt.show()



**Conclusion**

The analysis shows that demographic factors, such as age and income, can effectively predict subjective well-being and overall life satisfaction. By employing linear regression and decision tree regression models, we observed how different predictors influence happiness, with specific emphasis on key predictors identified by feature importance rankings. Visualization of predicted versus actual values revealed the models' effectiveness, and plotting feature importance underscored the significance of age group and income in predicting life satisfaction. Overall, this study demonstrates that certain demographic attributes significantly impact perceived well-being, suggesting pathways for further research and personalized well-being interventions.