DefineAndSolveMLProblem

August 2, 2025

1 Lab 8: Define and Solve an ML Problem of Your Choosing

[2]:
```python
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

In this lab assignment, you will follow the machine learning life cycle and implement a model to solve a machine learning problem of your choosing. You will select a data set and choose a predictive problem that the data set supports. You will then inspect the data with your problem in mind and begin to formulate a project plan. You will then implement the machine learning project plan.

You will complete the following tasks:

1. Build Your DataFrame
2. Define Your ML Problem
3. Perform exploratory data analysis to understand your data.
4. Define Your Project Plan
5. Implement Your Project Plan:
    - Prepare your data for your model.
    - Fit your model to the training data and evaluate your model.
    - Improve your model's performance.

1.1 Part 1: Build Your DataFrame

You will have the option to choose one of four data sets that you have worked with in this program:

- The "census" data set that contains Census information from 1994: `censusData.csv`
- Airbnb NYC "listings" data set: `airbnbListingsData.csv`
- World Happiness Report (WHR) data set: `WHR2018Chapter2OnlineData.csv`
- Book Review data set: `bookReviewsData.csv`

Note that these are variations of the data sets that you have worked with in this program. For example, some do not include some of the preprocessing necessary for specific models.

Load a Data Set and Save it as a Pandas DataFrame The code cell below contains filenames (path + filename) for each of the four data sets available to you.

Task: In the code cell below, use the same method you have been using to load the data using `pd.read_csv()` and save it to DataFrame `df`.

You can load each file as a new DataFrame to inspect the data before choosing your data set.

[5]:
```
# File names of the four data sets
adultDataSet_filename = os.path.join(os.getcwd(), "data", "censusData.csv")
airbnbDataSet_filename = os.path.join(os.getcwd(), "data", "airbnbListingsData.
↪csv")
WHRDataSet_filename = os.path.join(os.getcwd(), "data",␣
↪"WHR2018Chapter2OnlineData.csv")
bookReviewDataSet_filename = os.path.join(os.getcwd(), "data", "bookReviewsData.
↪csv")


df = pd.read_csv(WHRDataSet_filename)

df.head()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1562 entries, 0 to 1561
Data columns (total 19 columns):
 #   Column                                                    Non-Null Count
Dtype
---  ------                                                    --------------
-----
 0   country                                                   1562 non-null
object
 1   year                                                      1562 non-null
int64
 2   Life Ladder                                               1562 non-null
float64
 3   Log GDP per capita                                        1535 non-null
float64
 4   Social support                                            1549 non-null
float64
 5   Healthy life expectancy at birth                          1553 non-null
float64
 6   Freedom to make life choices                              1533 non-null
float64
 7   Generosity                                                1482 non-null
float64
 8   Perceptions of corruption                                 1472 non-null
float64
 9   Positive affect                                           1544 non-null
float64
 10  Negative affect                                           1550 non-null
```

```
float64
 11  Confidence in national government                        1401 non-null
float64
 12  Democratic Quality                                       1391 non-null
float64
 13  Delivery Quality                                         1391 non-null
float64
 14  Standard deviation of ladder by country-year             1562 non-null
float64
 15  Standard deviation/Mean of ladder by country-year        1562 non-null
float64
 16  GINI index (World Bank estimate)                         583 non-null
float64
 17  GINI index (World Bank estimate), average 2000-15        1386 non-null
float64
 18  gini of household income reported in Gallup, by wp5-year  1205 non-null
float64
dtypes: float64(17), int64(1), object(1)
memory usage: 232.0+ KB
```

1.2 Part 2: Define Your ML Problem

Next you will formulate your ML Problem. In the markdown cell below, answer the following questions:

1. List the data set you have chosen.
2. What will you be predicting? What is the label?
3. Is this a supervised or unsupervised learning problem? Is this a clustering, classification or regression problem? Is it a binary classificaiton or multi-class classifiction problem?
4. What are your features? (note: this list may change after your explore your data)
5. Explain why this is an important problem. In other words, how would a company create value with a model that predicts this label?

1. The dataset I have chosen is the world happiness report. (WHRDataSet_filename)
2. I will be predicting the happiness score of a country in a specific year. My label will be 'Life Ladder'.
3. This is a supervised learning problem because we have a specific label. It is a regression problem because the label is a continuous numerical value.
4. My features are: country, year, Log GDP per capita, Social support, Healthy life expectancy at birth, Freedom to make life choices, Generosity, Perceptions of corruption, Postive affect, Negative affect, Confidence in national government, Democratic Quality, Delivery quality, Standard deviation of the ladder by country-year, GINI index, GINI index average 2000-15, gini of household income reported in Gallup. This list may change after further exploration of the data, as some features may prove less useful or highly correlated.
5. This is an important problem because predicting happiness scores allows us to identify the factors that have the greatest effect on well-being. Policymakers and government officials can use these insights to design targeted initiatives that improve quality of life. It also helps evaluate which current practices are effective and which areas need improvement.

1.3 Part 3: Understand Your Data

The next step is to perform exploratory data analysis. Inspect and analyze your data set with your machine learning problem in mind. Consider the following as you inspect your data:

1. What data preparation techniques would you like to use? These data preparation techniques may include:
   - addressing missingness, such as replacing missing values with means
   - finding and replacing outliers
   - renaming features and labels
   - finding and replacing outliers
   - performing feature engineering techniques such as one-hot encoding on categorical features
   - selecting appropriate features and removing irrelevant features
   - performing specific data cleaning and preprocessing techniques for an NLP problem
   - addressing class imbalance in your data sample to promote fair AI
2. What machine learning model (or models) you would like to use that is suitable for your predictive problem and data?
   - Are there other data preparation techniques that you will need to apply to build a balanced modeling data set for your problem and model? For example, will you need to scale your data?
3. How will you evaluate and improve the model's performance?
   - Are there specific evaluation metrics and methods that are appropriate for your model?

Think of the different techniques you have used to inspect and analyze your data in this course. These include using Pandas to apply data filters, using the Pandas `describe()` method to get insight into key statistics for each column, using the Pandas `dtypes` property to inspect the data type of each column, and using Matplotlib and Seaborn to detect outliers and visualize relationships between features and labels. If you are working on a classification problem, use techniques you have learned to determine if there is class imbalance.

Task: Use the techniques you have learned in this course to inspect and analyze your data. You can import additional packages that you have used in this course that you will need to perform this task.

Note: You can add code cells if needed by going to the Insert menu and clicking on Insert Cell Below in the drop-drown menu.

[7]:
```
# first 5 rows
display(df.head())

# list all columns, dtype, and non-null count
display(df.info())

# show basic statistics
display(df.describe(include='all'))
```

```python
# check for missing values
print("\nMissing Values per Column:")
print(df.isnull().sum())

# histograms of numeric features
df.hist(bins=20, figsize=(15, 10))
plt.tight_layout()
plt.show()

# boxplots to check outliers for main numeric features
plt.figure(figsize=(15, 8))
sns.boxplot(data=df[['Life Ladder', 'Log GDP per capita', 'Social␣
↪support','Healthy life expectancy at birth', 'Freedom to make life␣
↪choices']])
plt.title("Boxplots of Key Features")
plt.show()

# correlation heatmap
plt.figure(figsize=(14, 10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()

# cleaning data

# drop GINI columns due to too many missing values
df_clean = df.drop(columns=['GINI index (World Bank estimate)','GINI index␣
↪(World Bank estimate), average 2000-15'])

# fill missing values for remaining numeric columns with median
for col in df_clean.columns:
    if df_clean[col].dtype != 'object':
        df_clean[col].fillna(df_clean[col].median(), inplace=True)

# check if there are still missing values
print("Missing Values After Cleaning:")
print(df_clean.isnull().sum())

# verify shape and data
display(df_clean.info())
display(df_clean.describe())
```

```
       country  year  Life Ladder  Log GDP per capita  Social support  \
0  Afghanistan  2008     3.723590            7.168690        0.450662
1  Afghanistan  2009     4.401778            7.333790        0.552308
2  Afghanistan  2010     4.758381            7.386629        0.539075
3  Afghanistan  2011     3.831719            7.415019        0.521104
```

```
4  Afghanistan  2012     3.782938           7.517126       0.520637

   Healthy life expectancy at birth  Freedom to make life choices  Generosity  \
0                         49.209663                      0.718114    0.181819
1                         49.624432                      0.678896    0.203614
2                         50.008961                      0.600127    0.137630
3                         50.367298                      0.495901    0.175329
4                         50.709263                      0.530935    0.247159

   Perceptions of corruption  Positive affect  Negative affect  \
0                   0.881686         0.517637         0.258195
1                   0.850035         0.583926         0.237092
2                   0.706766         0.618265         0.275324
3                   0.731109         0.611387         0.267175
4                   0.775620         0.710385         0.267919

   Confidence in national government  Democratic Quality  Delivery Quality  \
0                           0.612072           -1.929690         -1.655084
1                           0.611545           -2.044093         -1.635025
2                           0.299357           -1.991810         -1.617176
3                           0.307386           -1.919018         -1.616221
4                           0.435440           -1.842996         -1.404078

   Standard deviation of ladder by country-year  \
0                                      1.774662
1                                      1.722688
2                                      1.878622
3                                      1.785360
4                                      1.798283

   Standard deviation/Mean of ladder by country-year  \
0                                           0.476600
1                                           0.391362
2                                           0.394803
3                                           0.465942
4                                           0.475367

   GINI index (World Bank estimate)  \
0                               NaN
1                               NaN
2                               NaN
3                               NaN
4                               NaN

   GINI index (World Bank estimate), average 2000-15  \
0                                                NaN
1                                                NaN
2                                                NaN
```

```
3                                               NaN
4                                               NaN

   gini of household income reported in Gallup, by wp5-year
0                                               NaN
1                                          0.441906
2                                          0.327318
3                                          0.336764
4                                          0.344540

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1562 entries, 0 to 1561
Data columns (total 19 columns):
 #   Column                                                    Non-Null Count
Dtype
---  ------                                                    --------------
-----
 0   country                                                   1562 non-null
object
 1   year                                                      1562 non-null
int64
 2   Life Ladder                                               1562 non-null
float64
 3   Log GDP per capita                                        1535 non-null
float64
 4   Social support                                            1549 non-null
float64
 5   Healthy life expectancy at birth                          1553 non-null
float64
 6   Freedom to make life choices                              1533 non-null
float64
 7   Generosity                                                1482 non-null
float64
 8   Perceptions of corruption                                 1472 non-null
float64
 9   Positive affect                                           1544 non-null
float64
 10  Negative affect                                           1550 non-null
float64
 11  Confidence in national government                         1401 non-null
float64
 12  Democratic Quality                                        1391 non-null
float64
 13  Delivery Quality                                          1391 non-null
float64
 14  Standard deviation of ladder by country-year              1562 non-null
float64
```

```
 15  Standard deviation/Mean of ladder by country-year         1562 non-null
float64
 16  GINI index (World Bank estimate)                          583 non-null
float64
 17  GINI index (World Bank estimate), average 2000-15         1386 non-null
float64
 18  gini of household income reported in Gallup, by wp5-year  1205 non-null
float64
dtypes: float64(17), int64(1), object(1)
memory usage: 232.0+ KB
```

```
None
```

```
         country         year  Life Ladder  Log GDP per capita  \
count       1562  1562.000000  1562.000000         1535.000000
unique       164          NaN          NaN                 NaN
top     Zimbabwe          NaN          NaN                 NaN
freq          12          NaN          NaN                 NaN
mean         NaN  2011.820743     5.433676            9.220822
std          NaN     3.419787     1.121017            1.184035
min          NaN  2005.000000     2.661718            6.377396
25%          NaN  2009.000000     4.606351            8.310665
50%          NaN  2012.000000     5.332600            9.398610
75%          NaN  2015.000000     6.271025           10.190634
max          NaN  2017.000000     8.018934           11.770276

        Social support  Healthy life expectancy at birth  \
count      1549.000000                       1553.000000
unique             NaN                               NaN
top                NaN                               NaN
freq               NaN                               NaN
mean          0.810669                         62.249887
std           0.119370                          7.960671
min           0.290184                         37.766476
25%           0.748304                         57.299580
50%           0.833047                         63.803192
75%           0.904329                         68.098228
max           0.987343                         76.536362

        Freedom to make life choices   Generosity  Perceptions of corruption  \
count                    1533.000000  1482.000000                1472.000000
unique                           NaN          NaN                        NaN
top                              NaN          NaN                        NaN
freq                             NaN          NaN                        NaN
mean                        0.728975     0.000079                   0.753622
std                         0.145408     0.164202                   0.185538
min                         0.257534    -0.322952                   0.035198
25%                         0.633754    -0.114313                   0.697359
```

50%	0.748014	-0.022638	0.808115
75%	0.843628	0.094649	0.880089
max	0.985178	0.677773	0.983276

	Positive affect	Negative affect	Confidence in national government \
count	1544.000000	1550.000000	1401.000000
unique	NaN	NaN	NaN
top	NaN	NaN	NaN
freq	NaN	NaN	NaN
mean	0.708969	0.263171	0.480207
std	0.107644	0.084006	0.190724
min	0.362498	0.083426	0.068769
25%	0.621471	0.204116	0.334732
50%	0.717398	0.251798	0.463137
75%	0.800858	0.311515	0.610723
max	0.943621	0.704590	0.993604

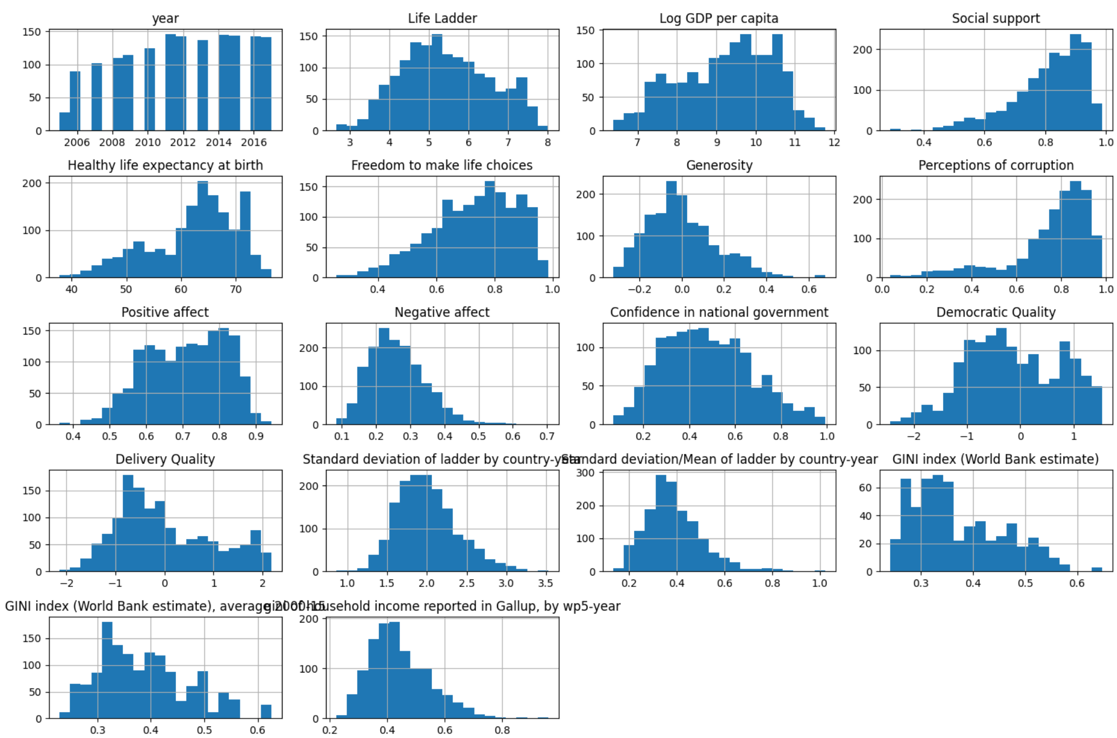
	Democratic Quality	Delivery Quality \
count	1391.000000	1391.000000
unique	NaN	NaN
top	NaN	NaN
freq	NaN	NaN
mean	-0.126617	0.004947
std	0.873259	0.981052
min	-2.448228	-2.144974
25%	-0.772010	-0.717463
50%	-0.225939	-0.210142
75%	0.665944	0.717996
max	1.540097	2.184725

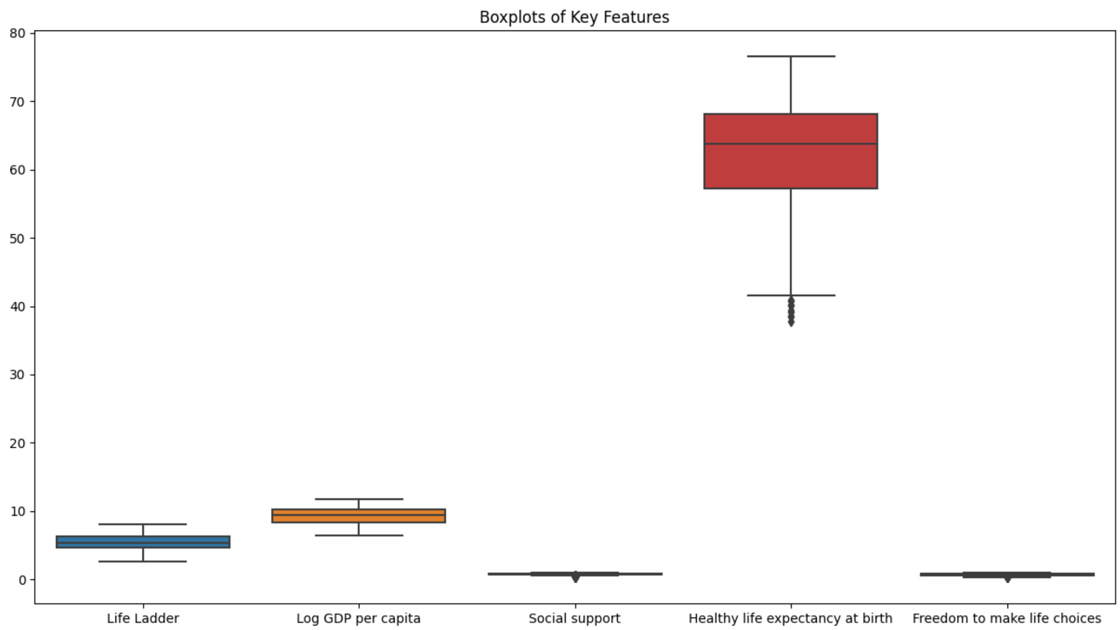
	Standard deviation of ladder by country-year \
count	1562.000000
unique	NaN
top	NaN
freq	NaN
mean	2.003501
std	0.379684
min	0.863034
25%	1.737934
50%	1.960345
75%	2.215920
max	3.527820

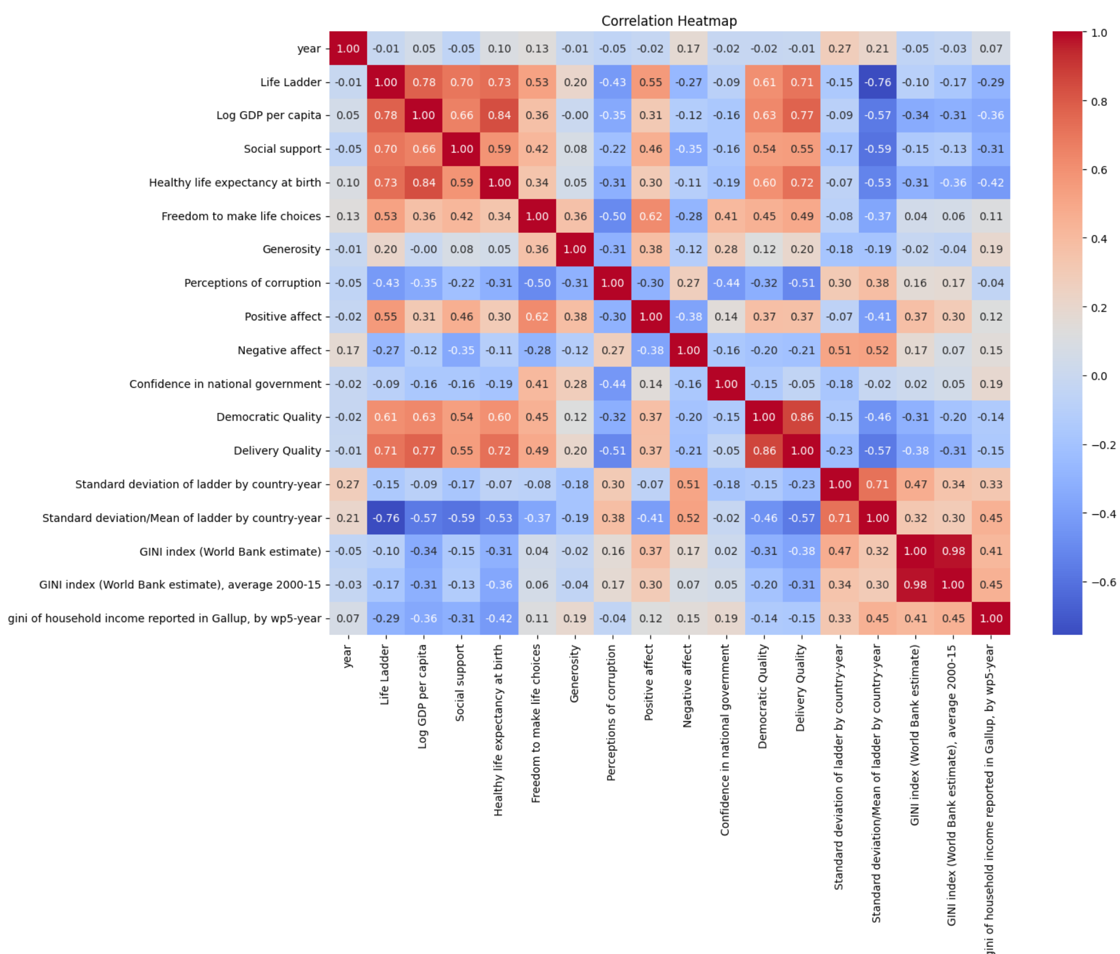
	Standard deviation/Mean of ladder by country-year \
count	1562.000000
unique	NaN
top	NaN
freq	NaN

```
mean                                            0.387271
std                                             0.119007
min                                             0.133908
25%                                             0.309722
50%                                             0.369751
75%                                             0.451833
max                                             1.022769

        GINI index (World Bank estimate)  \
count                         583.000000
unique                               NaN
top                                  NaN
freq                                 NaN
mean                            0.372846
std                             0.086609
min                             0.241000
25%                             0.307000
50%                             0.349000
75%                             0.433500
max                             0.648000

        GINI index (World Bank estimate), average 2000-15  \
count                                          1386.000000
unique                                                 NaN
top                                                    NaN
freq                                                   NaN
mean                                              0.386948
std                                               0.083694
min                                               0.228833
25%                                               0.321583
50%                                               0.371000
75%                                               0.433104
max                                               0.626000

        gini of household income reported in Gallup, by wp5-year
count                                          1205.000000
unique                                                 NaN
top                                                    NaN
freq                                                   NaN
mean                                              0.445204
std                                               0.105410
min                                               0.223470
25%                                               0.368531
50%                                               0.425395
75%                                               0.508579
max                                               0.961435
```

```
Missing Values per Column:
country                                                   0
year                                                      0
Life Ladder                                               0
Log GDP per capita                                       27
Social support                                           13
Healthy life expectancy at birth                          9
Freedom to make life choices                             29
Generosity                                               80
Perceptions of corruption                                90
Positive affect                                          18
Negative affect                                          12
Confidence in national government                       161
Democratic Quality                                      171
Delivery Quality                                        171
Standard deviation of ladder by country-year              0
Standard deviation/Mean of ladder by country-year         0
GINI index (World Bank estimate)                        979
GINI index (World Bank estimate), average 2000-15       176
gini of household income reported in Gallup, by wp5-year   357
dtype: int64
```




Boxplots of Key Features
80
70
60
50
40
30
20
10
0
Life Ladder
Log GDP per capita
Social support
Healthy life expectancy at birth
Freedom to make life choices


Correlation Heatmap
year
Life Ladder
Log GDP per capita
Social support
Healthy life expectancy at birth
Freedom to make life choices
Generosity
Perceptions of corruption
Positive affect
Negative affect
Confidence in national government
Democratic Quality
Delivery Quality
Standard deviation of ladder by country-year
Standard deviation/Mean of ladder by country-year
GINI index (World Bank estimate)
GINI index (World Bank estimate), average 2000-15
gini of household income reported in Gallup, by wp5-year

```
Missing Values After Cleaning:
country                                                   0
year                                                      0
Life Ladder                                               0
Log GDP per capita                                        0
Social support                                            0
Healthy life expectancy at birth                          0
Freedom to make life choices                              0
Generosity                                                0
Perceptions of corruption                                 0
Positive affect                                           0
Negative affect                                           0
Confidence in national government                         0
Democratic Quality                                        0
Delivery Quality                                          0
Standard deviation of ladder by country-year              0
Standard deviation/Mean of ladder by country-year         0
gini of household income reported in Gallup, by wp5-year  0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1562 entries, 0 to 1561
Data columns (total 17 columns):
 #   Column                                                    Non-Null Count
Dtype
---  ------                                                    --------------
-----
 0   country                                                   1562 non-null
object
 1   year                                                      1562 non-null
int64
 2   Life Ladder                                               1562 non-null
float64
 3   Log GDP per capita                                        1562 non-null
float64
 4   Social support                                            1562 non-null
float64
 5   Healthy life expectancy at birth                          1562 non-null
float64
 6   Freedom to make life choices                              1562 non-null
float64
 7   Generosity                                                1562 non-null
float64
 8   Perceptions of corruption                                 1562 non-null
float64
 9   Positive affect                                           1562 non-null
```

```
float64
 10  Negative affect                                           1562 non-null
float64
 11  Confidence in national government                         1562 non-null
float64
 12  Democratic Quality                                        1562 non-null
float64
 13  Delivery Quality                                          1562 non-null
float64
 14  Standard deviation of ladder by country-year              1562 non-null
float64
 15  Standard deviation/Mean of ladder by country-year         1562 non-null
float64
 16  gini of household income reported in Gallup, by wp5-year  1562 non-null
float64
dtypes: float64(15), int64(1), object(1)
memory usage: 207.6+ KB
```

```
None
```

```
              year  Life Ladder  Log GDP per capita  Social support  \
count  1562.000000  1562.000000         1562.000000     1562.000000
mean   2011.820743     5.433676            9.223895        0.810855
std       3.419787     1.121017            1.173979        0.118890
min    2005.000000     2.661718            6.377396        0.290184
25%    2009.000000     4.606351            8.330659        0.749794
50%    2012.000000     5.332600            9.398610        0.833047
75%    2015.000000     6.271025           10.167549        0.904097
max    2017.000000     8.018934           11.770276        0.987343

       Healthy life expectancy at birth  Freedom to make life choices  \
count                       1562.000000                   1562.000000
mean                          62.258837                      0.729328
std                            7.938561                      0.144074
min                           37.766476                      0.257534
25%                           57.344959                      0.635676
50%                           63.803192                      0.748014
75%                           68.064693                      0.841122
max                           76.536362                      0.985178

        Generosity  Perceptions of corruption  Positive affect  \
count  1562.000000                1562.000000      1562.000000
mean     -0.001084                   0.756762         0.709066
std       0.160017                   0.180558         0.107025
min      -0.322952                   0.035198         0.362498
25%      -0.108292                   0.702761         0.622581
50%      -0.022638                   0.808115         0.717398
75%       0.086098                   0.874675         0.799524
```

```
max       0.677773                   0.983276         0.943621

       Negative affect  Confidence in national government  Democratic Quality  \
count      1562.000000                        1562.000000         1562.000000
mean          0.263083                           0.478448           -0.137490
std           0.083688                           0.180696            0.824625
min           0.083426                           0.068769           -2.448228
25%           0.204680                           0.348685           -0.713479
50%           0.251798                           0.463137           -0.225939
75%           0.310713                           0.593869            0.504140
max           0.704590                           0.993604            1.540097

       Delivery Quality  Standard deviation of ladder by country-year  \
count       1562.000000                                   1562.000000
mean          -0.018600                                      2.003501
std            0.928193                                      0.379684
min           -2.144974                                      0.863034
25%           -0.671931                                      1.737934
50%           -0.210142                                      1.960345
75%            0.606049                                      2.215920
max            2.184725                                      3.527820

       Standard deviation/Mean of ladder by country-year  \
count                                        1562.000000
mean                                            0.387271
std                                             0.119007
min                                             0.133908
25%                                             0.309722
50%                                             0.369751
75%                                             0.451833
max                                             1.022769

       gini of household income reported in Gallup, by wp5-year
count                                        1562.000000
mean                                            0.440677
std                                             0.092948
min                                             0.223470
25%                                             0.386856
50%                                             0.425395
75%                                             0.480072
max                                             0.961435
```

1.4 Part 4: Define Your Project Plan

Now that you understand your data, in the markdown cell below, define your plan to implement the remaining phases of the machine learning life cycle (data preparation, modeling, evaluation) to solve your ML problem. Answer the following questions:

- Do you have a new feature list? If so, what are the features that you chose to keep and remove after inspecting the data?
- Explain different data preparation techniques that you will use to prepare your data for modeling.
- What is your model (or models)?
- Describe your plan to train your model, analyze its performance and then improve the model. That is, describe your model building, validation and selection plan to produce a model that generalizes well to new data.

1. The new feature list is: year, Log GDP per capita, Social support, Healthy life expectancy at birth, Freedom to make life choices, Generosity, Perceptions of corruption, Postive affect, Negative affect, Confidence in national government, Democratic Quality, Delivery quality, Standard deviation of the ladder by country-year, gini of household income reported in Gallup. The features I removed were GINI index and GINI index average 2000-15 due to the large amounts of missing values. I also removed country due to the large amount of unique values.
2. The different data preparation techniques I will use to prepare my data for modeling is spliting the data for testing, remove outliers, remove features that are invaluable at the moment, and handle missing values.
3. My model will begin as a simple Linear Regression model and then I will test more advanced models such as Random Forest Regressor and Gradient Boosting Regressor.
4. My plan to train my model is to begin by spliting the data into training and testing data. I will then evaluated the model using Mean Squared Error to see how it is performing. Lastly, I will be optimizing the model by trying different models and tuning the hyperparameters.

1.5 Part 5: Implement Your Project Plan

Task: In the code cell below, import additional packages that you have used in this course that you will need to implement your project plan.

[8]:
```
# different imports
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

Task: Use the rest of this notebook to carry out your project plan.

You will:

1. Prepare your data for your model.
2. Fit your model to the training data and evaluate your model.
3. Improve your model's performance by performing model selection and/or feature selection techniques to find best model for your problem.

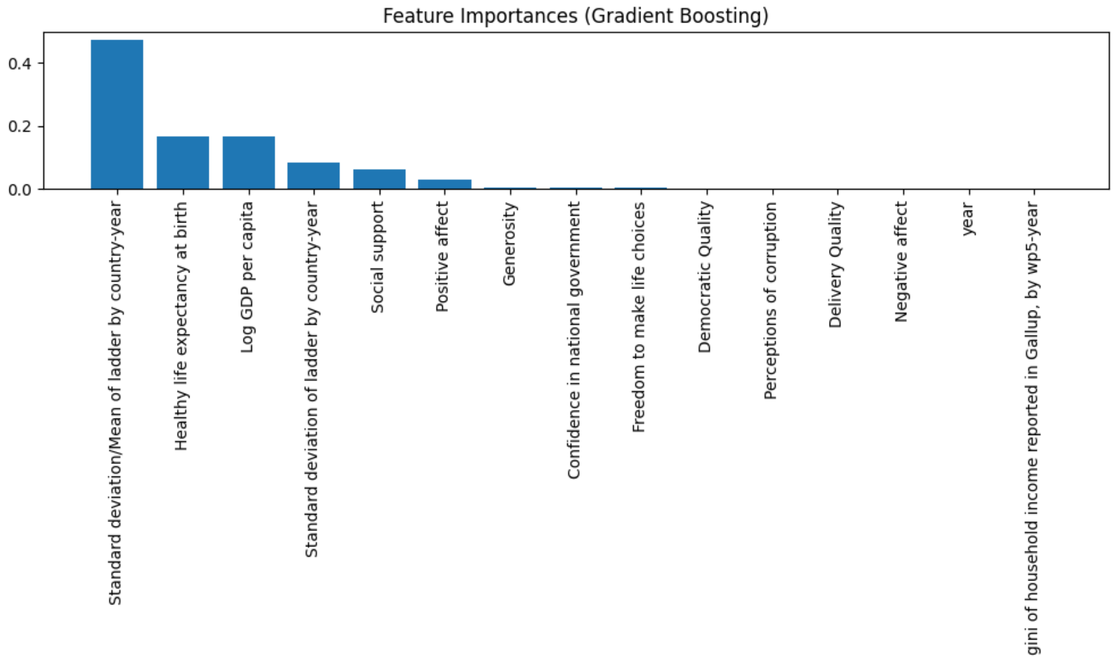
Add code cells below and populate the notebook with commentary, code, analyses, results, and figures as you see fit.

[10]:
```
# define features and target
X = df_clean.drop(columns=['Life Ladder', 'country'])  # drop non-numeric +␣
↪label
y = df_clean['Life Ladder']

# split 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)

# scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# train model
model = LinearRegression()
model.fit(X_train_scaled, y_train)

# predict
y_pred_model = model.predict(X_test_scaled)

# evaluate
mse_model = mean_squared_error(y_test, y_pred_model)
rmse_model = np.sqrt(mse_model)
r2_model = r2_score(y_test, y_pred_model)

print("Linear Regression Model Results:")
print(f"Mean Squared Error (MSE): {mse_model:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse_model:.4f}")
print(f"R^2 Score: {r2_model:.4f}")

# test other model to compare results
# train Random Forest model
rf_model = RandomForestRegressor(random_state=42)
rf_model.fit(X_train_scaled, y_train)

# predict
y_pred_rf = rf_model.predict(X_test_scaled)

# evaluate
mse_rf = mean_squared_error(y_test, y_pred_rf)
rmse_rf = np.sqrt(mse_rf)
r2_rf = r2_score(y_test, y_pred_rf)
```

```
print("Random Forest Regressor Results:")
print(f"Mean Squared Error (MSE): {mse_rf:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse_rf:.4f}")
print(f"R^2 Score: {r2_rf:.4f}")

# test another model to compare results
# train Gradient Boosting model
gb_model = GradientBoostingRegressor(random_state=42)
gb_model.fit(X_train_scaled, y_train)

# predict
y_pred_gb = gb_model.predict(X_test_scaled)

# evaluate
mse_gb = mean_squared_error(y_test, y_pred_gb)
rmse_gb = np.sqrt(mse_gb)
r2_gb = r2_score(y_test, y_pred_gb)

print("Gradient Boosting Regressor Results:")
print(f"Mean Squared Error (MSE): {mse_gb:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse_gb:.4f}")
print(f"R^2 Score: {r2_gb:.4f}")

# in the end Gradient Boosting Regressor performed the best thus this will be␣
↪the one to be chosen

# see features with most importance
feature_names = X.columns
importances = gb_model.feature_importances_
indices = np.argsort(importances)[::-1]

plt.figure(figsize=(10, 6))
plt.title("Feature Importances (Gradient Boosting)")
plt.bar(range(X.shape[1]), importances[indices], align="center")
plt.xticks(range(X.shape[1]), feature_names[indices], rotation=90)
plt.tight_layout()
plt.show()

# Standard deviation/mean of ladder by country-year was the strongest predictor.
↪ We can interpret that countryies with more stabile scores have more␣
↪accurate scores.

# tuning some hyperparameters
# trying different combinations
models_to_try = [
    GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, max_depth=3,␣
↪random_state=42),
```

```
    GradientBoostingRegressor(n_estimators=200, learning_rate=0.05,␣
↪max_depth=4, random_state=42),
    GradientBoostingRegressor(n_estimators=300, learning_rate=0.01,␣
↪max_depth=5, random_state=42)
]

for i, model in enumerate(models_to_try, start=1):
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test, y_pred)
    print(f"\nModel {i} Results:")
    print(f"MSE: {mse:.4f}, RMSE: {rmse:.4f}, R^2 Score: {r2:.4f}")

# In the end, Model 2 had the best performance when comparing all the results.␣
↪When changing n_estimators=200, learning_rate=0.05, and max_depth=4 had the␣
↪best results.
```

```
Linear Regression Model Results:
Mean Squared Error (MSE): 0.0815
Root Mean Squared Error (RMSE): 0.2854
R^2 Score: 0.9370
Random Forest Regressor Results:
Mean Squared Error (MSE): 0.0485
Root Mean Squared Error (RMSE): 0.2203
R^2 Score: 0.9625
Gradient Boosting Regressor Results:
Mean Squared Error (MSE): 0.0359
Root Mean Squared Error (RMSE): 0.1894
R^2 Score: 0.9723
```




```
Model 1 Results:
MSE: 0.0359, RMSE: 0.1894, R^2 Score: 0.9723

Model 2 Results:
MSE: 0.0272, RMSE: 0.1650, R^2 Score: 0.9789

Model 3 Results:
MSE: 0.0505, RMSE: 0.2248, R^2 Score: 0.9609
```

[ ]: