**Data Loading and Preprocessing**

* **Pandas for Data Handling:**
  + pd.read\_csv: Reads data from CSV files into Pandas DataFrames.
  + Reference: [Pandas read\_csv documentation](https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html)
* **Handling Missing Values (Imputation):**
  + SimpleImputer: Replaces missing values in numerical columns with the median and missing values in categorical columns with the most frequent value.
  + Reference: [Scikit-learn SimpleImputer documentation](https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html)
* **Encoding Categorical Features:**
  + OneHotEncoder: Converts categorical features into a one-hot encoded format (binary columns).
  + LabelEncoder: Converts categorical labels into numerical representation.
  + Reference: [Scikit-learn OneHotEncoder documentation](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html) and [Scikit-learn LabelEncoder documentation](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html)
* **Scaling Numerical Features:**
  + StandardScaler: Standardizes numerical features (zero mean and unit variance).
  + Reference: [Scikit-learn StandardScaler documentation](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html)
* **ColumnTransformer:**
  + Applies different transformers to different columns.
  + Reference: [Scikit-learn ColumnTransformer documentation](https://scikit-learn.org/stable/modules/generated/sklearn.compose.ColumnTransformer.html)

**Machine Learning Models**

* **Multiple Linear Regression:**
  + LinearRegression: Fits a linear model with coefficients w = (w1, ..., wp) to minimize the residual sum of squares between the observed targets in the dataset and the targets predicted by the linear approximation.
  + Reference: [Scikit-learn LinearRegression documentation](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html)
* **Random Forest Regression:**
  + RandomForestRegressor: A meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
  + Reference: [Scikit-learn RandomForestRegressor documentation](%3C3%3Ehttps:/scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html)
* **Gradient Boosting Regression:**
  + GradientBoostingRegressor: Builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions.
  + Reference: [Scikit-learn GradientBoostingRegressor documentation](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html)

**Model Evaluation**

* **R-squared:**
  + r2\_score: Coefficient of determination; measures the proportion of the variance in the dependent variable that is predictable from the independent variable(s).
  + Reference: [Scikit-learn r2\_score documentation](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html)
* **Mean Squared Error:**
  + mean\_squared\_error: The mean of the squared errors; measures the average squared difference between the estimated values and the actual value.
  + Reference: [Scikit-learn mean\_squared\_error documentation](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html)

**Graphical User Interface (GUI)**

* **Tkinter:** The standard Python interface to the Tk GUI toolkit.
* **ttk (Themed Tk):** A set of themed widgets for Tkinter.
* **Matplotlib:** A comprehensive library for creating static, animated, and interactive visualizations in Python.
* **fstring:** (or 'fstring-core') Improved string formatting syntax in Python.