**Project Instructions**

**Project Overview**

* **Tasks**:
  + Regression Task
  + Classification Task
* **Dataset Requirements**:
  + Find separate datasets for each task.
  + Use datasets that haven’t been utilized in previous labs.

**Regression Task**

1. **Potential Topics**:
   * Predicting Housing Prices
     + Example Datasets:
       - California Housing Data Set
       - Sydney Housing Data Set
       - Perth Housing Data Set
   * Predicting Weather in Dublin
2. **Data Preparation**:
   * **Reading Data**: Load the CSV file into a DataFrame.
   * **Handle Missing Data**:
     + Check for null values: cal\_df.isnull().sum()
     + Impute missing values if necessary, e.g., cal\_df.total\_bedrooms.fillna(cal\_df.total\_bedrooms.mean())
     + If there are too many missing values in columns/rows, consider dropping them.
       - Dropping columns: cal\_df.drop(['Col\_name'], axis=1)
   * **Categorical Predictors**:
     + Ignore or encode categorical variables.
     + Example: Use LabelEncoder:
       - python

Copy code

from sklearn.preprocessing import LabelEncoder

LE = LabelEncoder()

cal\_df['ocean\_proximity'] = LE.fit\_transform(cal\_df['ocean\_proximity'])

* + **Standardize Data**:
    - Ensure features are in the same unit:
      * python

Copy code

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaled\_df = scaler.fit\_transform(cal\_df)

scaled\_df = pd.DataFrame(scaled\_df, columns=names)

* + **Statistics and Visualizations**:
    - Produce summary statistics and plots (boxplots, mean, variance, standard deviation).
  + **Data Splitting**:
    - Split data into training and testing sets:
      * python

Copy code

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

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=1)

* + **Model Building**:
    - Train a Linear Regression model on the training data.
    - Evaluate using metrics (e.g., MAE, MSE, R-squared).

1. **Further Analysis**:
   * Select the most informative features (e.g., top 4).
   * Repeat cleaning, modeling, and evaluation with selected features.

**Classification Task**

1. **Potential Topics**:
   * Heart Disease Prediction (yes/no)
   * Cancer Prediction (yes/no)
2. **Data Preparation**:
   * Follow similar preprocessing steps as in the Regression Task.
   * Experiment with different parameters:
     + Vary K values (e.g., from 1 to 100).
     + Try different distance metrics (Euclidean, Manhattan).
     + Use uniform and distance-weighted methods.
3. **Classification Models**:
   * Use both **K-Nearest Neighbors (K-NN)** and **Logistic Regression**.
   * Ensure the dataset has a substantial number of instances (10,000 or close).
   * Feature Requirements:
     + Include as many predictors as possible (ideally 9-10, minimum 3).

**Report Guidelines**

1. **Structure**:
   * **Abstract**: High-level summary of the project.
   * **Introduction**:
     + Overview of both tasks (regression and classification) and datasets.
     + Key findings and motivation for choosing the problem areas.
   * **Data Statistics**:
     + Describe datasets, including mean, SD, quartiles, and missing values.
     + Explain any preprocessing steps, including feature removal.
   * **Methodology**:
     + Overview of applied algorithms (Linear Regression, K-NN, Logistic Regression).
     + Explain data scaling (normalization or standardization).
     + Outline chosen hyperparameters (e.g., K values, distance metrics).
   * **Results and Evaluation**:
     + Detail evaluation methods and results.
     + Include metrics (e.g., R-Squared, MAE, MSE, accuracy, confusion matrix).
     + Discuss key findings.
   * **Error Analysis**:
     + Identify poorly performing instances and any observed patterns.
   * **Conclusion**:
     + Summarize findings and discuss improvements for future work.
2. **Submission**:
   * Upload the final report to Turnitin.
   * Include a zip file with:
     + Regression Dataset
     + Classification Dataset
     + Code files (can be multiple Jupyter Notebooks)

**Additional Support and Tips**

* **Focus on Python Basics**:
  + If behind, review lab sessions on basic syntax, libraries, and simple models.
  + Key resources include Introduction to Numpy, Introduction to Pandas, and Introduction to Scikit-Learn.
* **Practice with Lab Code**:
  + Labs contain reusable code snippets for:
    - **Linear Regression**: California Housing data with basic cleaning and model evaluation.
    - **K-NN Classification**: Cancer dataset with model evaluation and hyperparameter tuning.
    - **Logistic Regression**: Toy datasets (e.g., Iris, Digit, and Admission data).