Project

Tasks

- There are 2 tasks
  - Regression task
  - Classification task

Datasets

- Need to find a dataset for each task
- Both datasets should be a dataset not used in the Lab

Regression Task

Choose a regression problem

- Predicting Housing Prices
- Predicting Weather in Dublin
- …

Choose a data set

- For “Predicting of Housing Prices”
  - E.g.
    - California Housing Data set
    - Sydney Housing Data Set
    - Perth Housing Data set

Read data file

- Reading CSV File into a DataFrame
- Checking for Null Values (isnull)
  - cal_df.isnull().sum()
- Do imputation if any null values
  - cal_df.total_bedrooms=cal_df.total_bedrooms.fillna(cal_df.total_bedrooms.mean())
- If too many problem in cols /rows; remove entire columns/rows
- Dropping columns
  - cal_df.drop([‘Col_name’], axis =1)

Categorical

- Check if any predictor (column) categorical?
- You may not consider Categorical predictor
- Or You can convert that into continuous predictor
  - LE = LabelEncoder()
  - cal_df['ocean_proximity']=LE.fit_transform(cal_df['ocean_proximity'])

Standardize your data

- If not in same unit
- Example (California Housing Data) [already shown in Lab 7]
  - # Get column names first of your data frame
  - **names = cal_df.columns**
  - # Create the Scaler object
  - **scaler = StandardScaler()**
  - # Fit your data on the scaler object
  - **scaled_df = scaler.fit_transform(cal_df)**
  - **scaled_df = pd.DataFrame(scaled_df, columns=names)**
  - **scaled_df.head()**

Data Statistics

- Produce some stat and plots
- Boxplots
- Mean, variance, SD
- Number of instances etc.
- Number of missing values

Training and Test data

- Split into train and test
    - X_train,X_test,Y_train,Y_test = train_test_split(X,y,test_size=0.20, random_state=1)
    - print(X_train.shape)
    - print(Y_train.shape)
    - print(X_test.shape)
    - print(Y_test.shape)
- Build your **Linear regression model** on Train data
- Measure performance of LR model on the test data
    - MAE,
    - MSE,
    - R-Squared

- See how parameters were learnt in training
  - Values of Beta0, Beta1 etc.
  - Already shown (reuse code if needed)

Further Training

- Say you have 8 predictors in California Housing data set
- Choose 4 most informative features (look at the data set and decide)
- You can perform same set of experiments (e.g. cleaning, building and evaluation)
- You can reuse codes that were used in LAB

Classification Task

Classification Task

- Choose a problem
  - Heart Disease prediction (yes/no)
  - Cancer Prediction (yes/no)
  - …
- Choose a data set

- Same preprocessing as in Regression Task
- You can reuse codes that were used in LAB

K-NN

- Play with different parameters
  - Different values of k (1 to 100)
  - Distance Metrics (Euclidean, Manhattan etc.)
  - Distance (uniform, distance weighted)
- You can reuse codes that were used in LAB

Logistic Regression

- Use the template codes that has been used in LAB
- Your classification data is the same for both classification tasks, i.e.
  - K-NN
  - Logistic Regression

Size of Data

- As big as possible
  - Say 10,000 instances
  - If they are close to 10K, that's fine
  - If you find say one that has say 5,000 instances or less, send an **email to me**
  - We will talk and sort out
- Number of predictors
  - As many as possible
  - 9-10
  - Say >=3

Summary

- Regression Task
  - Choose a problem (of your interest)
  - Choose a data (of your interest)
  - If predictors are not in same unit, Normalize or Standardize your data
  - Split the data into train and test sets
  - Build your Regression Model on train data, evaluate it on test data
  - Use **the template code file** that was used in LAB
- Classification Task
  - Choose a problem (of your interest)
  - Choose a data (of your interest)
  - If predictors are not in same unit, Normalize or Standardize your data
  - Split the data into train and test sets
  - Build your two Classification Models on train data (K-NN and Logistic Regression)
  - Evaluate them on test data
  - Use **the template code files** that have been used in LAB

Upload

- Turnitin
  - A **report** describing both tasks
    - See the Moodle page
  - A **zip file**
    - Regression Data Set
    - Classification Data Set
    - Codes [could be one or multiple Jupyter Notebook files (ipynb)]

Report Writing

- Sections
  - Abstract
    - A high-level overview of your project
  - Introduction
    - A high-level overview of two tasks (classification) and data sets; outlining key findings
  - Motivation
    - A few lines on why you chose to work on these problems
  - Data Statistics
    - Describe your data sets (stats, e.g mean, SD, quartile (Q1, Q2, Q3), IQR)
    - Boxplots etc.
    - What preprocessing you applied (e.g. did you remove a predictor? and why?)
  - Methodology
    - Describe your methods
      - Give high-level overview of the Algorithms (Linear regression, Knn, logistic regression) you applied
      - For scaling your data, did you use normalization or standardization or both. Give overview of the methods.
      - As for your training, did you choose different set hyperparameters (k, distance metrics)? Explain them how.
  - Results and Evaluation
    - Describe your evaluation methods
    - Describe your results
      - R-Squared, MAE, MSE
      - Accuracy, Confusion Matrix
    - Discuss your findings.
  - Error Analysis
    - Identify those instances for which your model performed poorly
    - Is there any pattern that you found in them?
  - Conclusion
    - Conclude your work
      - What about this report (project/tasks) and Key findings
    - In future, if you are given more time, how you can improve your tasks.

Python

- From first week
  - My emphasis was on **Learning Python** during LAB
  - Basic Syntax and some libraries that are needed for your project
- Python – much more simple language (if you compare this with Java)

Python

- If you are still Behind, go through
  - The Lab Sessions
- Learning Syntax
  - **Getting Started with Python**
  - Learning Programming in Python Set 1.ipynb
  - Learning Programming in Python Set 2.ipynb
  - Lab_IAIML_w1.pdf

Python

- If you are still Behind, go through
  - The Lab Sessions
- Learning Syntax
  - 12 programming assignments

Python

- If you are still Behind, go through
  - The Lab Sessions
- Learning Syntax
  - Ungraded assignments
  - 14 Small programmes

Python

- If you are still Behind, go through
  - The Lab Sessions
- Learning Syntax
  - Python program to handle text data
  - Solutions

Python

- If you are still behind, go through
  - The Lab Sessions
- Learning Syntax
  - Introduction to Numpy and Pandas (Similar concept of arrays in Java)
    - Introduction to NumPy with Exercises.ipynb
    - Introduction to Pandas with Exercises.ipynb
    - Series and Dataframe
  - Demo: Titanic data is used to make you familiar with Numpy and Pandas
  - Car data set
    - Understanding Characteristics of descriptive features
    - Producing stats from data (Mean, median, SD, IQR)
    - Visualization (boxplots)
    - Hardly 10 lines of code

Python

- If you still Behind, go through
  - The Lab Sessions
- Learning Syntax
  - Introduction to Scikit-Learn
    - How to read a data
    - How to handle features and target variables
    - Simple prediction models

Python

- If you still Behind, go through
  - The Lab Sessions
- Linear Regression
  - California Housing data
    - How to read a data
    - Basic Cleaning Methods,
    - Splitting data, Model building, evaluation
    - Handling categorical data
    - Hardly 20 lines of codes.
  - **You can reuse the code for the project**

Python

- If you still Behind, go through
  - The Lab Sessions
- KNN Classification
  - Cancer data
    - How to read a data
    - Splitting data, Model building, evaluation
    - Using different Hyperparameters (k, distance weighting etc.)
    - Again, hardly 20 lines of codes.
  - **You can reuse the code for the project**

Python

- If you still Behind, go through
  - The Lab Sessions
- Logistic Regression Classification
  - Digit Data (Toy)
    - How to read a data
    - Splitting data, Model building, evaluation [all default params of Scikit-learn used]
    - Hardly 10 lines of codes.
  - Iris Data (Toy)
    - Same as above
- Admission Data
  - **You can reuse the code for the project**