CS F320 (Foundations of Data Science) – Draft

(subject to a few changes in the description of problems)

Lab Problems

Instructions

1. You are allowed to use sklearn only for PCA and scaling operations.
2. You can use pandas, numpy, SciPy, and matplotlib for data manipulation and visualization.
3. Document all your steps clearly in your code with comments. Include plots wherever necessary to support your analysis.
4. You can access the dataset for these lab problems through this link: https://drive.google.com/drive/folders/1jphk1rq2yPXZKebSXZLKVMGTl25NYiMP?usp=drive_link

---

P1: KL Divergence Calculation

The objective of this task is to **code a function to calculate KL Divergence** between two discrete probability distributions, measuring how one diverges from the other.

Problem Statement:

You are given the following probability distributions for a **random variable X**:

X (Random Variable)	1	2	3	4	5
P(X) (p.d.f_1)	0.1	0.3	0.4	0.1	0.1
Q(X) (p.d.f_2)	0.2	0.2	0.3	0.2	0.1

Your task is to:

1. **Write a function** to calculate the KL Divergence between the given distributions.
2. **Allow the user to input their own distributions** for comparison if they wish.

Instructions:

1. **KL Divergence Formula:**

$$D_{KL}(P||Q) = \sum_i P(i)\log\frac{P(i)}{Q(i)}$$

   Note: If *P(i)=0*, skip that term to avoid log(0) errors.

2. **Steps to Implement:**
   - Write a function named kl_divergence() to compute the KL Divergence using the formula above.
   - Use the given **P(X)** and **Q(X)** distributions to calculate the KL Divergence.
   - **Validate** that all user inputs (if provided) are valid probability distributions.
   - Display the **KL Divergence score** rounded to 4 decimal places.

P2: Feature Selection 2 (Greedy Feature Selection Methods)

The goal is to **predict home sale prices** using the optimal subset of features from the given dataset. Two feature selection methods will be applied:

1. **Greedy Forward Selection**
2. **Greedy Backward Selection**

Instructions:

1. **Data Preprocessing:**
    - Load the dataset.
    - Handle missing or inconsistent data.
    - Apply standardization or Min-Max scaling on the features.
    - Perform an **80:20** or **90:10** train-test split.
2. **Greedy Forward Selection:**
    - Start with an empty model.
    - Add features one-by-one, selecting the feature that gives the best improvement in model performance at each step.
    - Stop when no further improvement is observed or all features are added.
3. **Greedy Backward Selection:**
    - Start with all features in the model.
    - Remove features one-by-one, selecting the feature whose removal leads to the smallest decrease in performance.
    - Stop when further removal decreases performance significantly.

Input:

House_Price_Prediction.csv

Output:

- **Forward Selection:** List of selected features, feature count, and training/testing error.
- **Backward Selection:** List of selected features, feature count, and training/testing error.

P3: Feature Selection 2 (Spearman Correlation Coefficient Method)

Use the dataset provided in **Part 3** to build regression models for predicting housing prices. Find the optimal subset of features by selecting those with the highest Spearman correlation with the target attribute (price). Select feature sets based on the **Spearman Correlation Coefficient** to identify features most correlated with the target attribute (price). Use these selected features to train regression models and evaluate their predictive performance.

Input Format:

- Dataset from **Part 3** (containing attributes such as price, bedrooms, and other property characteristics).

Instructions:

1. **Compute Spearman Correlation:**
    - Calculate the Spearman correlation coefficient between each feature and the target (price).
    - Create feature sets of sizes 1, 2, 3, ..., n (where n is the total number of features) based on their correlation ranks.
2. **Build Regression Models:**
    - Train regression models for each feature subset identified.
    - Record the **training and testing errors** for every model.
3. **Comparative Analysis:**
    - Compare the performance of models built using:
        - Greedy Forward Feature Selection
        - Greedy Backward Feature Selection
        - Spearman Correlation Coefficient Selection
        - Model with all features included

Output Format:

- **Spearman Feature Selection:** Display selected features, feature count, and the corresponding training and testing errors.

- **Comparison Table:**
    - Present a table showing the training and testing errors for:
        - Greedy Forward Feature Selection
        - Greedy Backward Feature Selection
        - Spearman Correlation Coefficient Selection

    - Model with all features

Conclusion:

- Identify the feature selection method that yields the best performance.
- Discuss trade-offs, such as using fewer features vs. better predictive accuracy.
- Provide recommendations for future models based on your analysis.

Part 4: Prior and Posterior Distributions

A study was conducted to assess public opinion on a new smartphone brand. Let ‘p’ denote the probability of a person liking the smartphone. Before the product launch, market analysts assumed that ‘p’ follows a beta distribution with parameters $\alpha, \beta = (3, 5)$. After the initial survey, 70 out of 100 respondents stated they liked the smartphone. Plot the prior and posterior probability distribution of ‘p.’

The following day, a second survey was conducted, where out of the 70 respondents who liked the smartphone, 40 said they would not recommend it. Plot the posterior distribution of ‘p’ after this survey.

Input Format:

- $\alpha, \beta = (3, 5)$
- First Survey: Like - 70, Total - 100 respondents
- Second Survey: Dislike - 40, Total - 70 respondents

Output Format:

- Plot for the prior distribution of ‘p’.
- Plot for the posterior distribution of ‘p’ after the first survey.
- Plot for the posterior distribution of ‘p’ after the second survey.
- Explain how the posterior distribution of ‘p’ was derived in both cases.

Part 5: Entropy

Implement an entropy-based feature selection algorithm for a dataset with n features and a target variable. The objective is to identify the k most informative features that maximize mutual information with the target variable. Write a Python function feature_selection(X, y, k) that takes as inputs the feature set X (a mD array), the target variable y (a 1D array of binary labels), and an integer k, and returns the indices of the k most informative features.The mutual information I(X, Y) between a feature and the target variable is given by

I(X, Y) = H(X) + H(Y) - H(X, Y)

where H(X) is the entropy of the feature, H(Y) is the entropy of the target variable, and H(X, Y) is their joint entropy.

Formula to be used in this question : (Log is with base 2)

$$H[X] = -\sum_{i=1}^{n} P(x_i) \log P(x_i)$$

where X is a n-dimensional vector

$$\mathrm{H(X,Y)} = \sum_{\mathrm{x,y}} \mathrm{p(x,y)\ log\ \frac{1}{p(x,y)}}.$$

p(x, y) is the joint probability.

Input Format:

The 1st line contains three integers n, m and k. n is the number of data points, m is the number of features and k is the number of required features. The next n lines contain m integers each separated by a space. This is the input matrix X. Each column of this matrix is a feature of the data point. The next line contains n integers separated by a space. This is the target matrix y.

Output Format:

A list of indexes of the k selected columns(1-based indexing) separated by space.

Part 6: Outlier Detection

Using the dataset provided, create boxplots for each feature to visually analyze the distribution and detect potential outliers. Based on the boxplot insights, apply the Interquartile Range (IQR) method to identify and remove outliers from each feature. For the IQR method, consider data points as outliers if they fall below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR. Document your process, showing the before-and-after results of your dataset.

Instructions:

1. **Data Import and Cleaning:**
    - Import the dataset **german_credit_risk.csv**
    - Drop irrelevant columns (if any) and handle missing or null values.
    - Display the first few rows and provide a summary using .info() and .describe()
2. **Preprocess the dataset**
3. **Make boxplots of every feature before and after the outlier removal.**

P7: Principal Component Analysis (PCA) on Gym Members' Data

The goal of this problem is to perform **PCA** on the gym members' dataset to reduce its dimensionality, extract meaningful patterns, and understand relationships between member attributes while retaining the majority of the variability. **Follow the following steps:**

1. **Data Import and Cleaning:**
    - Import the dataset **gym_customers.csv** containing information about gym members.
    - Drop irrelevant columns (e.g., name_personal_trainer) and handle missing or null values.
    - Display the first few rows and provide a summary using .info() and .describe().

2. **Data Scaling and Visualization:**
    - **Feature Encoding:**
        - Convert categorical variables like gender and abonement_type into numerical values using **one-hot encoding** or **label encoding**.
    - **Data Scaling:**
        - Use **StandardScaler** to scale the dataset, ensuring features are standardized with mean 0 and standard deviation 1.
    - **Visualize Relationships:**
        - Use **pair plots** and **correlation heatmaps** to visualize relationships between key features like visit_per_week, avg_time_in_gym, and abonement_type.

3. **PCA Application:**
    - **Apply PCA:**
        - Use **sklearn's PCA** to reduce dimensionality.
        - Identify how many principal components are required to cover **80-90% of the variance**.
    - **Scree Plot and Cumulative Variance Plot:**
        - Create a **scree plot** to show the explained variance ratio for each component.
        - Plot the **cumulative variance** to visualize how many components are needed to cover most of the variability.

4. **Visualization of PCA Results:**
    - **2D Scatter Plot:**

        - Visualize the first two principal components using a **2D scatter plot**.
        - Use colors or shapes to differentiate between abonement_type (e.g., Premium vs. Standard).
    - **Loadings Plot:**
        - Add **loadings vectors** to your 2D scatter plot to show how original features contribute to the principal components.
    - **Interpretation:**
        - Explain what the first two principal components represent (e.g., attendance behavior or time spent in the gym).
        - Discuss any observed patterns or groups in the PCA results.

5. **Conclusion and Analysis:**

    - **Discuss Findings:**
        - Interpret the principal components and their importance in explaining the variability of the dataset.
        - Comment on any interesting patterns or relationships observed in the gym members' data (e.g., how different membership types cluster or how visit frequency influences PCA).

Input:

gym_customers.csv

Output:

1. **Scree Plot** and **Cumulative Variance Plot** showing explained variance by components.
2. **2D Scatter Plot** of the first two principal components with **loading vectors**.
3. Insights from PCA results and interpretation of component meanings.

Part 8: Multivariate Regression, PCA Analysis

The objective of this part is to create predictive models for **flight fare prediction** using **multiple linear regression, Principal Component Analysis (PCA)** on the flight_data.csv dataset.

Steps:

1. Exploratory Data Analysis (EDA):

- **Load the dataset:**
  Import **flight_data.csv** and explore the structure, features, and distributions.
- **Data Cleaning:**
  - Convert columns like flight date, dep_time, and arr_time into appropriate data types.
  - Handle **NULL values** by either filling or removing rows with missing data.
  - Drop any unnecessary columns (e.g., flight numbers if not predictive).
- **Visualization:**
  - Use **pair plots** and **correlation matrices** to explore relationships between features such as duration, price, and stops.
  - Provide observations on trends (e.g., longer flights tend to have higher prices).

2. Multiple Linear Regression:

- **Prepare the Data:**
  - Identify predictor variables (e.g., duration, stops) and the target variable (price).
  - Use **one-hot encoding** to convert categorical columns like airline or class into numerical values.

- **Train-Test Split:**
  - Split the dataset into **80% training** and **20% testing** data.
- **Implement Multiple Linear Regression:**
  - Develop a **linear regression model** using the training set.
  - **Evaluate the model's performance** using metrics like **R^2** and **Mean Absolute Error (MAE)**.
  - Visualize the predictions vs. actual prices on a scatter plot.

3. PCA Analysis:

- **Apply PCA** on the predictor variables to **reduce dimensionality**.
- **Determine the optimal number of components** by covering **85-90% of the total variance**.
- **Transform the dataset** using the selected principal components.
- **Visualize PCA Results:**

- Plot a **scree plot** and **cumulative variance plot** to show the explained variance by components.
- Provide insights on feature contributions using **loading vectors**.

4. Multivariate Regression with PCA:

- Develop a **multivariate regression model** using the transformed dataset with selected principal components.
- **Train and evaluate** the model using the same metrics (R^2, MAE) for comparison with the previous regression model.
- Visualize the PCA-based regression predictions against actual prices.

5. Comparison and Analysis:

- **Compare the performances** of:
  - Multiple linear regression without PCA
  - PCA-based regression
- Use visualizations (e.g., bar plots or residual plots) to highlight performance differences across models.
- Discuss the **impact of PCA** on model performance.

6. Conclusion and Recommendations:

- **Summarize key findings** from the analysis.
- **Recommend the best approach** for fare prediction based on model comparisons.
- Suggest possible improvements (e.g., adding external factors like weather) to enhance model accuracy.

Input:

Flight_data.csv

Output:

1. Scree plot and cumulative variance plot for PCA.
2. Scatter plots of predictions vs. actual prices for all models.
3. Bar chart comparing performance metrics (R^2, MAE) of the models.
4. Insights and recommendations based on the comparisons.
