CS F320 (Foundation of Data Science)

Lab Problems

Instructions

1. You are allowed to use sklearn for the first two questions only.
2. You can use pandas, numpy, SciPy and matplotlib for all problems.

1. Principal Component Analysis (PCA) on Pokemon Dataset

The objective of this question is to apply Principal Component Analysis (PCA) on the Pokemon dataset to reduce dimensionality while retaining the majority of variability in the data.

Steps:

1. **Data Import and Cleaning:**
   - Import the dataset 'Pokemon.csv' containing information about different Pokemon.
   - Clean the dataset by handling unwanted columns and removing rows with NULL values.

2. **Data Scaling and Visualization:**
   - Scale the data appropriately using normalization techniques to ensure features are on a similar scale.
   - Utilize data visualization techniques, including pair plots and other graphs, to gain insights into the variation present in the dataset. Visualize relationships between different attributes.

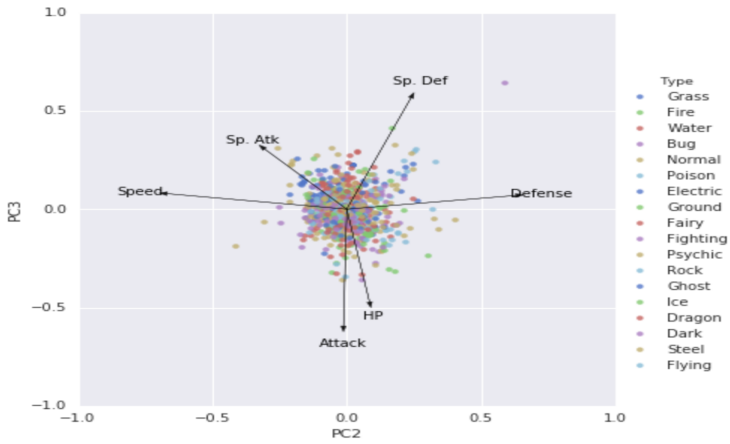
3. **PCA Application:**
   - Implement PCA on the cleaned and scaled dataset.
   - Choose the number of principal components that cover 80-90% of the total variability in the dataset.

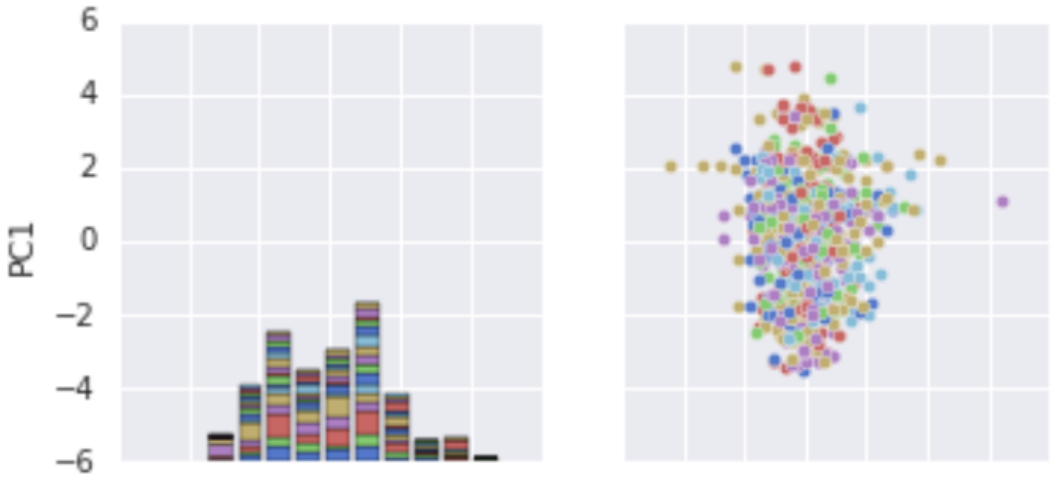
4. **Visualization of PCA Results:**
   - Plot pair plots and appropriate graphs showcasing principal components and their relationships with each other.
   - Represent different Pokemon types with distinct colors on the plots to distinguish between them.
   - Display the principal components on the graph as vectors to demonstrate their directions and importance in explaining the variability in the data.relationships with each other.
   - Represent different Pokemon types with distinct colors on the plots to distinguish between them.
   - Display the principal components on the graph as vectors to demonstrate their directions and importance in explaining the variability in the data.

5. **Conclusion and Analysis:**
   - Discuss the findings from the PCA analysis.
   - Interpret the principal components and their significance in explaining the variability in the Pokemon dataset.
   - Comment on any patterns or relationships observed among different Pokemon types based on the PCA results.

6. **Documentation and Presentation:**
   - Provide a well-documented report detailing the steps performed, observations, and conclusions drawn from the analysis.
   - Include visual representations, such as graphs and pair plots, supporting the analysis.
   - Prepare a visually appealing presentation summarizing the key findings for presentation purposes.

Input: Pokemon.csv
Output: Plots like the following,





2. Multivariate Regression, PCA Analysis, and Regularization Techniques

The objective of this question is to create predictive models for fare prediction using multiple linear regression, Principal Component Analysis (PCA), and regularization techniques (Ridge and Lasso) on the 'uber.csv' dataset.

Steps:

1. **Exploratory Data Analysis (EDA):**
   - Load the 'uber.csv' dataset and perform EDA to understand its structure, features, distributions, and relationships.
   - Handle NULL values and eliminate any unwanted columns or data inconsistencies.

2. **Multiple Linear Regression:**
   - Prepare the dataset for multiple linear regression, identifying predictor variables and the target variable (fare).
   - Split the dataset into training and testing sets.
   - Implement a multiple linear regression model on the dataset without using PCA.
   - Evaluate the model's performance using appropriate metrics and visualize the results.

3. **PCA Analysis:**
   - Apply PCA on the cleaned dataset to reduce dimensionality.
   - Determine the number of principal components required to cover approximately 85-90% of the total variability in the dataset.
   - Select the identified principal components and transform the dataset.

4. **Multivariate Regression with PCA:**
   - Develop a multivariate regression model using the selected principal components obtained from PCA.
   - Split the dataset into training and testing sets.
   - Evaluate the PCA-based regression model's performance using appropriate metrics and visualize the results.

5. **Regularization Techniques - Ridge and Lasso:**
   - Apply Ridge and Lasso regularization techniques on the multiple linear regression model developed earlier.
   - Evaluate the performance of Ridge and Lasso regression models using metrics and compare them with the regular multiple linear regression.

6. **Comparison and Analysis:**

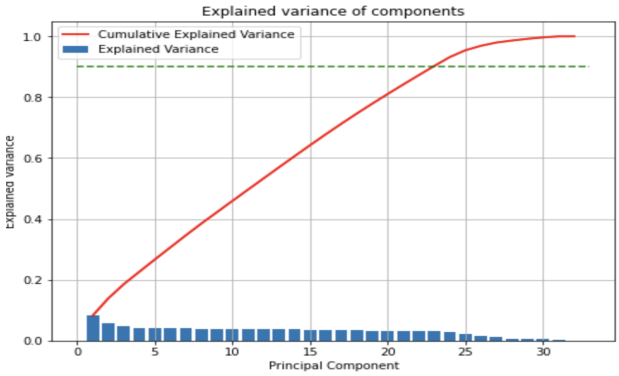
- Compare the performances of the multiple linear regression model without PCA, PCA-based regression, Ridge regression, and Lasso regression.
- Use appropriate plots and accuracy measures to illustrate the differences in predictive performance among these models.
- Analyze and discuss the impact of PCA and regularization techniques on the predictive accuracy of the models.

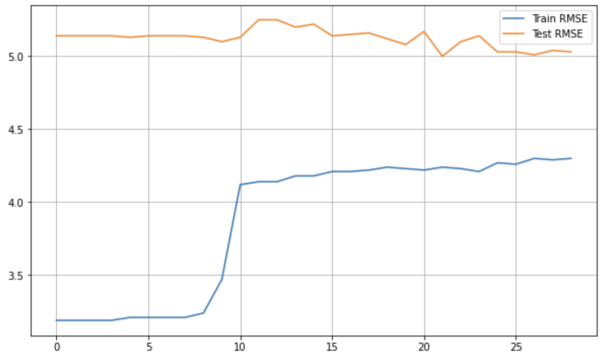
7. C**onclusion and Recommendations:**
- Conclude the findings and suggest the best approach for fare prediction based on the analyses performed.
- Provide recommendations for further improvements or adjustments in modeling techniques.

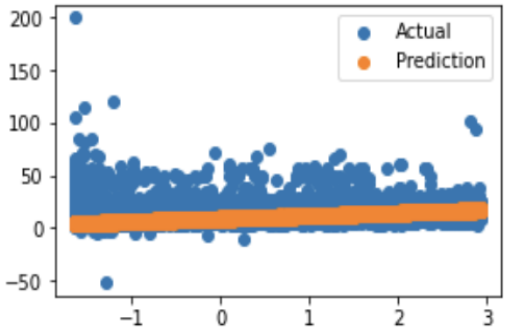
8. **Documentation and Presentation:**
- Present a detailed report documenting the steps taken, model implementations, results, and analysis.
- Include visualizations, performance metrics, and comparisons between different models.
- Prepare a clear and concise presentation summarizing the key findings and recommendations for a broader audience.larization), comparing their performances, and drawing conclusions based on the analyses.

Input: uber.csv

Outputs: Graphs on the same lines as the following,







3. KL Divergence

Code a function to calculate the KL divergence between two given discrete probability distributions.

```
def kl_divergence(p, q):

    """
    Calculate the Kullback-Leibler (KL) Divergence between two discrete probability distributions p and q.
    Args:
    p (list): A list of probabilities representing the 1st probability distribution.
    q (list): A list of probabilities representing the 2nd probability distribution.
    Returns:
    float: KL Divergence between p and q.
    """
```

X (Random Variable)	1	2	3	4
P(X) (p.d.f_1)	0.2	0.5	0.15	0.15
Q(X) (p.d.f_2)	0.3	0.3	0.3	0.10

The user enters the values of the probability distributions one after the other.
Calculate and output the KL divergence score using the function defined above.

4. Feature Selection 1

The given dataset is used to predict the housing prices of a locality (The column MEDV in the dataset - Median value of homes in $1000's) using 13 attributes such as the average no. of rooms of homes, per capita crime rate, accessibility to highways, property tax rate, etc.

Dataset: housing_data.csv

The objective is to build regression models to predict housing prices by finding the optimal subset of features using:
i) Greedy forward feature selection
ii) Greedy backward feature selection

Perform data preprocessing like standardization or min-max scaling and do an 80:20/90:10 train-test split.
Input Format: The provided dataset

Output Format: The feature names, feature count, training and testing errors of the best models from both the feature selection techniques.

5. Feature Selection 2

Use the dataset provided in Q4 and build regression models to predict housing prices by finding the optimal subset of features using the Pearson Correlation Coefficient:

Select a set of 1,2,3,...13 features by verifying which of these features shows a maximum linear relationship with the target attribute by using the Pearson correlation coefficient. Using these features, build a regression model and find the training and testing errors for each set of features.

Input Format: Q4 dataset
Output Format:
The feature names, feature count, training and testing error of the best model from the Pearson Correlation Coefficient Method

Comparative Analysis:
Perform a comparative analysis of the best models obtained from all the feature selection techniques (in Questions 6 and 7) and the regression model built using all 13 features.

Output Format:
A table displaying the training and testing errors of the best models from the three feature selection techniques and the model with all 13 features.
-> Note down which of these models gives the best performance.

6. Prior and Posterior Distributions

A survey was conducted by a media channel outside a theater to observe whether the audience liked the newly realized Tiger 3 movie or not. Let ‘p’ denote the probability of a person liking the movie. Before the first screening, movie experts assumed that ‘p’ follows a beta distribution with parameters $\alpha,\beta = (2,2)$. After the first screening, 50 of the 60 audience who were surveyed said they liked the movie. Plot the prior and posterior probability distribution of ‘p.’

The next day, the audience's opinion changed; out of the 50 people surveyed, 34 said they didn’t like the movie. Plot the posterior distribution of ‘p’ after this survey.

Input Format:
$\alpha,\beta = (2,2)$
First Screening: Like - 50, Total - 60 audience
Second Screening: Dislike - 34, Total - 50 audience

Output Format:
-> Plot for the prior distribution of ‘p’.
-> Plot for the posterior distribution of ‘p’ after the first screening.
-> Plot for the posterior distribution of ‘p’ after the second screening.

Explain in detail how you found the posterior distribution of ‘p’ in both cases.

7. Kernels

Definition
K(x, y) = <f(x), f(y)>. Here K is the kernel function, x, y are n dimensional inputs. f is a map from n-dimension to m-dimension space. <x,y> denotes the dot product. usually m is much larger than n.

Question
Given **feature mapping**:
Suppose x = (x1, x2, x3) , then
f(x) = (x1x1, x1x2, x1x3, x2x1, x2x2, x2x3, x3x1, x3x2, x3x3)

Given, **Kernel**: K(x, y) = $(<x, y>)^2$

Write two functions:

1. ***func_without_kernel(x, y) :*** Computes and returns the value of <f(x), f(y)>
2. ***func_with_kernel(x, y) :*** Computes the value of the kernel as defined above.

Example:
x = (1, 2, 3) , y = (4, 5, 6)
then f(x) = (1, 2, 3, 2, 4, 6, 3, 6, 9), f(y) = (16, 20, 24, 20, 25, 30, 24, 30, 36)
<f(x), f(y)> = 16 + 40 + 72 + 40 + 100 + 180 + 72 + 180 + 324 = 1024
K(x, y) = (<x, y>)^2 = (4 + 10 + 18)^2 = 1024

Input Format:

```
3                    #Dimension of the vectors
1 2 3                #Value of the 1st vector
4 5 6                #Value of the 2nd vector
```

Output Format:

```
1024                 #Value returned by the first function
1024                 #Value returned by the second function
```

Explainatio Required:
Determine if the values returned by both the functions are the same or not. If not, then why ?
If they are the same, then what is the advantage of using Kernels ?

8. Entropy

Problem Statement

Given a dataset with n features and a target variable. Your task is to implement an entropy-based feature sele
ction algorithm. The goal is to identify the k most informative features that maximize the mutual information with the target variable.
Write a Python function **feature_selection(X, y, k)** that takes the input features X (a 2D 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 question 10: (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.

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

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.

Example:

Input:

```
4 4 2
1 0 1 0
0 1 1 1
1 1 0 0
0 0 1 1
0 1 1 0
```

Output:

```
2 3
```