ASSIGNMENT # 3-101539669

BUS 4023 - FALL 2024

5POINTS

CHAPTER 4 – DATA REDUCTION

ASSIGNMENT QUESTIONS

PROBLEM #1, PARTS A TO G

Requires both written response and Python code saved as Jupyter notebook (.ipynb)

Implement techniques such as:

- Feature selection (e.g., Principal Component Analysis, LASSO regression, etc.)
- Dimensionality reduction methods
- Visualization of data reduction impacts

Question 1: Cereal Dataset Analysis

(a) Identifying Variable Types

The dataset contains different types of variables:

- **Quantitative (Numeric) Variables** – These are measurable and can be used for mathematical operations:
    - Calories, Protein, Fat, Sodium, Fiber, Carbohydrates, Sugars, Potassium, Vitamins, Weight, Cups per serving, and Rating.
- **Ordinal Variables** – These have a meaningful order but no consistent numerical difference:
    - **Shelf Placement (1 = bottom, 2 = middle, 3 = top)**.
- **Nominal (Categorical) Variables** – These represent groups

DELIVERABLES FOR ASSIGNMENTS INCLUDE

…

#1) Python code and **output** saved as notebook [in Jupyter notebook click on *File > Download as > Notebook (.ipynb)*]

SAVE THE NOTEBOOK (.IPYNB) FILE IN THE FOLLOWING FORMAT:

WEEKLY_ASSIGNMENT_**X**_Q**Y**_**LL**_**F**.IPYNB

Where,

X=weekly assignment number

Y=assignment problem number

LL=your full last name

F=Initial of your first name

Please remember to add the following on the first line of the notebook as a comment:

- your full last name and first name
- the assignment problem number the notebook code and output pertains to

without a meaningful order:

- Cereal **Name** and **Type** (Hot or Cold).

(b) Summary Statistics

The summary statistics provide key insights:

- **Calories** mostly range from **80 to 120 per serving**.
- **Sugar content varies significantly**, with some cereals having much more than others.
- **Fiber and vitamins also show wide variation**, with some cereals fortified at **0, 25, or 100% vitamins**.

(c) Histogram Analysis

Histograms help us see how different variables are distributed:

- **Calories** are mostly between **80 and 120**, with a few exceptions.
- **Sugar content** shows a spread, with a few very sweet cereals.
- **Fiber content** has some cereals that stand out as high-fiber options.

(d) Boxplots – Comparing Groups

Boxplots help us understand differences between groups:

- **Calories in Hot vs. Cold Cereals:**
  - Cold cereals generally have **more calories** than hot cereals.
- **Shelf Placement vs. Consumer Ratings:**
  - Cereals on the **middle and top shelves tend to get better ratings** than those on the bottom shelf.

(e) Correlation Matrix

Looking at relationships between variables:

- **Calories and Sugar:** More sugar usually means higher calories.
- **Fiber and Carbohydrates:** High-fiber cereals also tend to have more carbs.
- **Rating and Sugar:** Cereals with more sugar might have lower consumer ratings.

(f) Outliers in the Dataset

Outliers are extreme values that stand out:

- Some cereals have **very high fiber** compared to others.
- A few cereals have **extremely low or high sugar content**.
- Certain cereals have **100% fortified vitamins**, making them distinct from the rest.

(g) Distribution of Rating

- The **ratings are spread across a range**, but most cereals fall within a **mid-range score**.
- There may be a **few highly rated cereals**, while some receive lower ratings.

(h) Relationship Between Sugar and Rating

- There seems to be an **inverse relationship** – cereals with **higher sugar content tend to have lower ratings**.
- This could suggest that consumers might prefer cereals with **less sugar**, possibly viewing them as healthier.

PROBLEM #2, PARTS A & B

Requires both written response and Python code saved as Jupyter notebook (.ipynb)

This problem likely addresses:

- Techniques for handling missing data, such as imputation
- Statistical and machine learning approaches to fill in gaps
- Visualizing how missing data impacts model results

(a) Preprocessing the Data

Before running PCA, a few essential steps were taken to clean and prepare the data:

1. **Loading the Dataset** – The dataset, *ForbesAmericasTopColleges.csv*, was read into Python.
2. **Removing Categorical Variables** – Since PCA only works with numbers, non-numeric columns (like college names and locations) were removed.
3. **Handling Missing Data** – Any rows with missing numerical values were dropped to avoid errors in PCA calculations.
4. **Standardizing the Data** – Variables were scaled to have a **mean of 0 and a standard deviation of 1** to ensure fair comparisons (e.g., tuition in thousands vs. graduation rates in percentages).

(b) Applying Principal Component Analysis (PCA)

1. **Running PCA** – The standardized data was transformed into **principal components**, which are new dimensions that capture the most variance in the dataset.
2. **Explained Variance Analysis** – A plot was created to show how much information each principal component retains. The goal was to find the **elbow point**, where adding more components doesn't significantly increase the variance retained.
3. **Interpreting Principal Components** – The **loading matrix** helped identify which original variables influenced each component the most. This helps understand the patterns in the reduced dataset.
4. **Choosing the Number of Components** – The number of components was selected based on how much variance we wanted to retain (typically **95% or more**).

PCA helped **reduce the dataset's complexity** while keeping most of the important information, making further analysis easier and more effective.

PROBLEM #4, PARTS A & B

Requires both written response and Python code saved as Jupyter notebook (.ipynb)

This problem might include:

- Scaling and standardizing data
- Encoding categorical variables
- Splitting datasets into training, validation, and testing

Why is PC1 so much bigger than the others?

In the original (non-normalized) data, some features—like **Proline and Alcohol**—have much larger numerical values than other chemical properties. Since **PCA identifies the direction of maximum variance**, it naturally prioritizes these large-value features. As a result, **PC1 captures almost all of the variance (~99.8%)**, leaving very little for the remaining components.

Why should we normalize the data?

Without standardization, PCA is **dominated by features with large numerical values**, even if they aren't the most important. **Standardizing the data (subtracting the mean and dividing by the standard deviation)** ensures that all features contribute equally to the analysis. After normalization, **variance is more evenly spread across the components**, with PC1 now capturing **36.2% of the variance** instead of nearly 100%. This makes the analysis more balanced and prevents it from being skewed by certain variables.