STATS 3DA3

Homework Assignment 6

Instructor: Pratheepa Jeganathan

2025-03-21

Submission Deadline

- All submissions must be made before 10:00 PM on Wednesday, April 16, 2025.

Late Submissions

- Late Submission Penalty: A 15% deduction per day will be applied to assignments submitted after the deadline, including SAS accomdations.
- Late Submission Limit: Assignments submitted **more than 72 hours late** will receive a **grade of zero**, including SAS accomdations.

Submission Guidelines

- Format: Submit your work as a single PDF file via Avenue to Learn. You may submit individually or or as a group of up to three members.

Individual Submission

- Complete Questions 1–15.
- A GitHub repository is optional.

Group Submission

- Complete Questions 1–17.
- Group Size: Up to three members.
- Team Members' Contributions: For group submissions, you must complete Question 16, detailing each member's contributions. This should correspond with the commit history in the GitHub repository.
  - Note: While Question 16 is not graded, failure to include this information will result in the assignment not being graded.
  - Example:
    * Member A: Questions 1, 2, 4
    * Member B: Questions 3, 5, 6
- GitHub Repository: You must include a link to a public GitHub repository showing the assignment's version history.
  - Note: While Question 17 is not graded, failure to provide this information will result in the assignment not being graded.

Assignment Standards

Please ensure your assignment adheres to the following standards for submission:

- Title Page Requirements: Each submission must include a title page featuring your group members' names and student IDs. Assignments without a title page will not be considered for grading.
- Formatting Preferences: The use of Quarto Jupyter Notebook for document preparation is highly recommended.
- Font and Spacing: Submissions must utilize an eleven-point font (Times New Roman or a similar font) with 1.5 line spacing. Ensure margins of at least 1 inch on all sides.
- Individual Work: While discussing homework problems with peers and other groups is permitted, the final written submission must be your group work.

- Submission Content: Do not include the assignment questions within your PDF. Instead, clearly mark each response with the corresponding question number. Screenshots are not an acceptable form of submission under any circumstances.
- Academic Writing: Ensure that your writing and any references used are appropriate for an undergraduate level of study.
- Originality Checks: Be aware that the instructor may use various tools, including online resources or in-person meetings, to verify the originality of submitted work.

Assignment Policy on the Use of Generative AI

- The use of Generative AI is not permitted in assignments, except for using GitHub Copilot as a coding assistant.
  - If GitHub Copilot is used, you must clearly indicate this in the code comments.
- In alignment with McMaster academic integrity policy, it "shall be an offence knowingly to submit academic work for assessment that was purchased or acquired from another source". This includes work created by generative AI tools. Also state in the policy is the following, "Contract Cheating is the act of"outsourcing of student work to third parties" with or without payment." Using Generative AI tools is a form of contract cheating. Charges of academic dishonesty will be brought forward to the Office of Academic Integrity.

Heart Disease Classification Challenge

Overview

In this assignment, you will analyze the UCI Heart Disease Dataset, which contains medical records used to predict the presence of heart disease in patients. The dataset includes a mix of categorical and numerical variables, some missing values, and class imbalance.

For the context of data science methods for heart disease prediction, refer to - Detrano, R., et., al. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. The American journal of cardiology, 64(5), 304-310. DOI:10.1016/0002-9149(89)90524-9.

Dataset Information

The dataset is available at the UCI Machine Learning Repository:
https://archive.ics.uci.edu/ml/datasets/Heart+Disease

- Key Features:
  - The dataset includes 303 observations with 13 features.
  - Features include age, sex, chest pain type, resting blood pressure, cholesterol levels, fasting blood sugar, electrocardiographic results, and others.
  - The response variable is `num`, which will be transformed to binary in the analysis.

Objectives

Analyze the dataset using **two classification algorithms**. Your analysis should include exploratory data analysis, handling of missing values, feature selection, feature engineering, modeling, interpretation, and effective communication. The goal is to draw meaningful and well-supported conclusions from your analysis.

- Classifier requirement: **At least one** of the classifiers must be interpretable to allow for in-depth analysis and inference.

Assignment Questions

Address the following questions in your submission, providing detailed insights and conclusions based on your analysis.

1. Define and describe a classification problem using the dataset.
2. Apply any chosen data transformations, or explain why no transformations were necessary.
3. Provide a detailed description of the dataset, including variables, summaries, number of observations, data types, and distributions (include at least three statements).
4. Transform the response `num` into a binary outcome: `1` for heart disease and `0` for no heart disease. So combine 1, 2, 3, and 4 into `1` and 0 for `0`. For Questions 4-16, use the transformed binary outcome.
5. Analyze relationships between variables and discuss their implications for feature selection or extraction (include at least two statements).
6. Drop the rows with the missing values. How many osbervations after dropping the missing values. Skip the outlier analysis.
7. Sub-group analysis: Explore potential sub-groups within the data using appropriate data science methods. Identify and visualize these sub-groups without using the labels and categorical variables. Categroical variables already define sub groups so we don't need to include them for this analysis.
8. Split 30% of the data for testing using a random seed of 1. Use the remaining 70% for training and model selection.
9. Identify the two classifiers you have chosen. Justify your selections based on the classifier requirement for this assignment.
10. Specify two metrics to compare classifier performance. Provide technical details on how each metric is computed.
11. Train two selected classifiers in (9) and identify optimal tuning parameters (if applicable) using the training set.
12. Apply a feature selection or extraction method to one of the classifiers in (9). Train this third classifier on the training set and identify optimal tuning parameters (if applicable) using the training set.
13. Use the selected metrics to evaluate three classifiers in (11) and (12) on the test set.

- Discuss your findings (at least two statements).
- Discuss the impact of feature selection or extraction on the performance of the classifier (at least one statement).

14. For the best interpretable model identified in (13), analyze and interpret the most important predictor variables in the context of the classification challenge (at least two statements).
15. [**Bonus**] Sub-group improvement strategy: If sub-groups were identified, propose and implement a method to further improve the performance of **one** classifier. Compare the fourth classifier performance with the results from (13).
16. **Team Contributions:** Document each team member's specific contributions to the questions above. For group submissions, this should match the GitHub commit history. Individual submissions do not need to address this question.
17. **Link** to the public GitHub repository. This is optional for the individual submissions.

Notes

- Students can also choose one classifer not covered in the lectures.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, f1_score, classification_report
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.feature_selection import SelectKBest, chi2
```

```
dataset_url = "https://raw.githubusercontent.com/PratheepaJ/datasets/refs/heads/master/ass6-da
df = pd.read_csv(dataset_url)
```

```
print("Initial dataset head:")
print(df.head())
```

```
Initial dataset head:
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   63    1   1       145   233    1        2      150      0      2.3      3
1   67    1   4       160   286    0        2      108      1      1.5      2
2   67    1   4       120   229    0        2      129      1      2.6      2
3   37    1   3       130   250    0        0      187      0      3.5      3
4   41    0   2       130   204    0        2      172      0      1.4      1

    ca  thal  num
0  0.0   6.0    0
1  3.0   3.0    2
2  2.0   7.0    1
3  0.0   3.0    0
4  0.0   3.0    0
```

1 The goal is to predict whether a patient has heart disease (binary outcome) using various clinical and demographic features

```
# 3
print("\n--- Dataset Information ---")
print(df.info())
print("\n--- Statistical Summary ---")
print(df.describe())
```

```
--- Dataset Information ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
```

```
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        299 non-null    float64
 12  thal      301 non-null    float64
 13  num       303 non-null    int64
dtypes: float64(3), int64(11)
memory usage: 33.3 KB
None

--- Statistical Summary ---
              age         sex          cp    trestbps        chol         fbs  \
count  303.000000  303.000000  303.000000  303.000000  303.000000  303.000000
mean    54.438944    0.679868    3.158416  131.689769  246.693069    0.148515
std      9.038662    0.467299    0.960126   17.599748   51.776918    0.356198
min     29.000000    0.000000    1.000000   94.000000  126.000000    0.000000
25%     48.000000    0.000000    3.000000  120.000000  211.000000    0.000000
50%     56.000000    1.000000    3.000000  130.000000  241.000000    0.000000
75%     61.000000    1.000000    4.000000  140.000000  275.000000    0.000000
max     77.000000    1.000000    4.000000  200.000000  564.000000    1.000000

          restecg     thalach       exang     oldpeak       slope          ca  \
```

count	303.000000	303.000000	303.000000	303.000000	303.000000	299.000000
mean	0.990099	149.607261	0.326733	1.039604	1.600660	0.672241
std	0.994971	22.875003	0.469794	1.161075	0.616226	0.937438
min	0.000000	71.000000	0.000000	0.000000	1.000000	0.000000
25%	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000
50%	1.000000	153.000000	0.000000	0.800000	2.000000	0.000000
75%	2.000000	166.000000	1.000000	1.600000	2.000000	1.000000
max	2.000000	202.000000	1.000000	6.200000	3.000000	3.000000

	thal	num
count	301.000000	303.000000
mean	4.734219	0.937294
std	1.939706	1.228536
min	3.000000	0.000000
25%	3.000000	0.000000
50%	3.000000	0.000000
75%	7.000000	2.000000
max	7.000000	4.000000

Three descriptive statements:

The dataset consists of clinical and demographic variables (e.g., age, sex, chest pain type, blood pressure, cholesterol) that are typical in a heart disease dataset. The continuous variables (such as age, trestbps, chol, and thalach) show a broad range and their distributions may be normal or slightly skewed. The dataset initially contains a number of observations (df.shape[0]) and includes some missing values.

```
# 4
df['num'] = df['num'].apply(lambda x: 0 if x == 0 else 1)
print("\nValue counts for transformed binary target 'num':")
print(df['num'].value_counts())
```

```
Value counts for transformed binary target 'num':
num
0    164
1    139
Name: count, dtype: int64
```

```python
# 5(need further explaining)
corr_matrix = df.corr()
print("\nCorrelation matrix:")
print(corr_matrix)
```

```
Correlation matrix:
               age       sex        cp  trestbps      chol       fbs  \
age       1.000000 -0.097542  0.104139  0.284946  0.208950  0.118530
sex      -0.097542  1.000000  0.010084 -0.064456 -0.199915  0.047862
cp        0.104139  0.010084  1.000000 -0.036077  0.072319 -0.039975
trestbps  0.284946 -0.064456 -0.036077  1.000000  0.130120  0.175340
chol      0.208950 -0.199915  0.072319  0.130120  1.000000  0.009841
fbs       0.118530  0.047862 -0.039975  0.175340  0.009841  1.000000
restecg   0.148868  0.021647  0.067505  0.146560  0.171043  0.069564
thalach  -0.393806 -0.048663 -0.334422 -0.045351 -0.003432 -0.007854
exang     0.091661  0.146201  0.384060  0.064762  0.061310  0.025665
oldpeak   0.203805  0.102173  0.202277  0.189171  0.046564  0.005747
slope     0.161770  0.037533  0.152050  0.117382 -0.004062  0.059894
ca        0.362605  0.093185  0.233214  0.098773  0.119000  0.145478
thal      0.127389  0.380936  0.265246  0.133554  0.014214  0.071358
num       0.223120  0.276816  0.414446  0.150825  0.085164  0.025264

           restecg   thalach     exang   oldpeak     slope        ca  \
age       0.148868 -0.393806  0.091661  0.203805  0.161770  0.362605
```

```
sex       0.021647 -0.048663  0.146201  0.102173  0.037533  0.093185
cp        0.067505 -0.334422  0.384060  0.202277  0.152050  0.233214
trestbps  0.146560 -0.045351  0.064762  0.189171  0.117382  0.098773
chol      0.171043 -0.003432  0.061310  0.046564 -0.004062  0.119000
fbs       0.069564 -0.007854  0.025665  0.005747  0.059894  0.145478
restecg   1.000000 -0.083389  0.084867  0.114133  0.133946  0.128343
thalach  -0.083389  1.000000 -0.378103 -0.343085 -0.385601 -0.264246
exang     0.084867 -0.378103  1.000000  0.288223  0.257748  0.145570
oldpeak   0.114133 -0.343085  0.288223  1.000000  0.577537  0.295832
slope     0.133946 -0.385601  0.257748  0.577537  1.000000  0.110119
ca        0.128343 -0.264246  0.145570  0.295832  0.110119  1.000000
thal      0.024531 -0.279631  0.329680  0.341004  0.287232  0.256382
num       0.169202 -0.417167  0.431894  0.424510  0.339213  0.460442

              thal       num
age       0.127389  0.223120
sex       0.380936  0.276816
cp        0.265246  0.414446
trestbps  0.133554  0.150825
chol      0.014214  0.085164
fbs       0.071358  0.025264
restecg   0.024531  0.169202
thalach  -0.279631 -0.417167
exang     0.329680  0.431894
oldpeak   0.341004  0.424510
slope     0.287232  0.339213
ca        0.256382  0.460442
thal      1.000000  0.525689
num       0.525689  1.000000
```

```
# 6
df_clean = df.dropna()
```

```
print("\nNumber of observations after dropping missing values:", df_clean.shape[0])
```

```
Number of observations after dropping missing values: 297
```

```
numeric_cols = df_clean.select_dtypes(include=[np.number]).columns.tolist()
if 'num' in numeric_cols:
    numeric_cols.remove('num')
print("\nContinuous variables used for subgroup analysis:", numeric_cols)
```

```
Continuous variables used for subgroup analysis: ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs
```

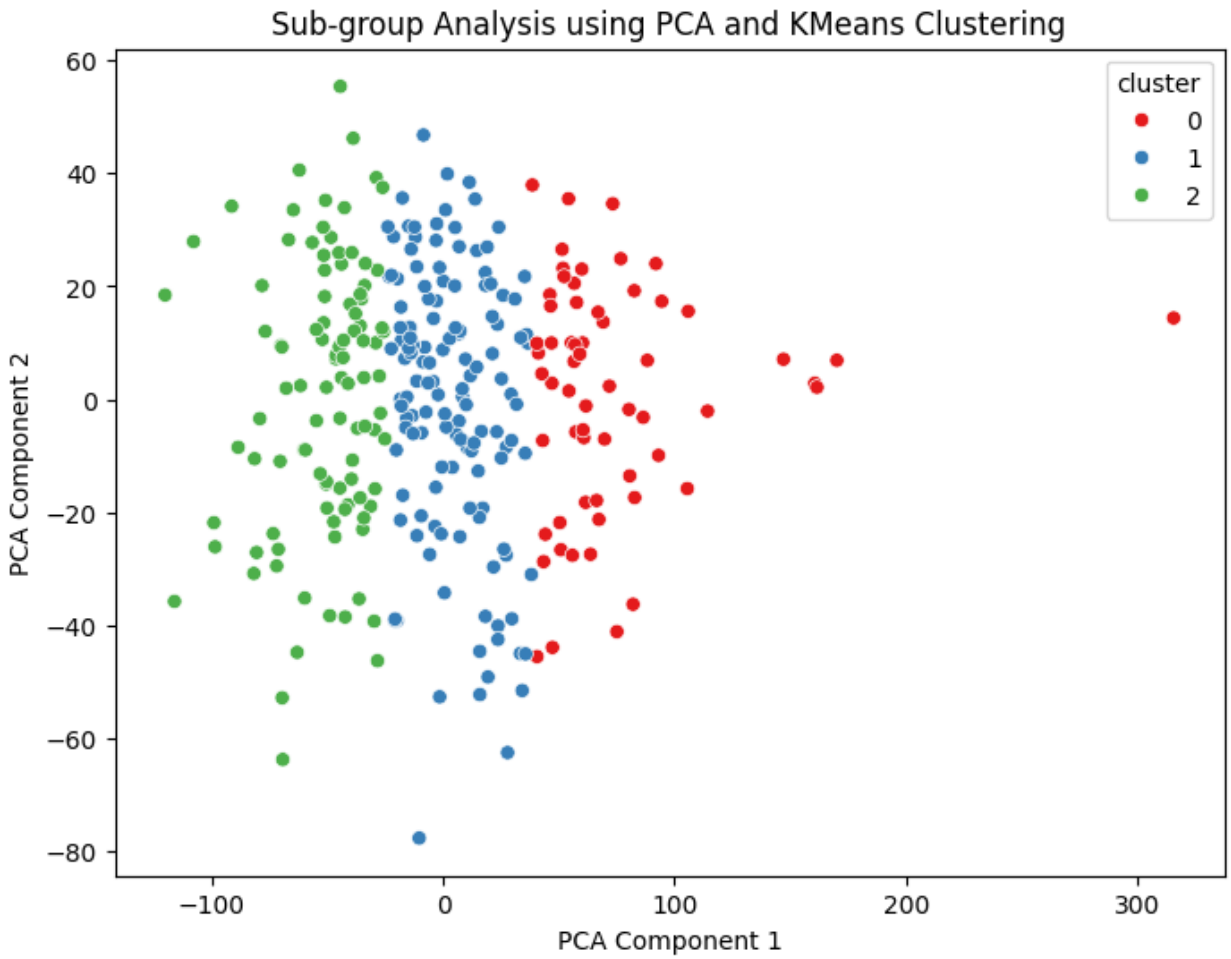
```
pca = PCA(n_components=2, random_state=1)
pca_results = pca.fit_transform(df_clean[numeric_cols])
df_clean['pca1'] = pca_results[:, 0]
df_clean['pca2'] = pca_results[:, 1]
```

```
/var/folders/bz/_xbkcp397bb676k7lcghv7b40000gn/T/ipykernel_15134/2647241709.py:3: SettingWithC
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/i
  df_clean['pca1'] = pca_results[:, 0]
/var/folders/bz/_xbkcp397bb676k7lcghv7b40000gn/T/ipykernel_15134/2647241709.py:4: SettingWithC
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/i
  df_clean['pca2'] = pca_results[:, 1]
```

```
kmeans = KMeans(n_clusters=3, random_state=1)
df_clean['cluster'] = kmeans.fit_predict(df_clean[numeric_cols])
```

```
/var/folders/bz/_xbkcp397bb676k7lcghv7b40000gn/T/ipykernel_15134/4096796358.py:2: SettingWithC
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/
  df_clean['cluster'] = kmeans.fit_predict(df_clean[numeric_cols])
```

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='pca1', y='pca2', hue='cluster', data=df_clean, palette='Set1')
plt.title('Sub-group Analysis using PCA and KMeans Clustering')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.show()
```




```
# 8
# Prepare the features and target variables while excluding the additional subgroup columns.
features = df_clean.drop(columns=['num', 'cluster', 'pca1', 'pca2'])
target = df_clean['num']

# Split the data using 30% for testing and 70% for training (random seed = 1).
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.3, random_sta
print("\nTraining set size:", X_train.shape[0])
print("Test set size:", X_test.shape[0])
```

```
Training set size: 207
Test set size: 90
```

9

Clusters

Grading scheme

1.	Answer [1]
2.	Codes (variable type transformation, etc.) [1] OR rationale for no transformation [1]
3.	Codes [3] and three statements [2]
4.	Codes for transforming the response variable [1]
5.	Codes for association [2] and interpretation of figures or tables [2]
6.	Codes [1] answer [1]
7.	Codes to identify sub groups [3] and Plot the sub groups [1]
8.	Codes [1]
9.	classifiers and justification [2]
10.	Describe the two metrics [2]
11.	Codes for training two classifiers [2] Codes for tuning parameters (if any) [1]
12.	Codes for feature selection or feature extraction [1] Codes for training the third classifier with the selected or extracted features [1] Codes for tuning parameters (if any) [1]
13.	Codes for evaluating three classifiers on the test set using two metrics in (10) [3] Two statements for the findings [2] One statement for the impact of feature selection or extraction [1]
14.	Codes finding the important variables [1] Two statements for the analysis and interpretation of the most important predictor variables [2]

15.	Codes for the sub-group improvement strategy (training and tuning parameters, if any) [Bonus 2] Comparison of the performance with the results from (13) [Bonus 1] **Bonus 3 points will be added to the final grade**
16.	Document each team member's specific contributions
17.	Link to the public GitHub repository

The maximum point for this assignment is 39. We will convert this to 100%. The bonus 3 points will be added to the final grade.

All group members will receive the same grade if they contribute to the same.