Final Project

Jenna Orvitz, Noah Ripstein, Viransh Shah

2024-04-18

```python
from ucimlrepo import fetch_ucirepo
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
import seaborn as sns
```

```python
from ucimlrepo import fetch_ucirepo

# fetch dataset
chronic_kidney_disease = fetch_ucirepo(id=336)

# data (as pandas dataframes)
X = chronic_kidney_disease.data.features
y = chronic_kidney_disease.data.targets

# metadata
print(chronic_kidney_disease.metadata)

# variable information
print(chronic_kidney_disease.variables)

df = pd.concat([X,y], axis=1)
df.head()
```

```
{'uci_id': 336, 'name': 'Chronic Kidney Disease', 'repository_url': 'https://archive.ics.uci.e
     name     role         type demographic              description  \
0     age  Feature      Integer         Age                     None
1      bp  Feature      Integer        None           blood pressure
2      sg  Feature  Categorical        None         specific gravity
3      al  Feature  Categorical        None                  albumin
4      su  Feature  Categorical        None                    sugar
5     rbc  Feature       Binary        None          red blood cells
6      pc  Feature       Binary        None                 pus cell
7     pcc  Feature       Binary        None          pus cell clumps
8      ba  Feature       Binary        None                 bacteria
9     bgr  Feature      Integer        None     blood glucose random
10     bu  Feature      Integer        None               blood urea
11     sc  Feature   Continuous        None         serum creatinine
12    sod  Feature      Integer        None                   sodium
13    pot  Feature   Continuous        None                potassium
14   hemo  Feature   Continuous        None               hemoglobin
15    pcv  Feature      Integer        None        packed cell volume
16   wbcc  Feature      Integer        None   white blood cell count
17   rbcc  Feature   Continuous        None     red blood cell count
18    htn  Feature       Binary        None             hypertension
19     dm  Feature       Binary        None        diabetes mellitus
20    cad  Feature       Binary        None  coronary artery disease
21  appet  Feature       Binary        None                 appetite
22     pe  Feature       Binary        None              pedal edema
23    ane  Feature       Binary        None                   anemia
24  class   Target       Binary        None           ckd or not ckd

       units missing_values
0       year            yes
1      mm/Hg            yes
2       None            yes
```

```
3           None            yes
4           None            yes
5           None            yes
6           None            yes
7           None            yes
8           None            yes
9         mgs/dl            yes
10        mgs/dl            yes
11        mgs/dl            yes
12         mEq/L            yes
13         mEq/L            yes
14           gms            yes
15          None            yes
16     cells/cmm            yes
17  millions/cmm            yes
18          None            yes
19          None            yes
20          None            yes
21          None            yes
22          None            yes
23          None            yes
24          None             no
```

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	wbcc	rbc
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	...	44.0	7800.0	5.2
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	...	38.0	6000.0	Na
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	...	31.0	7500.0	Na
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	...	32.0	6700.0	3.9
4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	...	35.0	7300.0	4.6

1. Classification Problem Identification: Define and describe a classification problem based on the dataset.

Using different variables does a person have Chronic Kidney Disease.

2. Variable Transformation: Implement any transformations chosen or justify the absence of such modifications.

```
df.describe()
```

	age	bp	sg	al	su	bgr	bu	sc
count	391.000000	388.000000	353.000000	354.000000	351.000000	356.000000	381.000000	383.000000
mean	51.483376	76.469072	1.017408	1.016949	0.450142	148.036517	57.425722	3.072454
std	17.169714	13.683637	0.005717	1.352679	1.099191	79.281714	50.503006	5.741126
min	2.000000	50.000000	1.005000	0.000000	0.000000	22.000000	1.500000	0.400000
25%	42.000000	70.000000	1.010000	0.000000	0.000000	99.000000	27.000000	0.900000
50%	55.000000	80.000000	1.020000	0.000000	0.000000	121.000000	42.000000	1.300000
75%	64.500000	80.000000	1.020000	2.000000	0.000000	163.000000	66.000000	2.800000
max	90.000000	180.000000	1.025000	5.000000	5.000000	490.000000	391.000000	76.000000

```
float64_columns = df.select_dtypes(
    include=['float64']
    ).columns
float64_columns
scaler = StandardScaler()
df[float64_columns] = scaler.fit_transform(df[float64_columns])
```

```
cat_columns = df.select_dtypes(
    include=['object']
    ).columns

for col in cat_columns:
    print(df[col].value_counts(normalize=True))
```

```
rbc
```

```
normal      0.810484
abnormal    0.189516
Name: proportion, dtype: float64
pc
normal      0.773134
abnormal    0.226866
Name: proportion, dtype: float64
pcc
notpresent    0.893939
present       0.106061
Name: proportion, dtype: float64
ba
notpresent    0.944444
present       0.055556
Name: proportion, dtype: float64
htn
no     0.630653
yes    0.369347
Name: proportion, dtype: float64
dm
no      0.653266
yes     0.344221
\tno    0.002513
Name: proportion, dtype: float64
cad
no     0.914573
yes    0.085427
Name: proportion, dtype: float64
appet
good    0.794486
poor    0.205514
Name: proportion, dtype: float64
```

```
pe
no     0.809524
yes    0.190476
Name: proportion, dtype: float64
ane
no     0.849624
yes    0.150376
Name: proportion, dtype: float64
class
ckd       0.620
notckd    0.375
ckd\t     0.005
Name: proportion, dtype: float64
```

```
for col in cat_columns:
    df[col] = df[col].astype('category').cat.codes
df.head(5)
```

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv
0	-0.203139	0.258373	0.454071	-0.012548	-0.410106	-1	1	0	0	-0.341498	...	0.569881
1	-2.594124	-1.936857	0.454071	2.208413	-0.410106	-1	1	0	0	NaN	...	-0.098536
2	0.613295	0.258373	-1.297699	0.727772	2.323069	1	1	0	0	3.473064	...	-0.878356
3	-0.203139	-0.473370	-2.173584	2.208413	-0.410106	1	0	1	0	-0.392022	...	-0.766953
4	-0.028189	0.258373	-1.297699	0.727772	-0.410106	1	1	0	0	-0.530963	...	-0.432744

3. Dataset Overview: Provide a detailed description of the dataset, covering variables, summaries, observation counts, data types, and distributions (at least three statements).

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
```

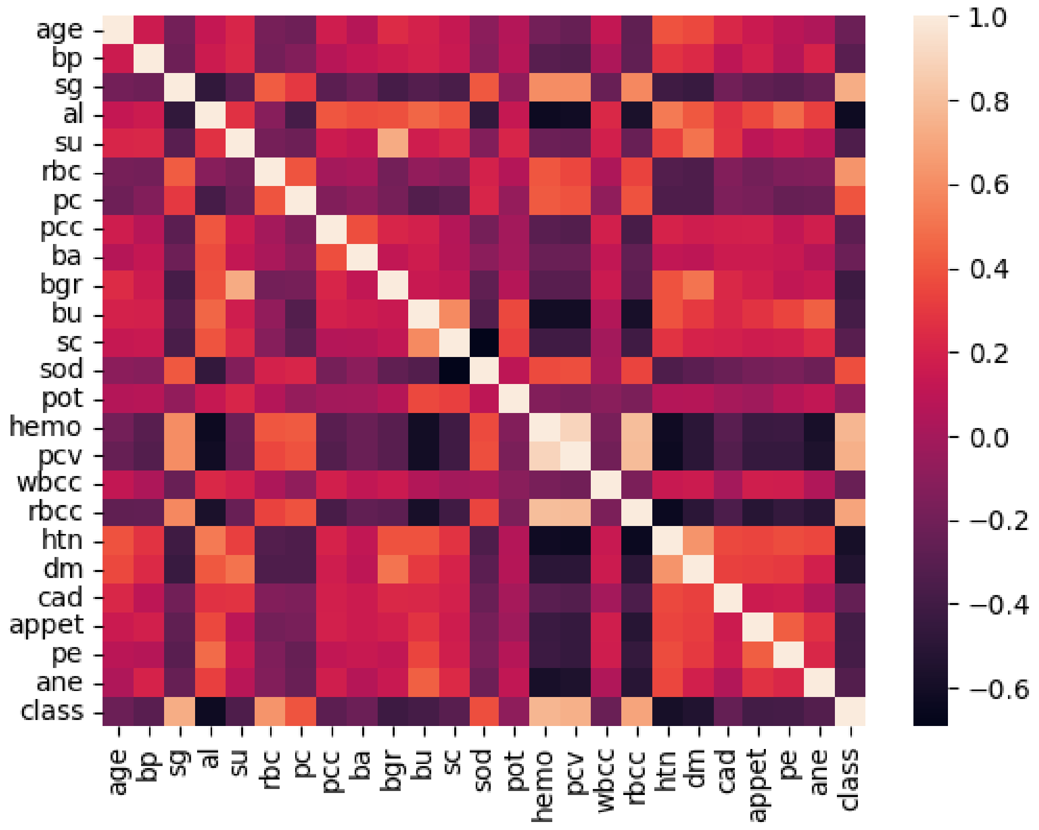
```
Data columns (total 25 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   age     391 non-null    float64
 1   bp      388 non-null    float64
 2   sg      353 non-null    float64
 3   al      354 non-null    float64
 4   su      351 non-null    float64
 5   rbc     400 non-null    int8
 6   pc      400 non-null    int8
 7   pcc     400 non-null    int8
 8   ba      400 non-null    int8
 9   bgr     356 non-null    float64
 10  bu      381 non-null    float64
 11  sc      383 non-null    float64
 12  sod     313 non-null    float64
 13  pot     312 non-null    float64
 14  hemo    348 non-null    float64
 15  pcv     329 non-null    float64
 16  wbcc    294 non-null    float64
 17  rbcc    269 non-null    float64
 18  htn     400 non-null    int8
 19  dm      400 non-null    int8
 20  cad     400 non-null    int8
 21  appet   400 non-null    int8
 22  pe      400 non-null    int8
 23  ane     400 non-null    int8
 24  class   400 non-null    int8
dtypes: float64(14), int8(11)
memory usage: 48.2 KB
```

```
df.describe()
```

	age	bp	sg	al	su	rbc	pc	pc
count	3.910000e+02	3.880000e+02	3.530000e+02	354.000000	351.000000	400.00000	400.000000	40
mean	9.994847e-17	-2.380684e-16	2.415443e-15	0.000000	0.000000	0.12250	0.485000	0.
std	1.001281e+00	1.001291e+00	1.001419e+00	1.001415	1.001428	0.93256	0.759089	0.
min	-2.885708e+00	-1.936857e+00	-2.173584e+00	-0.752868	-0.410106	-1.00000	-1.000000	-1
25%	-5.530393e-01	-4.733701e-01	-1.297699e+00	-0.752868	-0.410106	-1.00000	0.000000	0.
50%	2.050779e-01	2.583733e-01	4.540705e-01	-0.752868	-0.410106	1.00000	1.000000	0.
75%	7.590867e-01	2.583733e-01	4.540705e-01	0.727772	-0.410106	1.00000	1.000000	0.
max	2.246163e+00	7.575807e+00	1.329955e+00	2.948733	4.145186	1.00000	1.000000	1.

4. Association Between Variables: Analyze variable relationships and their implications for feature selection or extraction (at least three statements)

```
sns.heatmap(df.corr())
```



5. Missing Value Analysis and Handling: Implement your strategy for identifying and addressing missing values in the dataset, or provide reasons for not addressing them.
6. Outlier Analysis: Implement your approach for identifying and managing outliers, or provide reasons for not addressing them.
7. Sub-group Analysis: Explore potential sub-groups within the data, employing appropriate data science methods to find the sub-groups of patients and visualize the sub-groups. The sub-group analysis must not include the labels (for CKD patients and healthy controls).
8. Data Splitting: Segregate 30% of the data for testing, using a random seed of 1. Use the remaining 70% for training and model selection.
9. Classifier Choices: Identify the two classifiers you have chosen and justify your selections.
10. Performance Metrics: Outline the two metrics for comparing the performance of the classifiers.
11. Feature Selection/Extraction: Implement methods to enhance the performance of at least one classifier in (9). The answer for this question can be included in (12).
12. Classifier Comparison: Utilize the selected metrics to compare the classifiers based on the test set. Discuss your findings (at least two statements).
13. Interpretable Classifier Insight: After re-training the interpretable classifier with all available data, analyze and interpret the significance of predictor variables in the context of the data and the challenge (at least two statements).
14. Sub-group Improvement Strategy: If sub-groups were identified, propose and implement a method to improve one classifier performance further. Compare the performance of the new classifer with the results in (12).

Contributions

Jenna: Created/set up repository and jupyter notebook, started working on questions 1-4

Github Link

Github link (https://github.com/JennaOrvitz/Stats3DA3FinalProject/tree/main)

References

Rubini, Soundarapandian, L., and P. Eswaran. 2015. "Chronic Kidney Disease." UCI Machine Learning Repository.

Sanmarchi, Francesco, Claudio Fanconi, Davide Golinelli, Davide Gori, Tina Hernandez-Boussard, and Angelo Capodici. 2023. "Predict, Diagnose, and Treat Chronic Kidney Disease with Machine Learning: A Systematic Literature Review - Journal of Nephrology." *SpringerLink*. Springer International Publishing. https://link.springer.com/article/10.1007/s40620-023-01573-4.