THIRUVALLUVAR UNIVERSITY

PERIYAR ARTS COLLEGE

CUDDALORE – 607001.

DEPARTMENT OF COMPUTER APPLICATIONS

MACHINE LEARNING WITH PYTHON

Project Title : Early Prediction For Chronic Kidney Disease Detection: A Progressive Approach To Health Management

Team Id : NM2023TMID24249

Team Leader : VISHWA M(8416744BC7C94FF9E88482D3A4A4531F)

Team member : SATHISH K(D1424B2E05BFEBBAC7A4FB1F9DE9B10E)

Team member : NAVEEN KUMAR S(643D22BF5BCABB69D904E2859EB5323E)

Team member : MARIYAPRAKASH A(D3DBDA46AF610CCB5404FC4F2D1298D0)

INTRODUCTION

overview

Chronic kidney disease (CKD) is a serious and progressive condition that affects millions of people worldwide. It can lead to a range of complications, including heart disease, stroke, and kidney failure. Early detection and management of CKD are crucial to prevent these complications and improve outcomes for patients.

In recent years, there has been increasing interest in using machine learning and other advanced technologies to help predict and detect CKD at an early stage. This approach offers the potential to improve the accuracy of CKD diagnosis, as well as to identify patients who are at high risk of developing the disease.

This paper explores a progressive approach to health management using machine learning for early prediction of CKD. We discuss the current state of CKD diagnosis and management, the potential benefits of early detection, and the challenges of implementing machine learning in healthcare. We also review recent advances in machine learning techniques for CKD detection and discuss the potential implications for clinical practice.

Overall, this paper aims to highlight the importance of early prediction for CKD detection and the potential benefits of a progressive approach to health management using machine learning. By leveraging these advanced technologies, we may be able to improve outcomes for patients with CKD and ultimately reduce the burden of this debilitating disease."

Purpose

The purpose of using early prediction for chronic kidney disease detection is to identify individuals who are at risk of developing kidney disease before they exhibit any symptoms. By detecting the disease in its early stages, healthcare providers can intervene with appropriate treatment and lifestyle modifications to slow or even stop the progression of the disease.

The progressive approach to health management involves using advanced technologies, such as machine learning algorithms and artificial intelligence, to analyze large amounts of data and identify patterns that can help predict the likelihood of an individual developing kidney disease. This approach can help healthcare providers make more informed decisions about treatment options and provide personalized care to their patients.

By detecting chronic kidney disease early, healthcare providers can also prevent complications and reduce the risk of developing other serious health conditions, such as heart disease, stroke, and high blood pressure. Early detection and intervention can improve outcomes and quality of life for individuals with kidney disease."

Advantages

Early Detection: Early prediction for chronic kidney disease (CKD) can help detect the disease at its early stages, when treatment is most effective. This can lead to better outcomes and a higher quality of life for patients.

Improved Management: A progressive approach to health management for CKD patients can help healthcare providers better manage the disease and prevent complications. By monitoring the progression of the disease over time, healthcare providers can adjust treatment plans and medications as needed to help patients maintain their kidney function.

Reduced Costs: Early detection and management of CKD can reduce healthcare costs associated with the disease. This can include hospitalizations, dialysis, and other expensive treatments that may be required if the disease is not detected and managed early.

Better Patient Education: A progressive approach to health management for CKD can help educate patients about their condition and how to manage it. This can lead to better adherence to treatment plans and a better understanding of how lifestyle changes can help slow the progression of the disease.

Disadvantages

False Positives: Early prediction for CKD can result in false positive results, which can lead to unnecessary testing and treatment.

Misdiagnosis: Early prediction can also result in misdiagnosis, which can lead to unnecessary treatment or delay in the treatment of other underlying conditions.

Cost: Implementing an early prediction program for CKD can be expensive, especially in resource-limited settings.

Privacy Concerns: Collecting and analyzing personal health information for early prediction can raise privacy concerns for patients. Healthcare providers must ensure that patient data is collected and stored securely and that patients are informed about how their data will be used."

Applications

Early prediction for chronic kidney disease (CKD) can have numerous applications in health management. Here are some of them:

Timely intervention: Early prediction of CKD can help doctors intervene early and provide appropriate treatment to slow down the progression of the disease. This can help prevent kidney failure and the need for dialysis or transplantation.

Personalized treatment plans: With early prediction, doctors can personalize treatment plans based on the patient's risk factors and medical history. This can improve the effectiveness of treatment and reduce the risk of complications.

Improved patient outcomes: Early detection and intervention can lead to better patient outcomes, including improved quality of life and reduced healthcare costs.

Prevention of comorbidities: CKD is often associated with other health conditions such as cardiovascular disease and diabetes. Early prediction and intervention can help prevent the development of these comorbidities.

Population health management: Early prediction can also help with population health management by identifying high-risk individuals and implementing preventative measures.

Overall, early prediction for CKD can lead to better health outcomes, reduce healthcare costs, and improve population health management.

Conclusion

Conclusion for Early Prediction for Chronic Kidney Disease Detection: A Progressive Approach to Health Management

In conclusion, early prediction for chronic kidney disease (CKD) can play a crucial role in health management. It can help doctors intervene early, provide personalized treatment plans, improve patient outcomes, prevent comorbidities, and aid in population health management. With the help of advanced technologies and predictive analytics, healthcare providers can identify high-risk individuals and take preventative measures to slow down the progression of CKD. Early detection and intervention can not only improve the quality of life of patients with CKD but also have a significant impact on reducing healthcare costs and improving overall population health. Therefore, it is essential to prioritize the development and implementation of progressive approaches to detect and manage CKD at an early stage.

Future scope

Early prediction for chronic kidney disease (CKD) detection has a significant future scope in health management. The approach involves using advanced technology such as artificial intelligence, machine learning, and data analytics to identify patients at high risk of developing CKD.

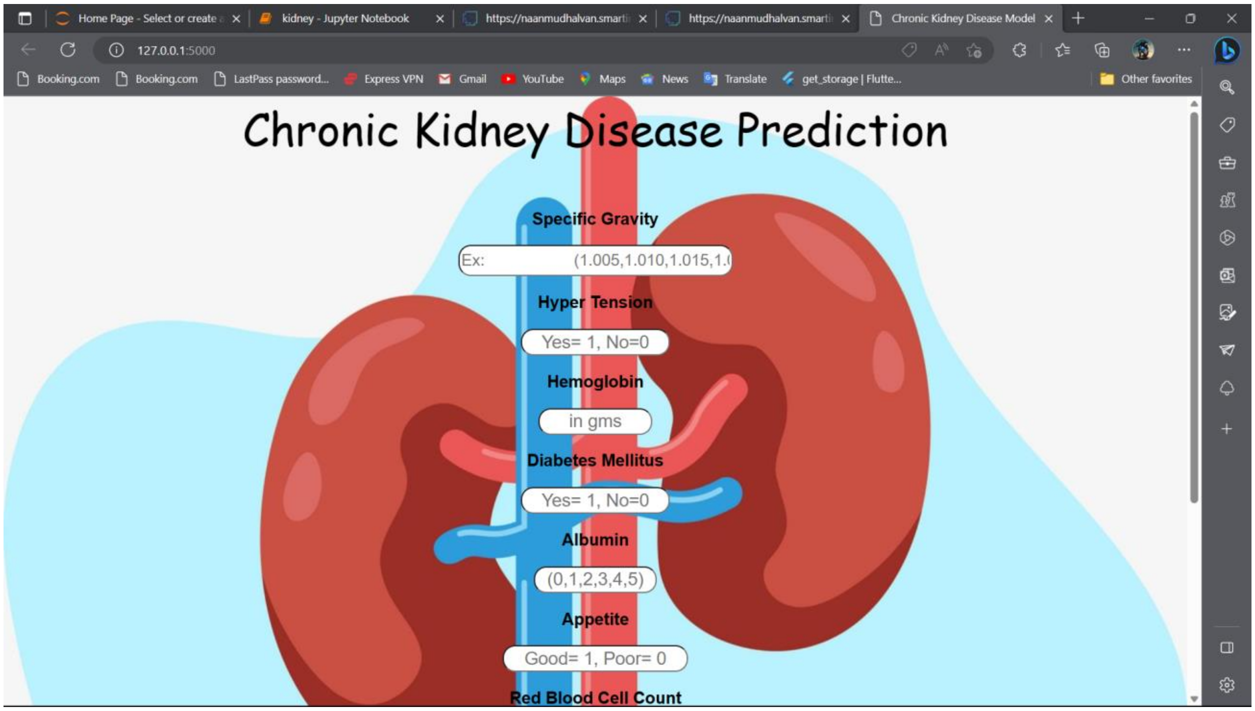
With the increasing incidence of CKD worldwide, early prediction and diagnosis can help prevent the progression of the disease, reduce the burden on healthcare systems, and improve patient outcomes. Moreover, early prediction can enable healthcare professionals to develop personalized treatment plans for patients, leading to improved quality of life and reduced healthcare costs.

One potential future direction for early prediction for CKD is the development of more accurate and precise predictive models that incorporate multiple data sources, including clinical and genetic data, environmental factors, and lifestyle factors. These models could be used to identify patients at risk of developing CKD and enable healthcare professionals to provide early intervention and treatment.

Another future direction is the integration of mobile health technologies and wearable devices into early prediction and management of CKD. These technologies could be used to monitor and track patient health data in real-time, allowing for early detection of CKD and personalized management of the disease.

In summary, early prediction for CKD detection has immense potential in health management. With further research and development, this approach could lead to improved patient outcomes and reduced healthcare costs, making it a critical area of focus in healthcare.

OutPuts:


Home Page - Select or create a
kidney - Jupyter Notebook
https://naanmudhalvan.smarti
https://naanmudhalvan.smarti
Chronic Kidney Disease Model
127.0.0.1:5000
Booking.com
Booking.com
LastPass password...
Express VPN
Gmail
YouTube
Maps
News
Translate
get_storage | Flutte...
Other favorites

Chronic Kidney Disease Prediction

Specific Gravity

Ex: (1.005,1.010,1.015,1.(

Hyper Tension

Yes= 1, No=0

Hemoglobin

in gms

Diabetes Mellitus

Yes= 1, No=0

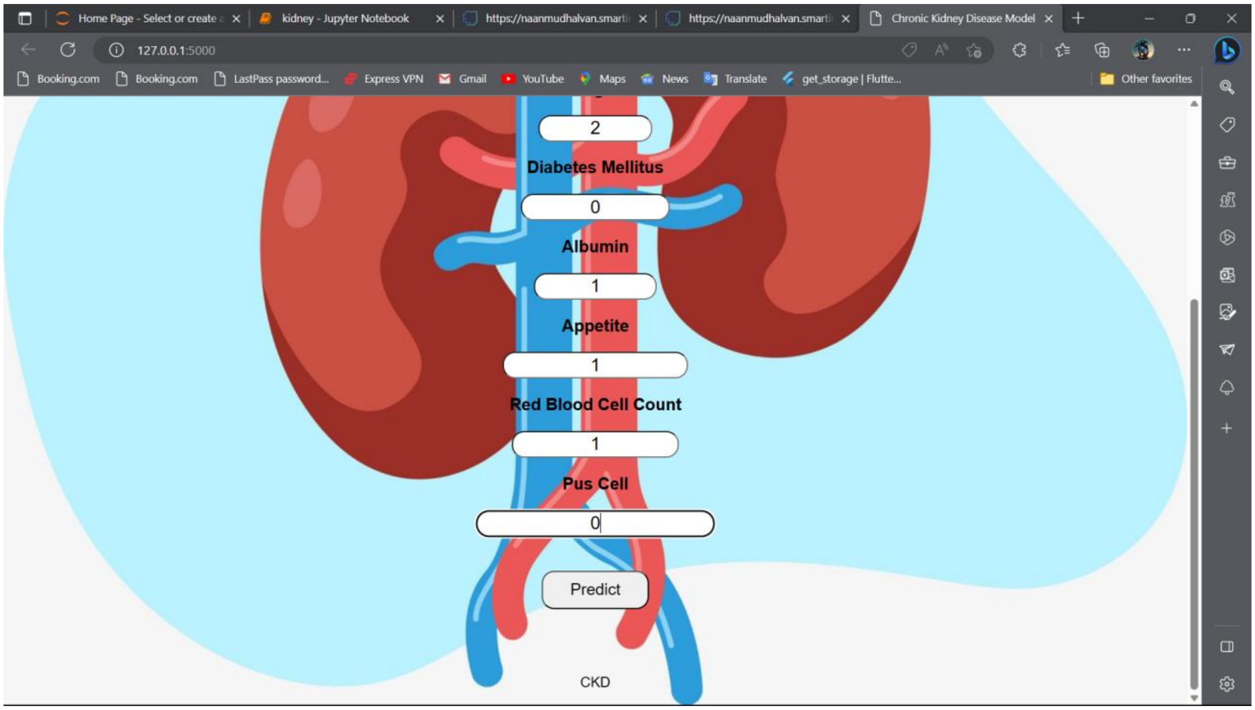
Albumin

(0,1,2,3,4,5)

Appetite

Good= 1, Poor= 0

Red Blood Cell Count


Home Page - Select or create a
kidney - Jupyter Notebook
https://naanmudhalvan.smarti
https://naanmudhalvan.smarti
Chronic Kidney Disease Model
127.0.0.1:5000
Booking.com
Booking.com
LastPass password...
Express VPN
Gmail
YouTube
Maps
News
Translate
get_storage | Flutte...
Other favorites

2

Diabetes Mellitus

0

Albumin

1

Appetite

1

Red Blood Cell Count

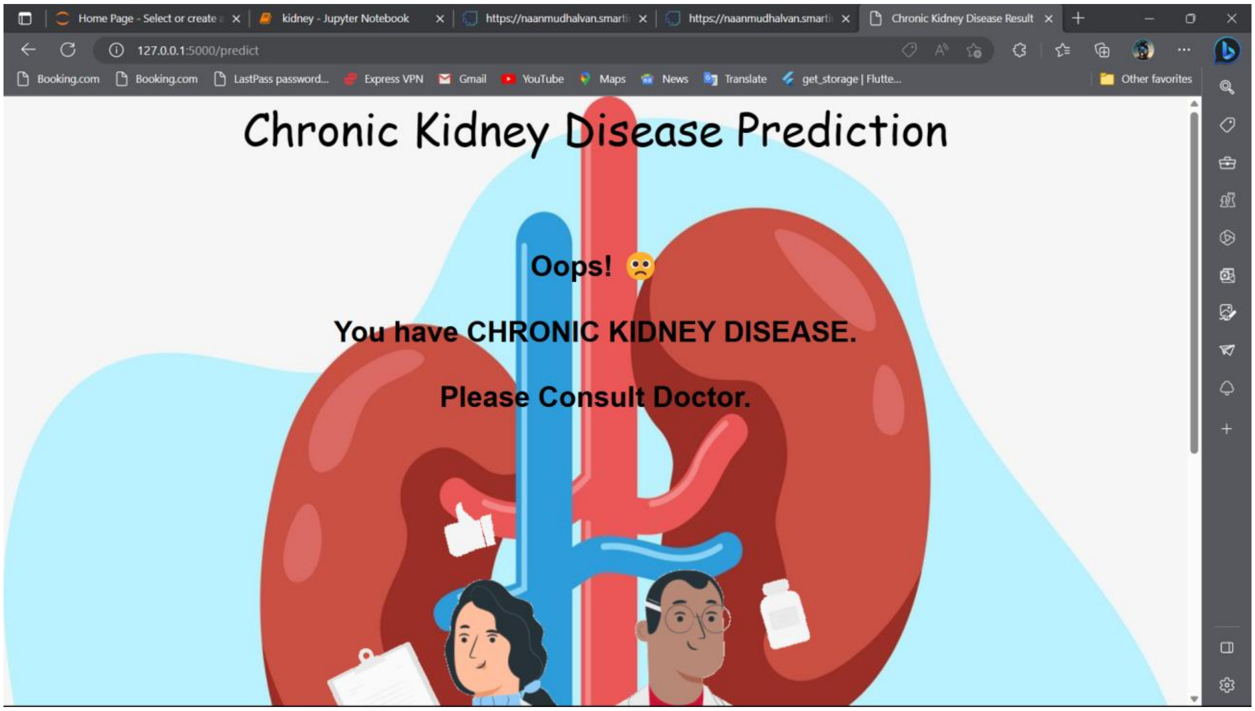
1

Pus Cell

0

Predict

CKD



Chronic Kidney Disease Prediction

Oops! 🙁

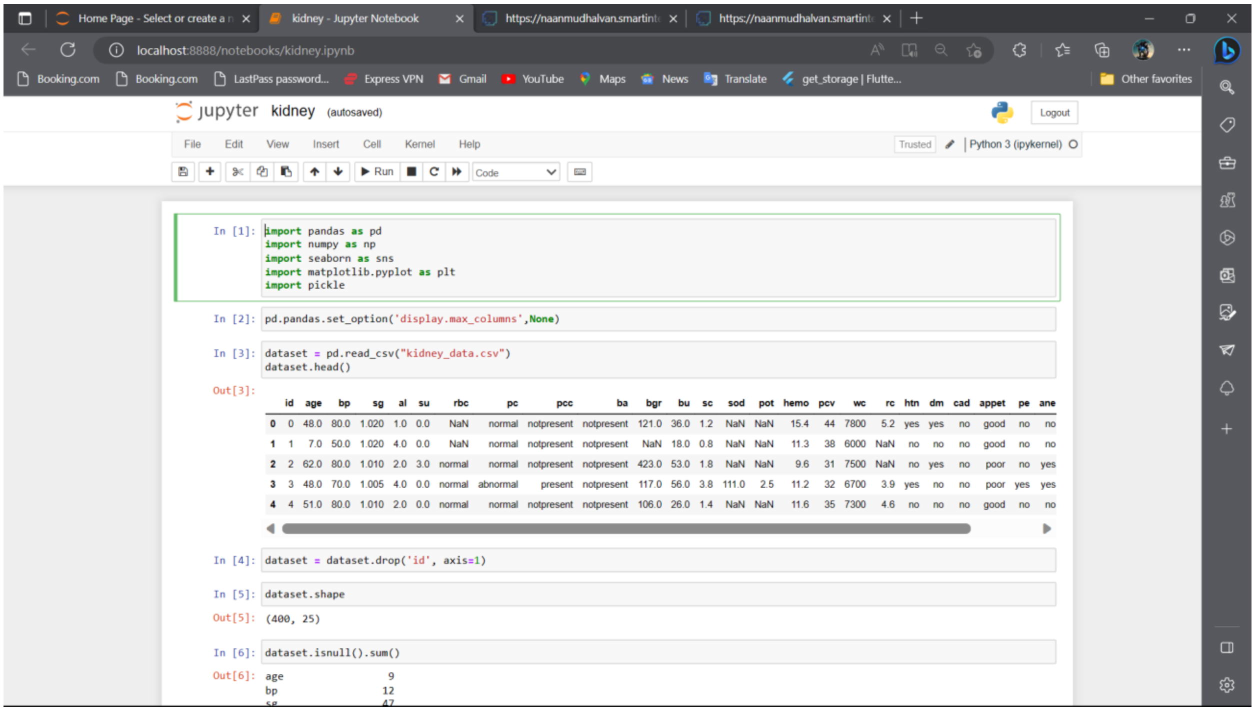
You have CHRONIC KIDNEY DISEASE.

Please Consult Doctor.

Appendix

Source Code:

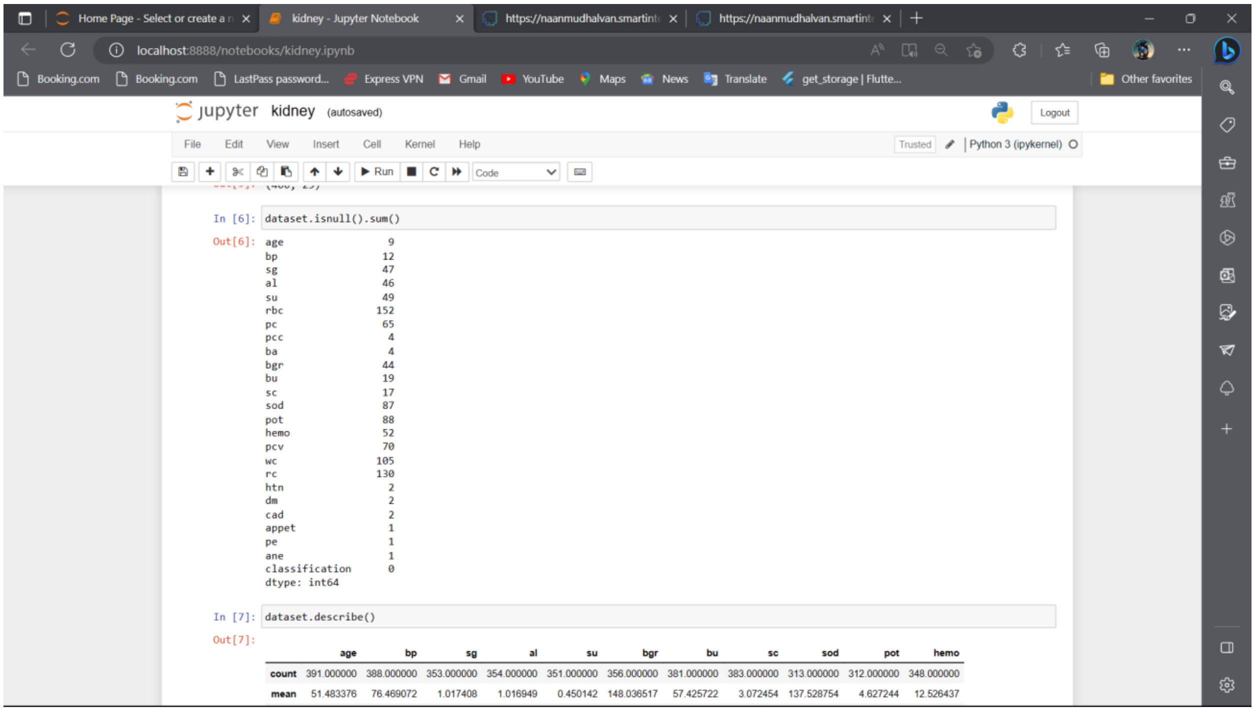
kidney.ipynb



```
In [1]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
        import pickle

In [2]: pd.pandas.set_option('display.max_columns',None)

In [3]: dataset = pd.read_csv("kidney_data.csv")
        dataset.head()

Out[3]:
```

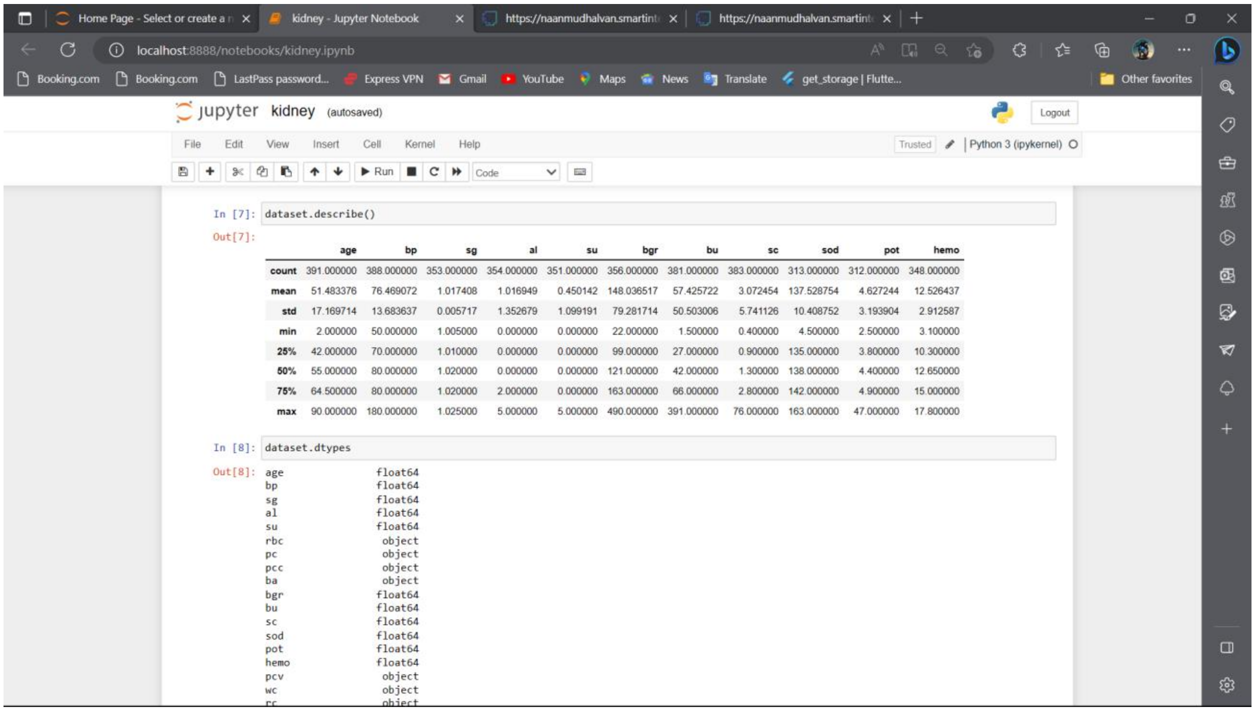
	id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	36.0	1.2	NaN	NaN	15.4	44	7800	5.2	yes	yes	no	good	no	no
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	18.0	0.8	NaN	NaN	11.3	38	6000	NaN	no	no	no	good	no	no
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	53.0	1.8	NaN	NaN	9.6	31	7500	NaN	no	yes	no	poor	no	yes
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	56.0	3.8	111.0	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes	yes
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	26.0	1.4	NaN	NaN	11.6	35	7300	4.6	no	no	no	good	no	no

```
In [4]: dataset = dataset.drop('id', axis=1)

In [5]: dataset.shape

Out[5]: (400, 25)

In [6]: dataset.isnull().sum()

Out[6]: age              9
        bp              12
        sg              47
```



```
        (400, 25)

In [6]: dataset.isnull().sum()

Out[6]: age                9
        bp                12
        sg                47
        al                46
        su                49
        rbc              152
        pc                65
        pcc                4
        ba                 4
        bgr               44
        bu                19
        sc                17
        sod               87
        pot               88
        hemo              52
        pcv               70
        wc               105
        rc               130
        htn                2
        dm                 2
        cad                2
        appet              1
        pe                 1
        ane                1
        classification     0
        dtype: int64

In [7]: dataset.describe()

Out[7]:
```

	age	bp	sg	al	su	bgr	bu	sc	sod	pot	hemo
count	391.000000	388.000000	353.000000	354.000000	351.000000	356.000000	381.000000	383.000000	313.000000	312.000000	348.000000
mean	51.483376	76.469072	1.017408	1.016949	0.450142	148.036517	57.425722	3.072454	137.528754	4.627244	12.526437



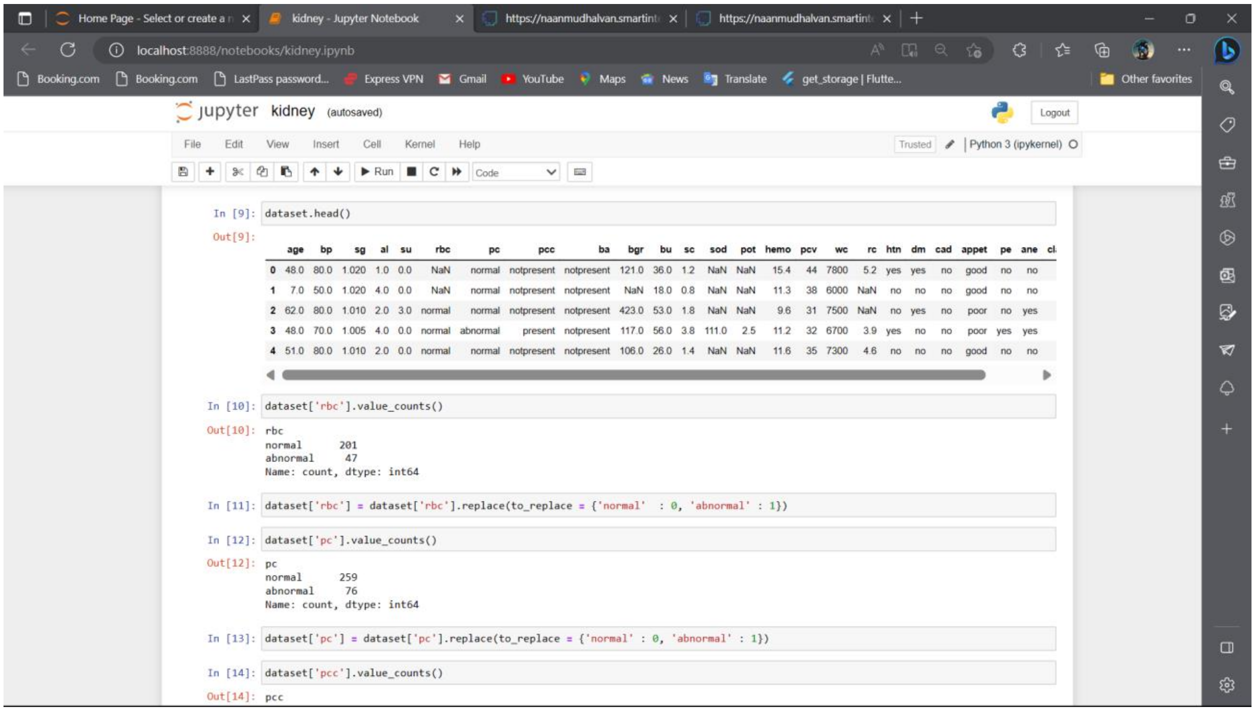
In [7]:
```
dataset.describe()
```

Out[7]:

	age	bp	sg	al	su	bgr	bu	sc	sod	pot	hemo
count	391.000000	388.000000	353.000000	354.000000	351.000000	356.000000	381.000000	383.000000	313.000000	312.000000	348.000000
mean	51.483376	76.469072	1.017408	1.016949	0.450142	148.036517	57.425722	3.072454	137.528754	4.627244	12.526437
std	17.169714	13.683637	0.005717	1.352679	1.099191	79.281714	50.503006	5.741126	10.408752	3.193904	2.912587
min	2.000000	50.000000	1.005000	0.000000	0.000000	22.000000	1.500000	0.400000	4.500000	2.500000	3.100000
25%	42.000000	70.000000	1.010000	0.000000	0.000000	99.000000	27.000000	0.900000	135.000000	3.800000	10.300000
50%	55.000000	80.000000	1.020000	0.000000	0.000000	121.000000	42.000000	1.300000	138.000000	4.400000	12.650000
75%	64.500000	80.000000	1.020000	2.000000	0.000000	163.000000	66.000000	2.800000	142.000000	4.900000	15.000000
max	90.000000	180.000000	1.025000	5.000000	5.000000	490.000000	391.000000	76.000000	163.000000	47.000000	17.800000

In [8]:
```
dataset.dtypes
```

Out[8]:
```
age            float64
bp             float64
sg             float64
al             float64
su             float64
rbc             object
pc              object
pcc             object
ba              object
bgr            float64
bu             float64
sc             float64
sod            float64
pot            float64
hemo           float64
pcv             object
wc              object
rc              object
```



In [9]:
```
dataset.head()
```

Out[9]:

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	cl
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	36.0	1.2	NaN	NaN	15.4	44	7800	5.2	yes	yes	no	good	no	no	
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	18.0	0.8	NaN	NaN	11.3	38	6000	NaN	no	no	no	good	no	no	
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	53.0	1.8	NaN	NaN	9.6	31	7500	NaN	no	yes	no	poor	no	yes	
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	56.0	3.8	111.0	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes	yes	
4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	26.0	1.4	NaN	NaN	11.6	35	7300	4.6	no	no	no	good	no	no	

In [10]:
```
dataset['rbc'].value_counts()
```

Out[10]:
```
rbc
normal      201
abnormal     47
Name: count, dtype: int64
```

In [11]:
```
dataset['rbc'] = dataset['rbc'].replace(to_replace = {'normal'  : 0, 'abnormal' : 1})
```

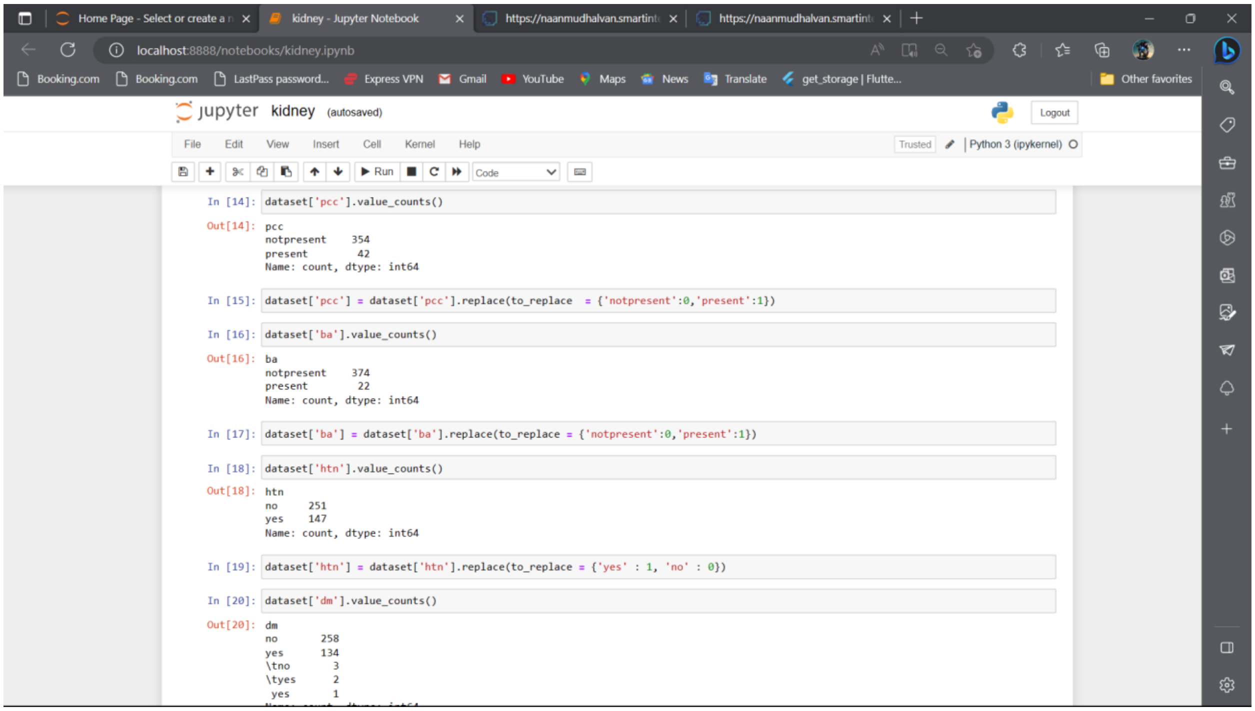
In [12]:
```
dataset['pc'].value_counts()
```

Out[12]:
```
pc
normal      259
abnormal     76
Name: count, dtype: int64
```

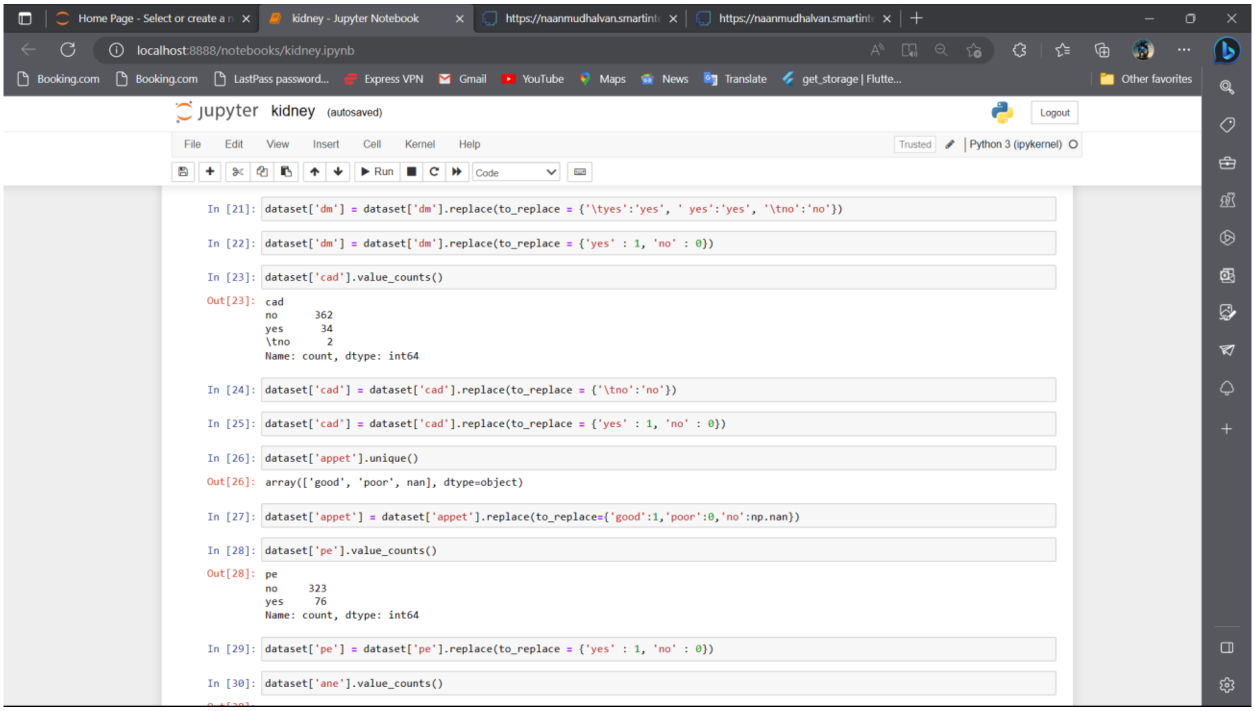
In [13]:
```
dataset['pc'] = dataset['pc'].replace(to_replace = {'normal' : 0, 'abnormal' : 1})
```

In [14]:
```
dataset['pcc'].value_counts()
```

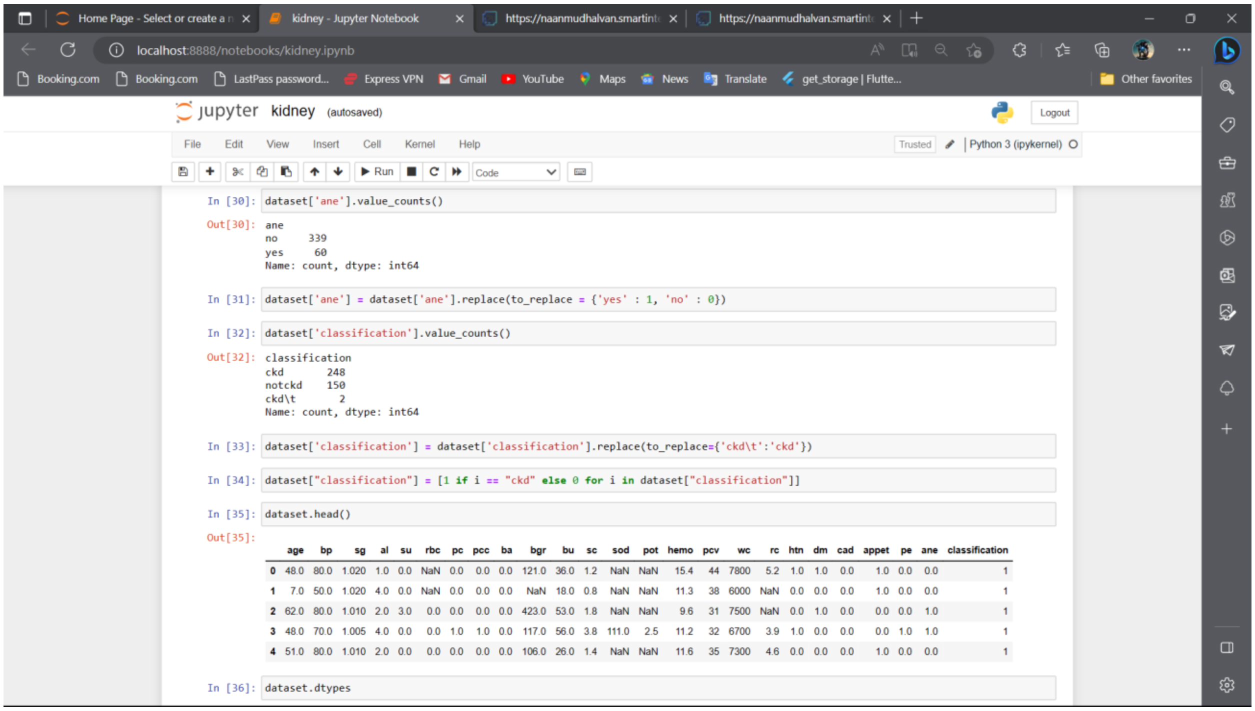
Out[14]:
```
pcc
```



```
In [14]: dataset['pcc'].value_counts()

Out[14]: pcc
         notpresent    354
         present        42
         Name: count, dtype: int64

In [15]: dataset['pcc'] = dataset['pcc'].replace(to_replace  = {'notpresent':0,'present':1})

In [16]: dataset['ba'].value_counts()

Out[16]: ba
         notpresent    374
         present        22
         Name: count, dtype: int64

In [17]: dataset['ba'] = dataset['ba'].replace(to_replace = {'notpresent':0,'present':1})

In [18]: dataset['htn'].value_counts()

Out[18]: htn
         no     251
         yes    147
         Name: count, dtype: int64

In [19]: dataset['htn'] = dataset['htn'].replace(to_replace = {'yes' : 1, 'no' : 0})

In [20]: dataset['dm'].value_counts()

Out[20]: dm
         no       258
         yes      134
         \tno       3
         \tyes      2
          yes       1
```



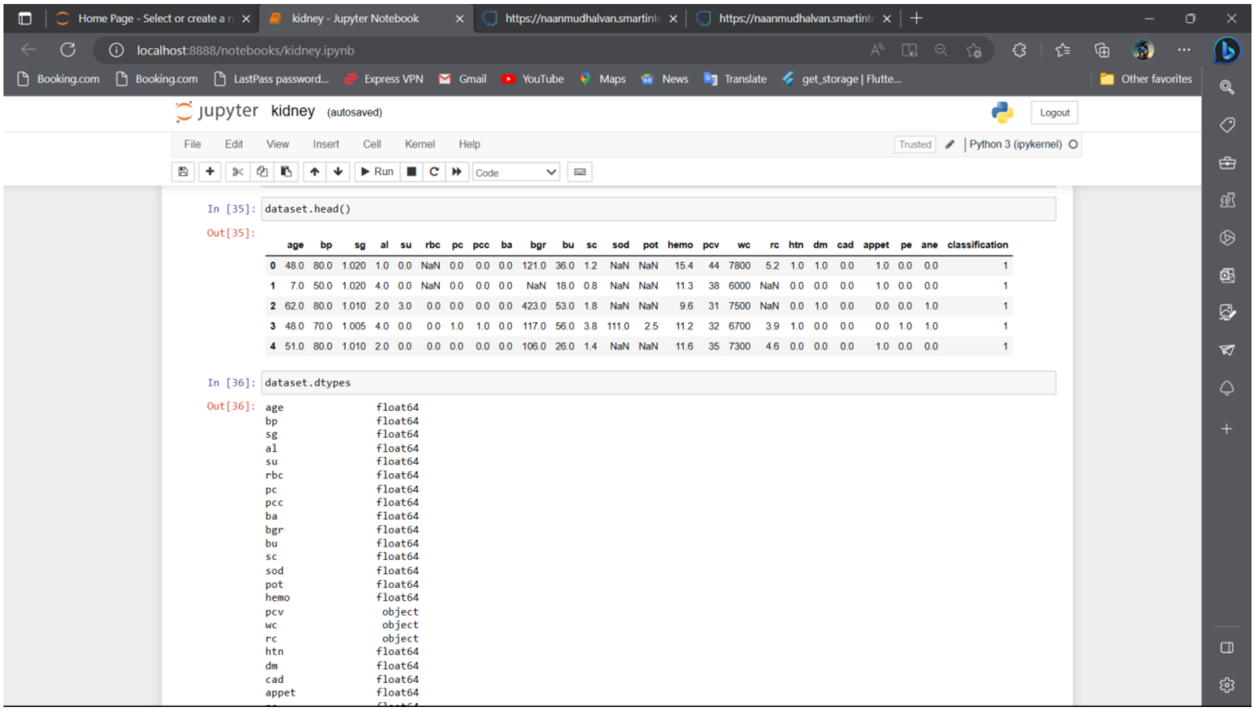
```
In [21]: dataset['dm'] = dataset['dm'].replace(to_replace = {'\tyes':'yes', ' yes':'yes', '\tno':'no'})

In [22]: dataset['dm'] = dataset['dm'].replace(to_replace = {'yes' : 1, 'no' : 0})

In [23]: dataset['cad'].value_counts()

Out[23]: cad
         no      362
         yes      34
         \tno      2
         Name: count, dtype: int64

In [24]: dataset['cad'] = dataset['cad'].replace(to_replace = {'\tno':'no'})

In [25]: dataset['cad'] = dataset['cad'].replace(to_replace = {'yes' : 1, 'no' : 0})

In [26]: dataset['appet'].unique()

Out[26]: array(['good', 'poor', nan], dtype=object)

In [27]: dataset['appet'] = dataset['appet'].replace(to_replace={'good':1,'poor':0,'no':np.nan})

In [28]: dataset['pe'].value_counts()

Out[28]: pe
         no     323
         yes     76
         Name: count, dtype: int64

In [29]: dataset['pe'] = dataset['pe'].replace(to_replace = {'yes' : 1, 'no' : 0})

In [30]: dataset['ane'].value_counts()
```



```
In [30]: dataset['ane'].value_counts()

Out[30]: ane
         no     339
         yes     60
         Name: count, dtype: int64

In [31]: dataset['ane'] = dataset['ane'].replace(to_replace = {'yes' : 1, 'no' : 0})

In [32]: dataset['classification'].value_counts()

Out[32]: classification
         ckd       248
         notckd    150
         ckd\t       2
         Name: count, dtype: int64

In [33]: dataset['classification'] = dataset['classification'].replace(to_replace={'ckd\t':'ckd'})

In [34]: dataset["classification"] = [1 if i == "ckd" else 0 for i in dataset["classification"]]

In [35]: dataset.head()

Out[35]:
```

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	48.0	80.0	1.020	1.0	0.0	NaN	0.0	0.0	0.0	121.0	36.0	1.2	NaN	NaN	15.4	44	7800	5.2	1.0	1.0	0.0	1.0	0.0	0.0	1
1	7.0	50.0	1.020	4.0	0.0	NaN	0.0	0.0	0.0	NaN	18.0	0.8	NaN	NaN	11.3	38	6000	NaN	0.0	0.0	0.0	1.0	0.0	0.0	1
2	62.0	80.0	1.010	2.0	3.0	0.0	0.0	0.0	0.0	423.0	53.0	1.8	NaN	NaN	9.6	31	7500	NaN	0.0	1.0	0.0	0.0	0.0	1.0	1
3	48.0	70.0	1.005	4.0	0.0	0.0	1.0	1.0	0.0	117.0	56.0	3.8	111.0	2.5	11.2	32	6700	3.9	1.0	0.0	0.0	0.0	1.0	1.0	1
4	51.0	80.0	1.010	2.0	0.0	0.0	0.0	0.0	0.0	106.0	26.0	1.4	NaN	NaN	11.6	35	7300	4.6	0.0	0.0	0.0	1.0	0.0	0.0	1

```
In [36]: dataset.dtypes
```



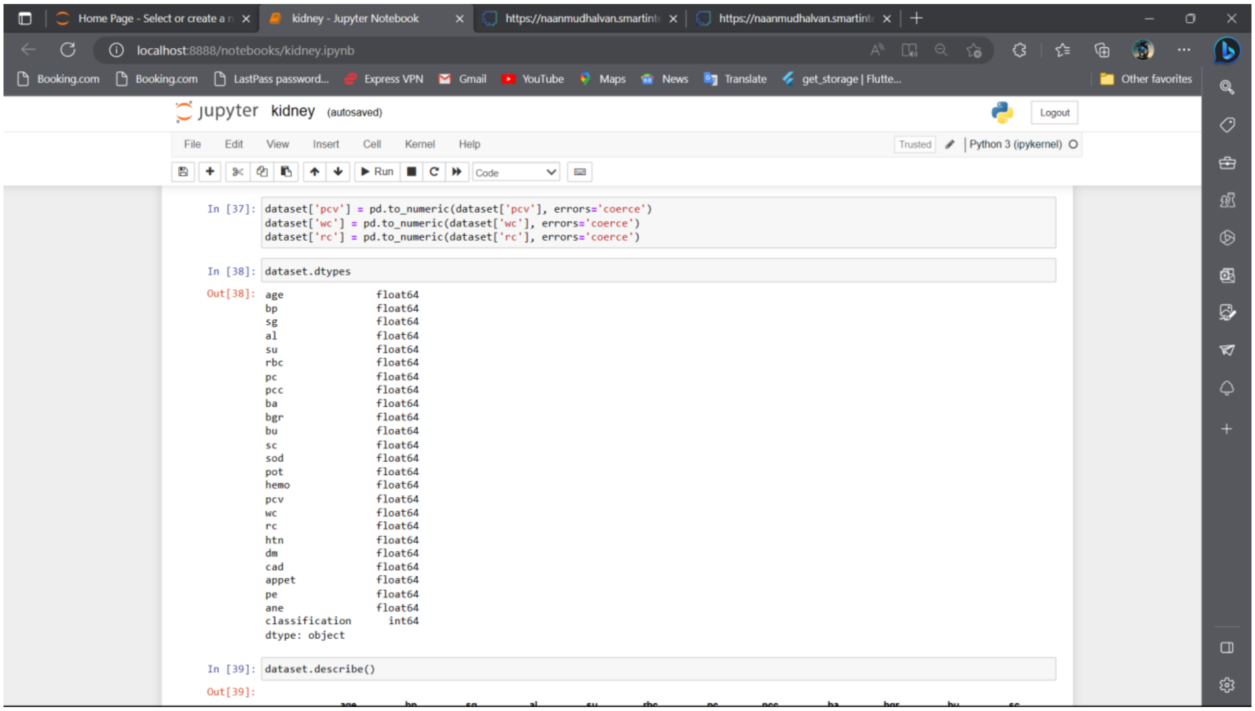
In [35]: dataset.head()

Out[35]:

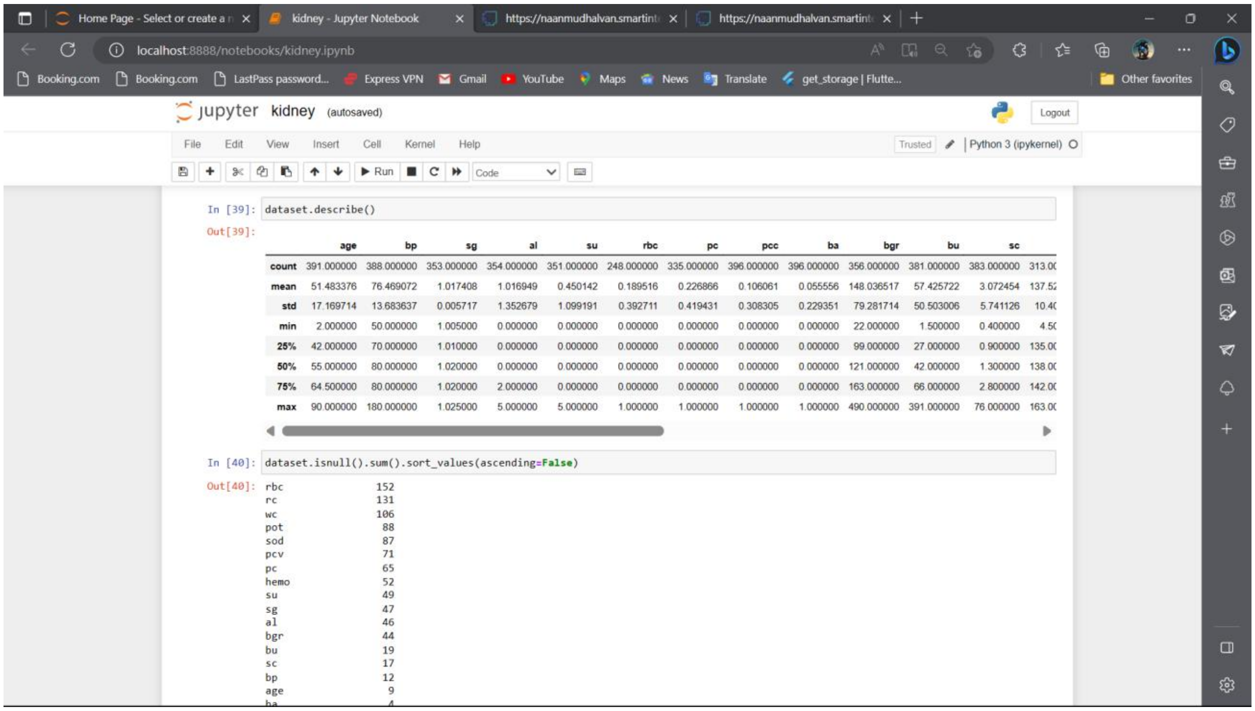
	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	48.0	80.0	1.020	1.0	0.0	NaN	0.0	0.0	0.0	121.0	36.0	1.2	NaN	NaN	15.4	44	7800	5.2	1.0	1.0	0.0	1.0	0.0	0.0	1
1	7.0	50.0	1.020	4.0	0.0	NaN	0.0	0.0	0.0	NaN	18.0	0.8	NaN	NaN	11.3	38	6000	NaN	0.0	0.0	0.0	1.0	0.0	0.0	1
2	62.0	80.0	1.010	2.0	3.0	0.0	0.0	0.0	0.0	423.0	53.0	1.8	NaN	NaN	9.6	31	7500	NaN	0.0	1.0	0.0	0.0	0.0	1.0	1
3	48.0	70.0	1.005	4.0	0.0	0.0	1.0	1.0	0.0	117.0	56.0	3.8	111.0	2.5	11.2	32	6700	3.9	1.0	0.0	0.0	0.0	1.0	1.0	1
4	51.0	80.0	1.010	2.0	0.0	0.0	0.0	0.0	0.0	106.0	26.0	1.4	NaN	NaN	11.6	35	7300	4.6	0.0	0.0	0.0	1.0	0.0	0.0	1

In [36]: dataset.dtypes

```
Out[36]: age               float64
         bp                float64
         sg                float64
         al                float64
         su                float64
         rbc               float64
         pc                float64
         pcc               float64
         ba                float64
         bgr               float64
         bu                float64
         sc                float64
         sod               float64
         pot               float64
         hemo              float64
         pcv                object
         wc                 object
         rc                 object
         htn               float64
         dm                float64
         cad               float64
         appet             float64
```


Home Page - Select or create a n
kidney - Jupyter Notebook
https://naanmudhalvan.smartint
localhost:8888/notebooks/kidney.ipynb
jupyter kidney (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code

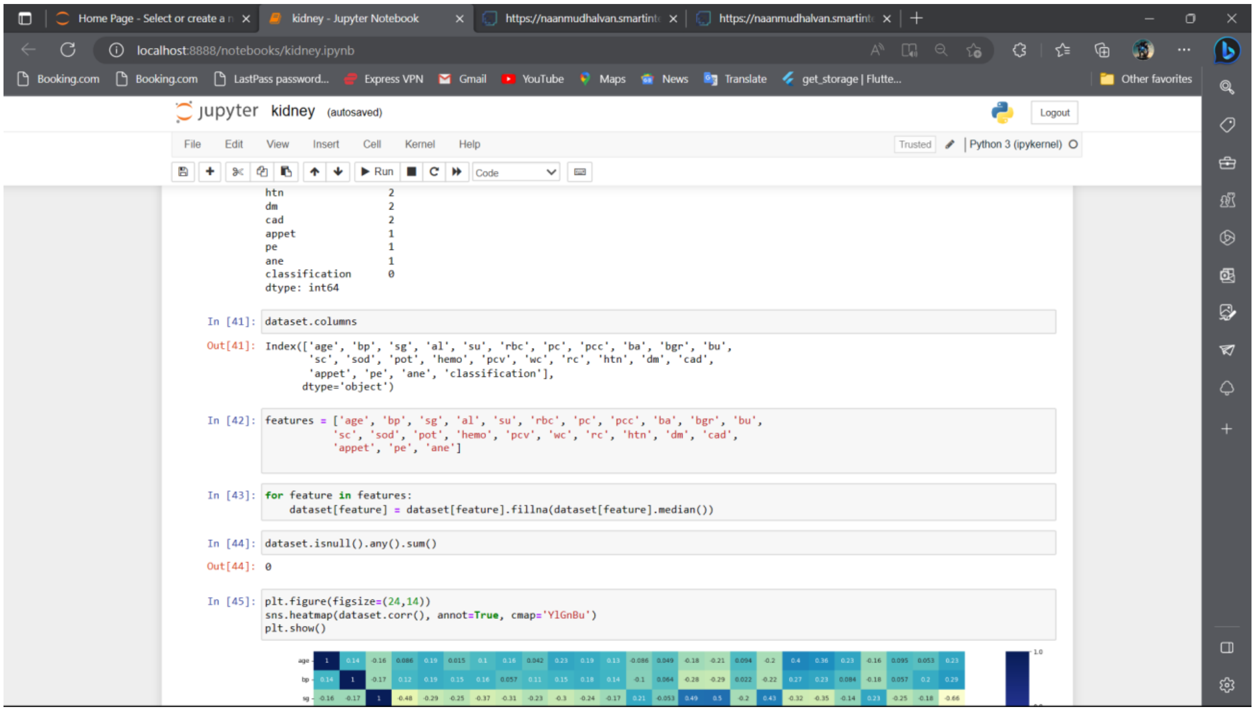
```
In [37]: dataset['pcv'] = pd.to_numeric(dataset['pcv'], errors='coerce')
         dataset['wc'] = pd.to_numeric(dataset['wc'], errors='coerce')
         dataset['rc'] = pd.to_numeric(dataset['rc'], errors='coerce')

In [38]: dataset.dtypes

Out[38]: age               float64
         bp                float64
         sg                float64
         al                float64
         su                float64
         rbc               float64
         pc                float64
         pcc               float64
         ba                float64
         bgr               float64
         bu                float64
         sc                float64
         sod               float64
         pot               float64
         hemo              float64
         pcv               float64
         wc                float64
         rc                float64
         htn               float64
         dm                float64
         cad               float64
         appet             float64
         pe                float64
         ane               float64
         classification      int64
         dtype: object

In [39]: dataset.describe()

Out[39]:
```


Home Page - Select or create a n
kidney - Jupyter Notebook
https://naanmudhalvan.smartint
localhost:8888/notebooks/kidney.ipynb
jupyter kidney (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [39]: dataset.describe()

Out[39]:
```

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	
count	391.000000	388.000000	353.000000	354.000000	351.000000	248.000000	335.000000	396.000000	396.000000	356.000000	381.000000	383.000000	313.0
mean	51.483376	76.469072	1.017408	1.016949	0.450142	0.189516	0.226866	0.106061	0.055556	148.036517	57.425722	3.072454	137.52
std	17.169714	13.683637	0.005717	1.352679	1.099191	0.392711	0.419431	0.308305	0.229351	79.281714	50.503006	5.741126	10.4
min	2.000000	50.000000	1.005000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	22.000000	1.500000	0.400000	4.5
25%	42.000000	70.000000	1.010000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	99.000000	27.000000	0.900000	135.0
50%	55.000000	80.000000	1.020000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	121.000000	42.000000	1.300000	138.0
75%	64.500000	80.000000	1.020000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	163.000000	66.000000	2.800000	142.0
max	90.000000	180.000000	1.025000	5.000000	5.000000	1.000000	1.000000	1.000000	1.000000	490.000000	391.000000	76.000000	163.0

```
In [40]: dataset.isnull().sum().sort_values(ascending=False)

Out[40]: rbc               152
         rc                131
         wc                106
         pot                88
         sod                87
         pcv                71
         pc                 65
         hemo               52
         su                 49
         sg                 47
         al                 46
         bgr                44
         bu                 19
         sc                 17
         bp                 12
         age                 9
         ba                  4
```


Home Page - Select or create a n
kidney - Jupyter Notebook
https://naanmudhalvan.smartint
localhost:8888/notebooks/kidney.ipynb
jupyter kidney (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code

```
htn               2
dm                2
cad               2
appet             1
pe                1
ane               1
classification    0
dtype: int64
```

In [41]:
```
dataset.columns
```

Out[41]:
```
Index(['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',
       'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',
       'appet', 'pe', 'ane', 'classification'],
      dtype='object')
```

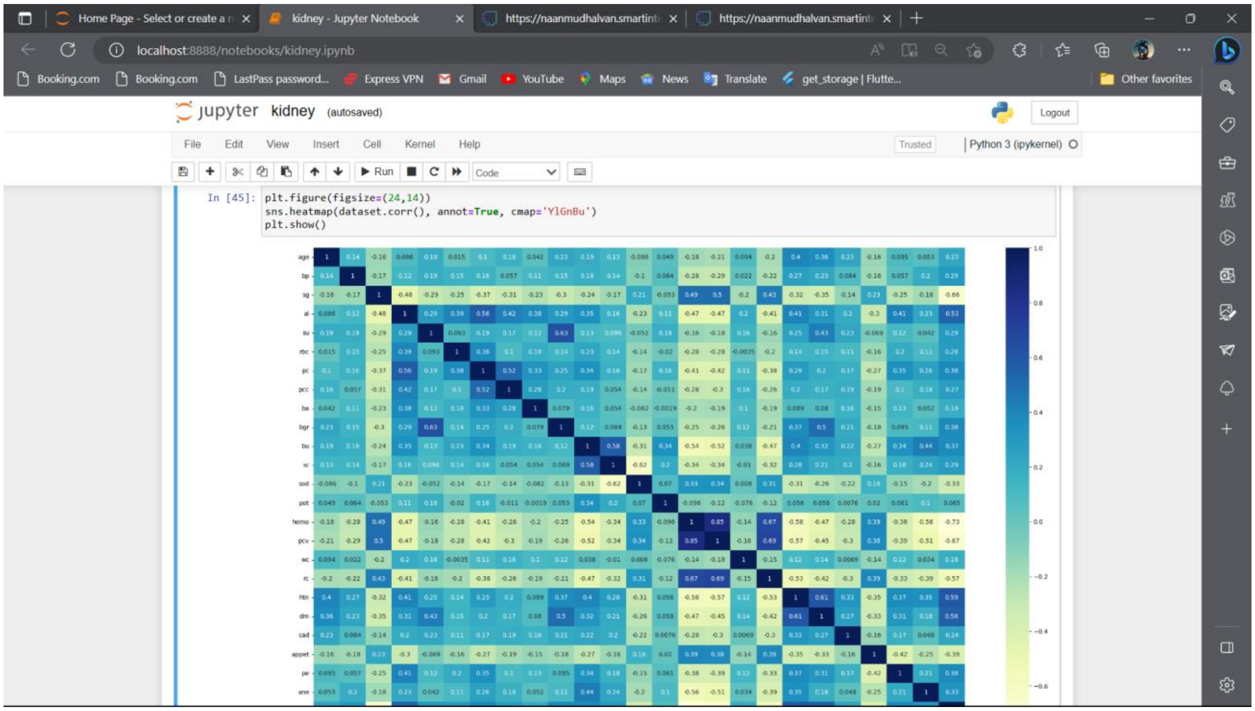
In [42]:
```
features = ['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',
            'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',
            'appet', 'pe', 'ane']
```

In [43]:
```
for feature in features:
    dataset[feature] = dataset[feature].fillna(dataset[feature].median())
```

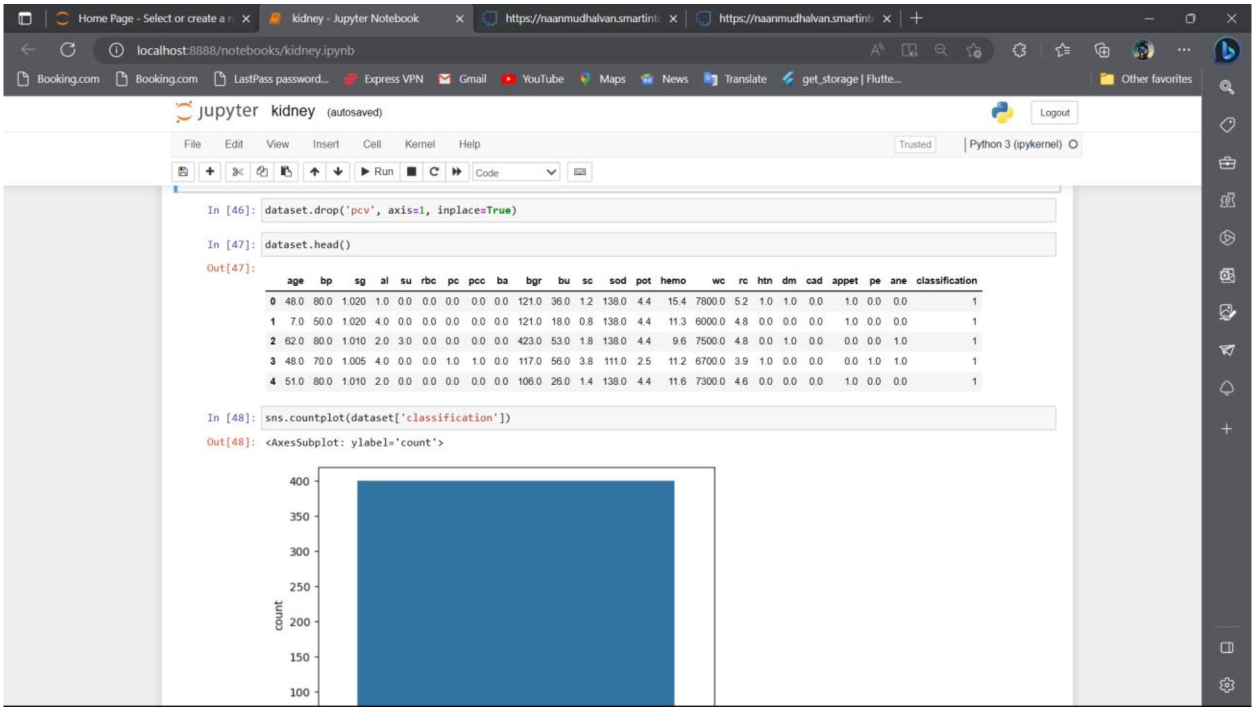
In [44]:
```
dataset.isnull().any().sum()
```

Out[44]:
```
0
```

In [45]:
```
plt.figure(figsize=(24,14))
sns.heatmap(dataset.corr(), annot=True, cmap='YlGnBu')
plt.show()
```


Home Page - Select or create a n
kidney - Jupyter Notebook
https://naanmudhalvan.smartint
localhost:8888/notebooks/kidney.ipynb
jupyter kidney (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code
age
bp
sg
al
su
rbc
pc
pcc
ba
bgr
bu
sc
sod
pot
hemo
pcv
wc
rc
htn
dm
cad
appet
pe
ane
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

In [45]:
```
plt.figure(figsize=(24,14))
sns.heatmap(dataset.corr(), annot=True, cmap='YlGnBu')
plt.show()
```



jupyter kidney (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code

In [46]:
```python
dataset.drop('pcv', axis=1, inplace=True)
```

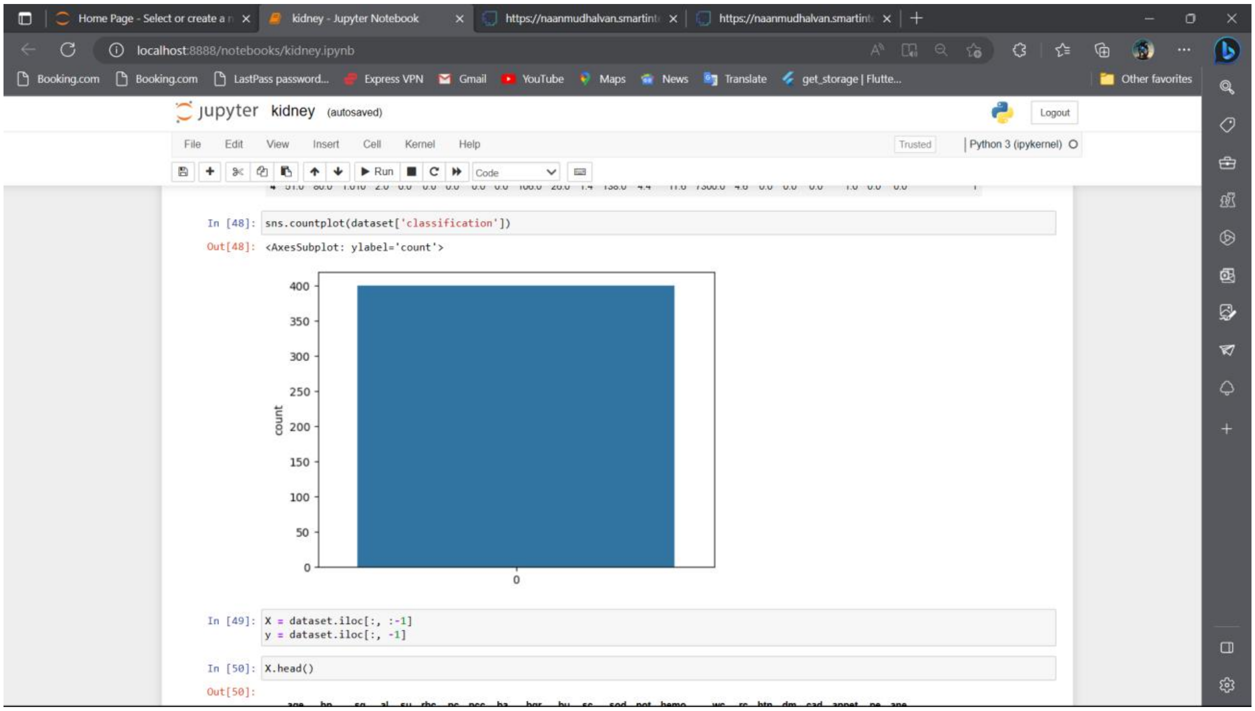
In [47]:
```python
dataset.head()
```

Out[47]:

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	48.0	80.0	1.020	1.0	0.0	0.0	0.0	0.0	0.0	121.0	36.0	1.2	138.0	4.4	15.4	7800.0	5.2	1.0	1.0	0.0	1.0	0.0	0.0	1
1	7.0	50.0	1.020	4.0	0.0	0.0	0.0	0.0	0.0	121.0	18.0	0.8	138.0	4.4	11.3	6000.0	4.8	0.0	0.0	0.0	1.0	0.0	0.0	1
2	62.0	80.0	1.010	2.0	3.0	0.0	0.0	0.0	0.0	423.0	53.0	1.8	138.0	4.4	9.6	7500.0	4.8	0.0	1.0	0.0	0.0	0.0	1.0	1
3	48.0	70.0	1.005	4.0	0.0	0.0	1.0	1.0	0.0	117.0	56.0	3.8	111.0	2.5	11.2	6700.0	3.9	1.0	0.0	0.0	0.0	1.0	1.0	1
4	51.0	80.0	1.010	2.0	0.0	0.0	0.0	0.0	0.0	106.0	26.0	1.4	138.0	4.4	11.6	7300.0	4.6	0.0	0.0	0.0	1.0	0.0	0.0	1

In [48]:
```python
sns.countplot(dataset['classification'])
```

Out[48]:
```
<AxesSubplot: ylabel='count'>
```



jupyter kidney (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code

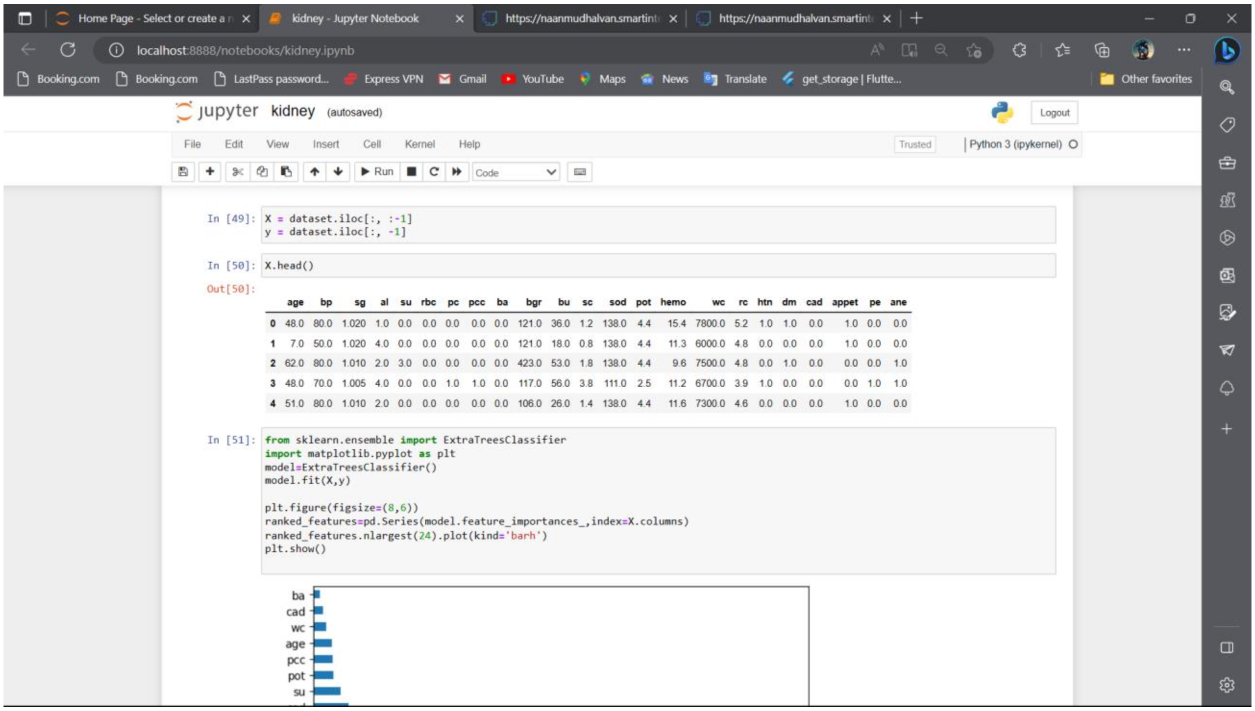
In [48]:
```python
sns.countplot(dataset['classification'])
```

Out[48]:
```
<AxesSubplot: ylabel='count'>
```

In [49]:
```python
X = dataset.iloc[:, :-1]
y = dataset.iloc[:, -1]
```

In [50]:
```python
X.head()
```

Out[50]:



In [49]:

```
X = dataset.iloc[:, :-1]
y = dataset.iloc[:, -1]
```

In [50]:

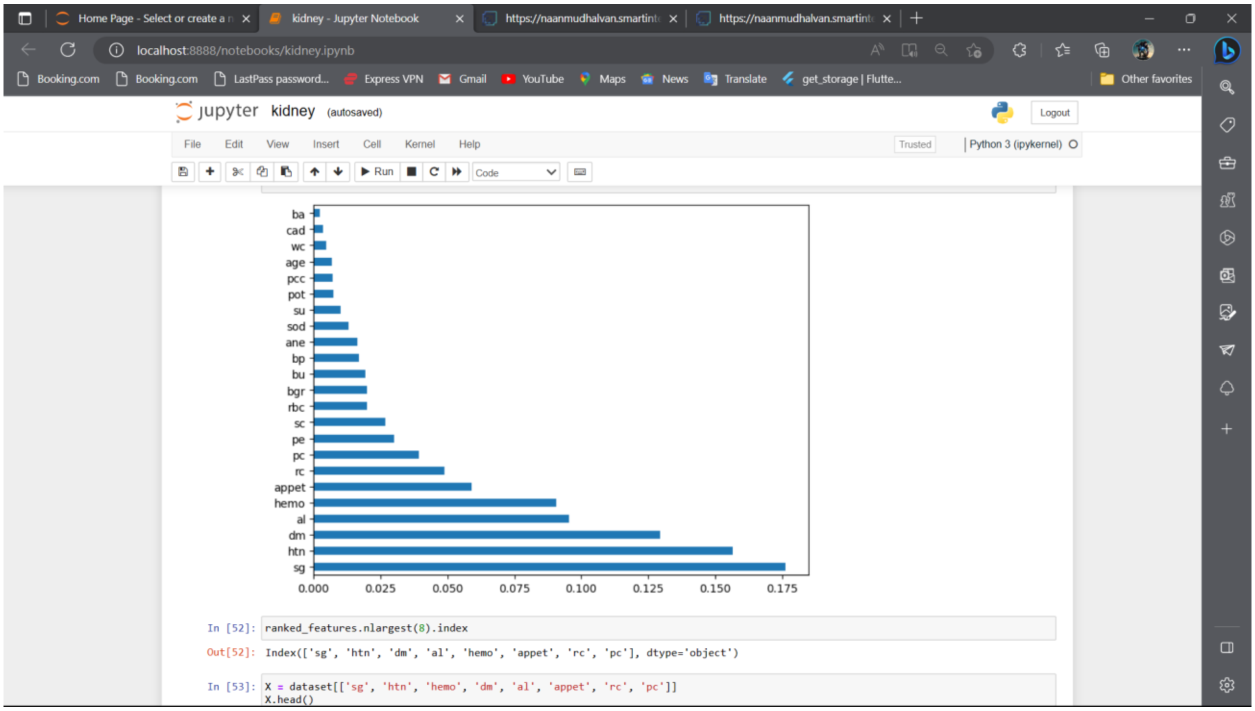
```
X.head()
```

Out[50]:

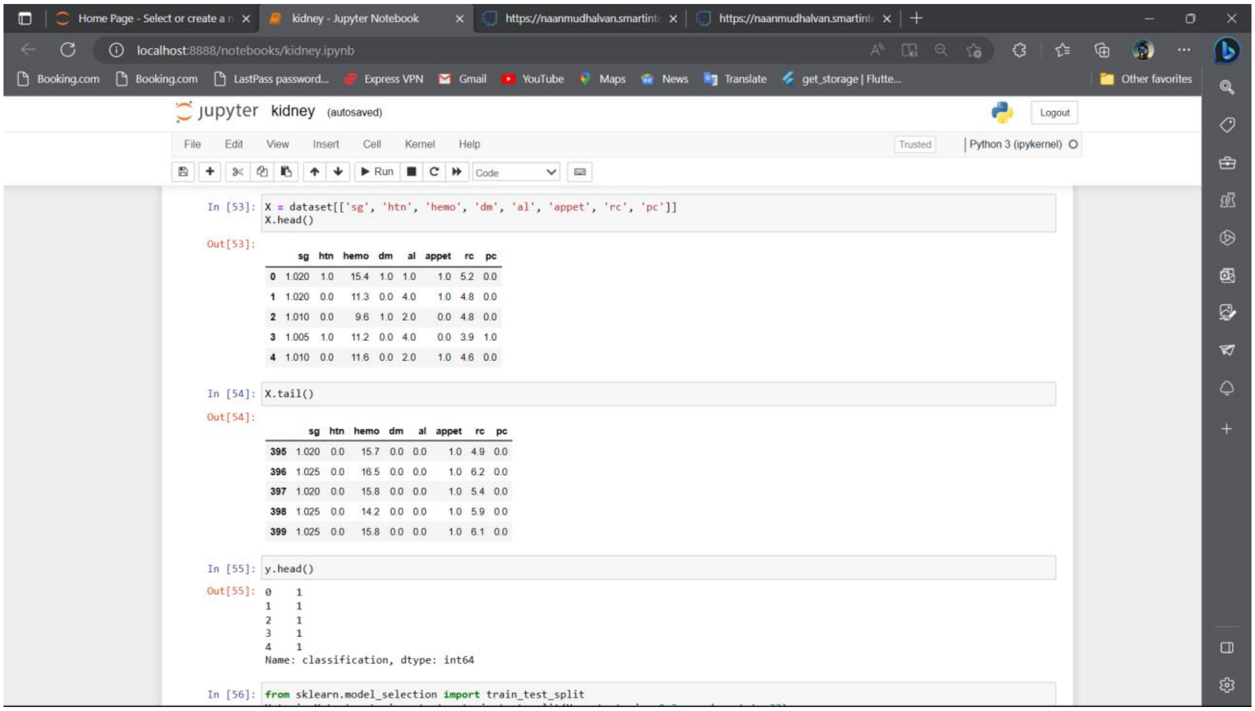
	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	wc	rc	htn	dm	cad	appet	pe	ane
0	48.0	80.0	1.020	1.0	0.0	0.0	0.0	0.0	0.0	121.0	36.0	1.2	138.0	4.4	15.4	7800.0	5.2	1.0	1.0	0.0	1.0	0.0	0.0
1	7.0	50.0	1.020	4.0	0.0	0.0	0.0	0.0	0.0	121.0	18.0	0.8	138.0	4.4	11.3	6000.0	4.8	0.0	0.0	0.0	1.0	0.0	0.0
2	62.0	80.0	1.010	2.0	3.0	0.0	0.0	0.0	0.0	423.0	53.0	1.8	138.0	4.4	9.6	7500.0	4.8	0.0	1.0	0.0	0.0	0.0	1.0
3	48.0	70.0	1.005	4.0	0.0	0.0	1.0	1.0	0.0	117.0	56.0	3.8	111.0	2.5	11.2	6700.0	3.9	1.0	0.0	0.0	0.0	1.0	1.0
4	51.0	80.0	1.010	2.0	0.0	0.0	0.0	0.0	0.0	106.0	26.0	1.4	138.0	4.4	11.6	7300.0	4.6	0.0	0.0	0.0	1.0	0.0	0.0

In [51]:

```
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt
model=ExtraTreesClassifier()
model.fit(X,y)

plt.figure(figsize=(8,6))
ranked_features=pd.Series(model.feature_importances_,index=X.columns)
ranked_features.nlargest(24).plot(kind='barh')
plt.show()
```



```
In [52]: ranked_features.nlargest(8).index
Out[52]: Index(['sg', 'htn', 'dm', 'al', 'hemo', 'appet', 'rc', 'pc'], dtype='object')

In [53]: X = dataset[['sg', 'htn', 'hemo', 'dm', 'al', 'appet', 'rc', 'pc']]
         X.head()
```



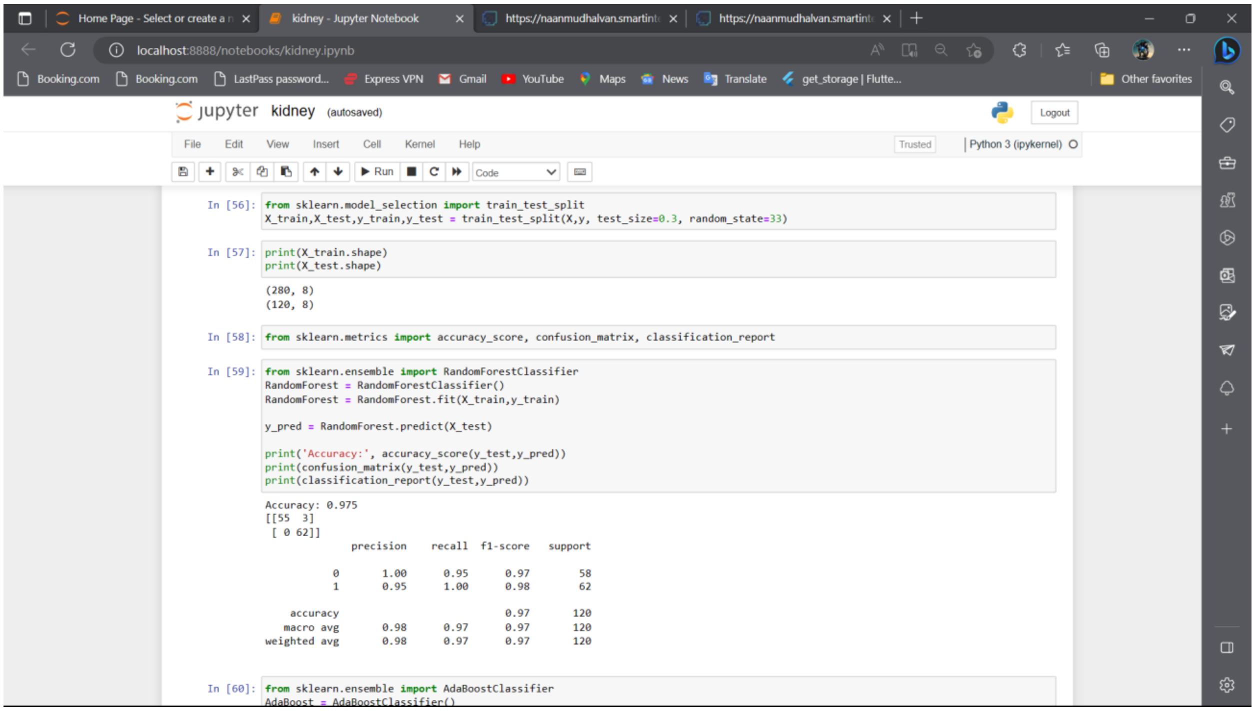
```
In [53]: X = dataset[['sg', 'htn', 'hemo', 'dm', 'al', 'appet', 'rc', 'pc']]
         X.head()
Out[53]:
```

	sg	htn	hemo	dm	al	appet	rc	pc
0	1.020	1.0	15.4	1.0	1.0	1.0	5.2	0.0
1	1.020	0.0	11.3	0.0	4.0	1.0	4.8	0.0
2	1.010	0.0	9.6	1.0	2.0	0.0	4.8	0.0
3	1.005	1.0	11.2	0.0	4.0	0.0	3.9	1.0
4	1.010	0.0	11.6	0.0	2.0	1.0	4.6	0.0

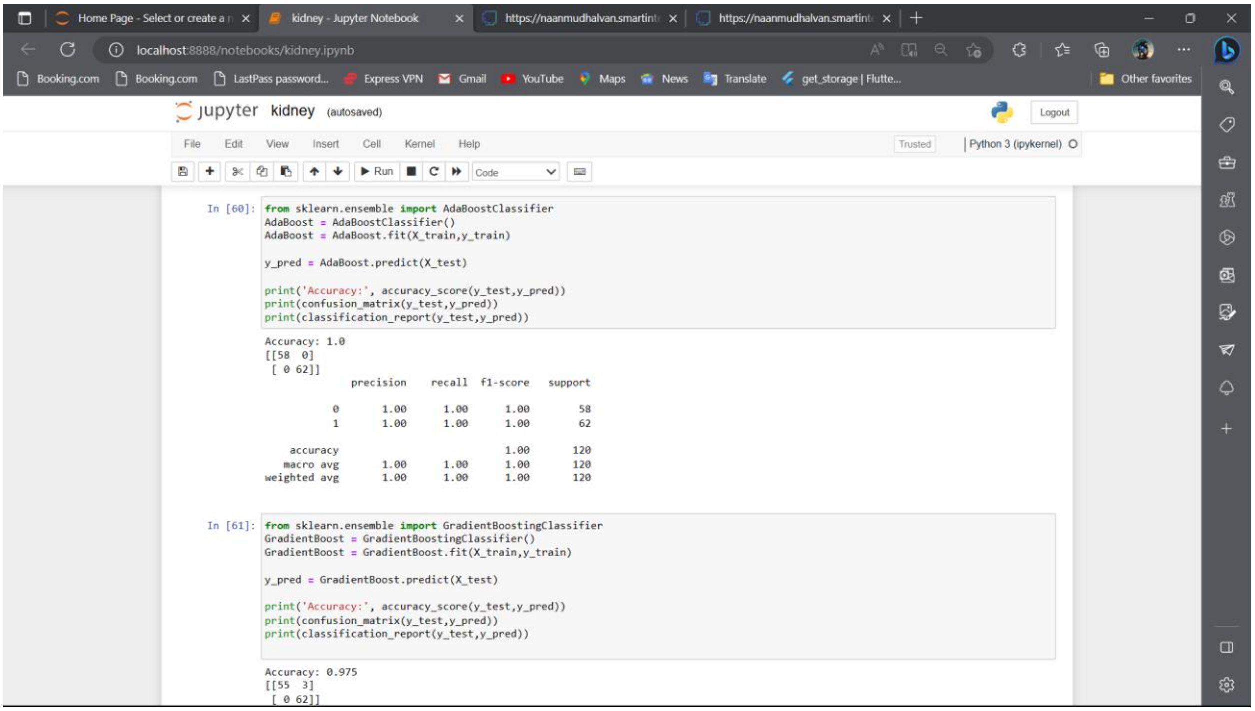
```
In [54]: X.tail()
Out[54]:
```

	sg	htn	hemo	dm	al	appet	rc	pc
395	1.020	0.0	15.7	0.0	0.0	1.0	4.9	0.0
396	1.025	0.0	16.5	0.0	0.0	1.0	6.2	0.0
397	1.020	0.0	15.8	0.0	0.0	1.0	5.4	0.0
398	1.025	0.0	14.2	0.0	0.0	1.0	5.9	0.0
399	1.025	0.0	15.8	0.0	0.0	1.0	6.1	0.0

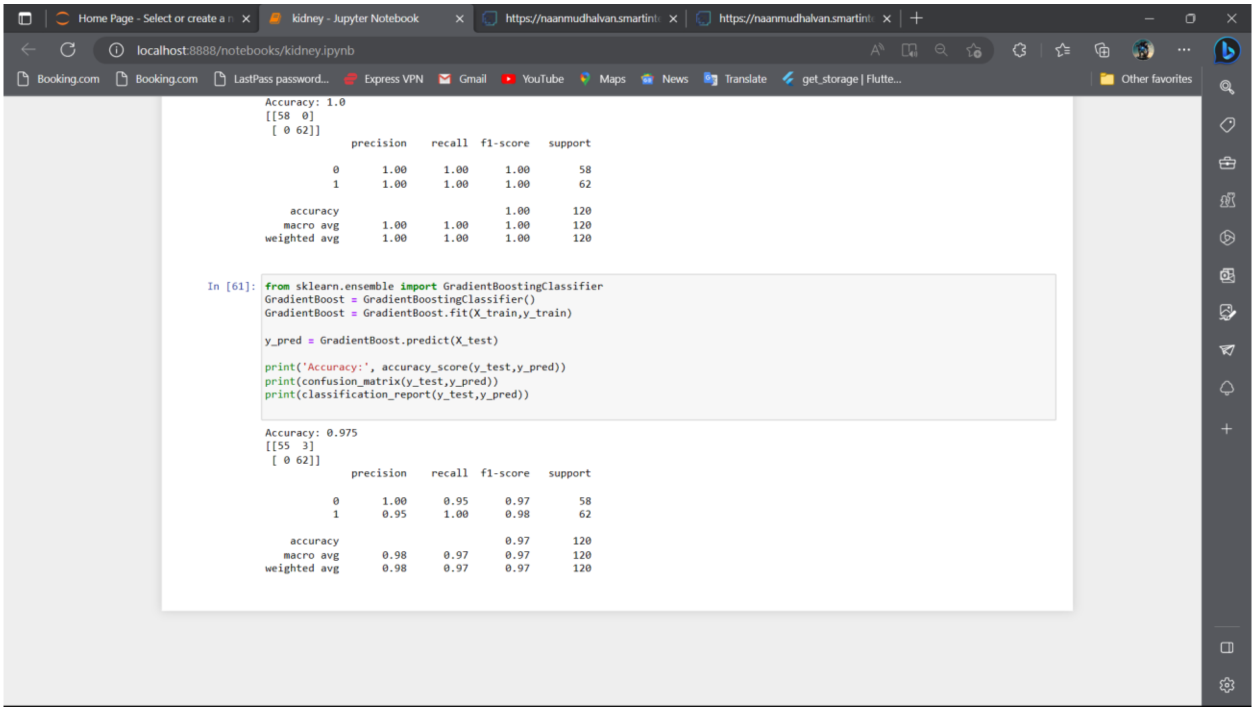
```
In [55]: y.head()
Out[55]: 0    1
         1    1
         2    1
         3    1
         4    1
         Name: classification, dtype: int64

In [56]: from sklearn.model_selection import train_test_split
```


Home Page - Select or create a n
kidney - Jupyter Notebook
https://naanmudhalvan.smartint
localhost:8888/notebooks/kidney.ipynb
jupyter kidney (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [56]: from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.3, random_state=33)

In [57]: print(X_train.shape)
         print(X_test.shape)

         (280, 8)
         (120, 8)

In [58]: from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

In [59]: from sklearn.ensemble import RandomForestClassifier
         RandomForest = RandomForestClassifier()
         RandomForest = RandomForest.fit(X_train,y_train)

         y_pred = RandomForest.predict(X_test)

         print('Accuracy:', accuracy_score(y_test,y_pred))
         print(confusion_matrix(y_test,y_pred))
         print(classification_report(y_test,y_pred))

         Accuracy: 0.975
         [[55  3]
          [ 0 62]]
                       precision    recall  f1-score   support

                    0       1.00      0.95      0.97        58
                    1       0.95      1.00      0.98        62

             accuracy                           0.97       120
            macro avg       0.98      0.97      0.97       120
         weighted avg       0.98      0.97      0.97       120

In [60]: from sklearn.ensemble import AdaBoostClassifier
         AdaBoost = AdaBoostClassifier()
```


Home Page - Select or create a n
kidney - Jupyter Notebook
https://naanmudhalvan.smartint
localhost:8888/notebooks/kidney.ipynb
jupyter kidney (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [60]: from sklearn.ensemble import AdaBoostClassifier
         AdaBoost = AdaBoostClassifier()
         AdaBoost = AdaBoost.fit(X_train,y_train)

         y_pred = AdaBoost.predict(X_test)

         print('Accuracy:', accuracy_score(y_test,y_pred))
         print(confusion_matrix(y_test,y_pred))
         print(classification_report(y_test,y_pred))

         Accuracy: 1.0
         [[58  0]
          [ 0 62]]
                       precision    recall  f1-score   support

                    0       1.00      1.00      1.00        58
                    1       1.00      1.00      1.00        62

             accuracy                           1.00       120
            macro avg       1.00      1.00      1.00       120
         weighted avg       1.00      1.00      1.00       120

In [61]: from sklearn.ensemble import GradientBoostingClassifier
         GradientBoost = GradientBoostingClassifier()
         GradientBoost = GradientBoost.fit(X_train,y_train)

         y_pred = GradientBoost.predict(X_test)

         print('Accuracy:', accuracy_score(y_test,y_pred))
         print(confusion_matrix(y_test,y_pred))
         print(classification_report(y_test,y_pred))

         Accuracy: 0.975
         [[55  3]
          [ 0 62]]
```

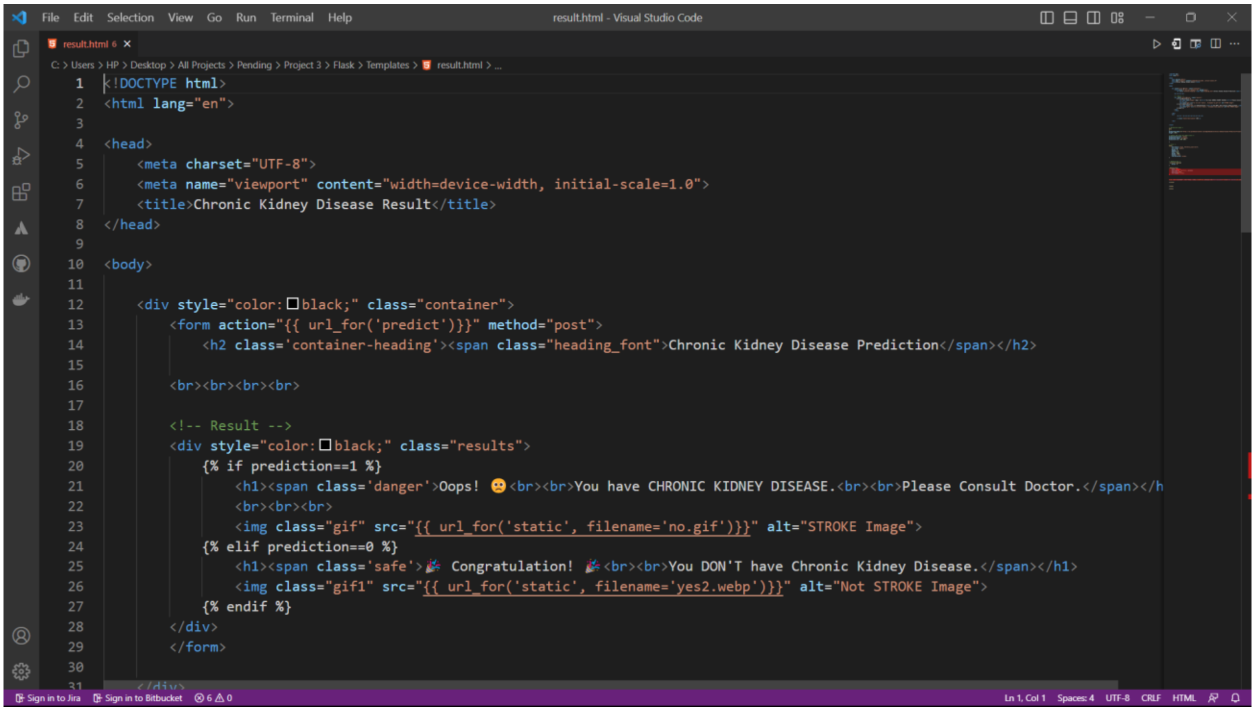

Home Page - Select or create a n
kidney - Jupyter Notebook
https://naanmudhalvan.smartint
https://naanmudhalvan.smartint
localhost:8888/notebooks/kidney.ipynb
Booking.com
Booking.com
LastPass password...
Express VPN
Gmail
YouTube
Maps
News
Translate
get_storage | Flutte...
Other favorites

```
Accuracy: 1.0
[[58  0]
 [ 0 62]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        58
           1       1.00      1.00      1.00        62

    accuracy                           1.00       120
   macro avg       1.00      1.00      1.00       120
weighted avg       1.00      1.00      1.00       120
```

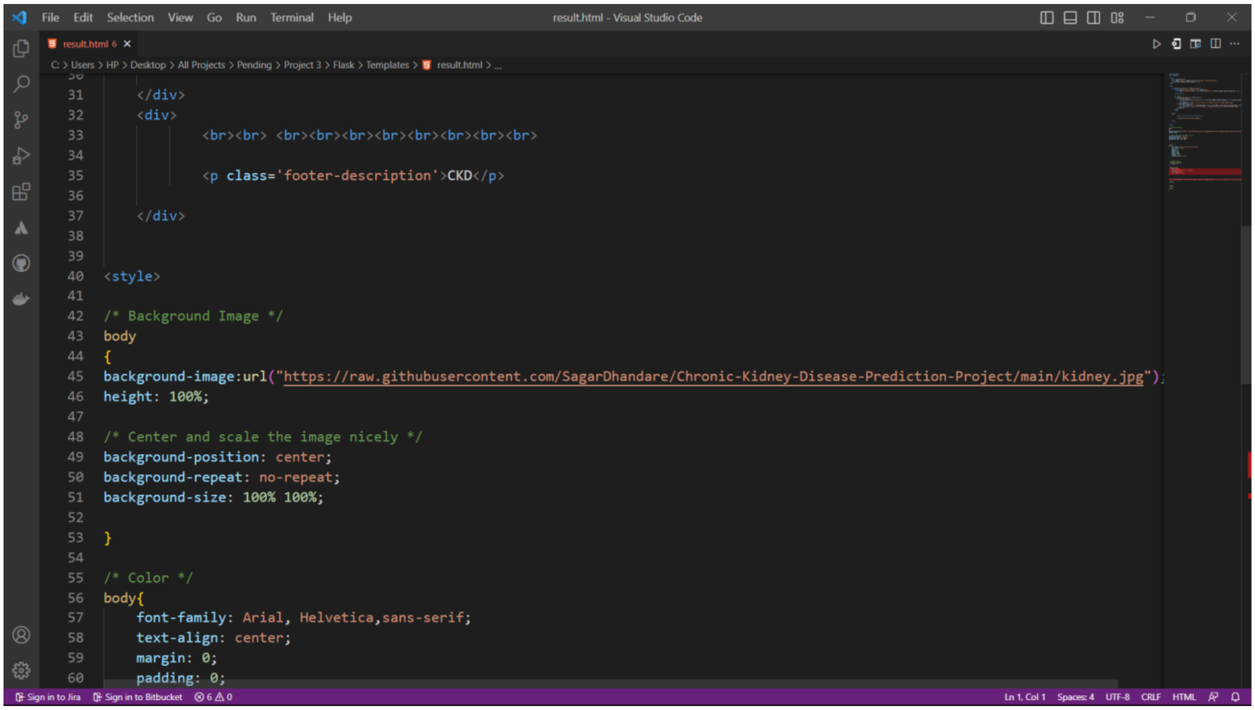
```
In [61]: from sklearn.ensemble import GradientBoostingClassifier
         GradientBoost = GradientBoostingClassifier()
         GradientBoost = GradientBoost.fit(X_train,y_train)

         y_pred = GradientBoost.predict(X_test)

         print('Accuracy:', accuracy_score(y_test,y_pred))
         print(confusion_matrix(y_test,y_pred))
         print(classification_report(y_test,y_pred))
```

```
Accuracy: 0.975
[[55  3]
 [ 0 62]]
              precision    recall  f1-score   support

           0       1.00      0.95      0.97        58
           1       0.95      1.00      0.98        62

    accuracy                           0.97       120
   macro avg       0.98      0.97      0.97       120
weighted avg       0.98      0.97      0.97       120
```

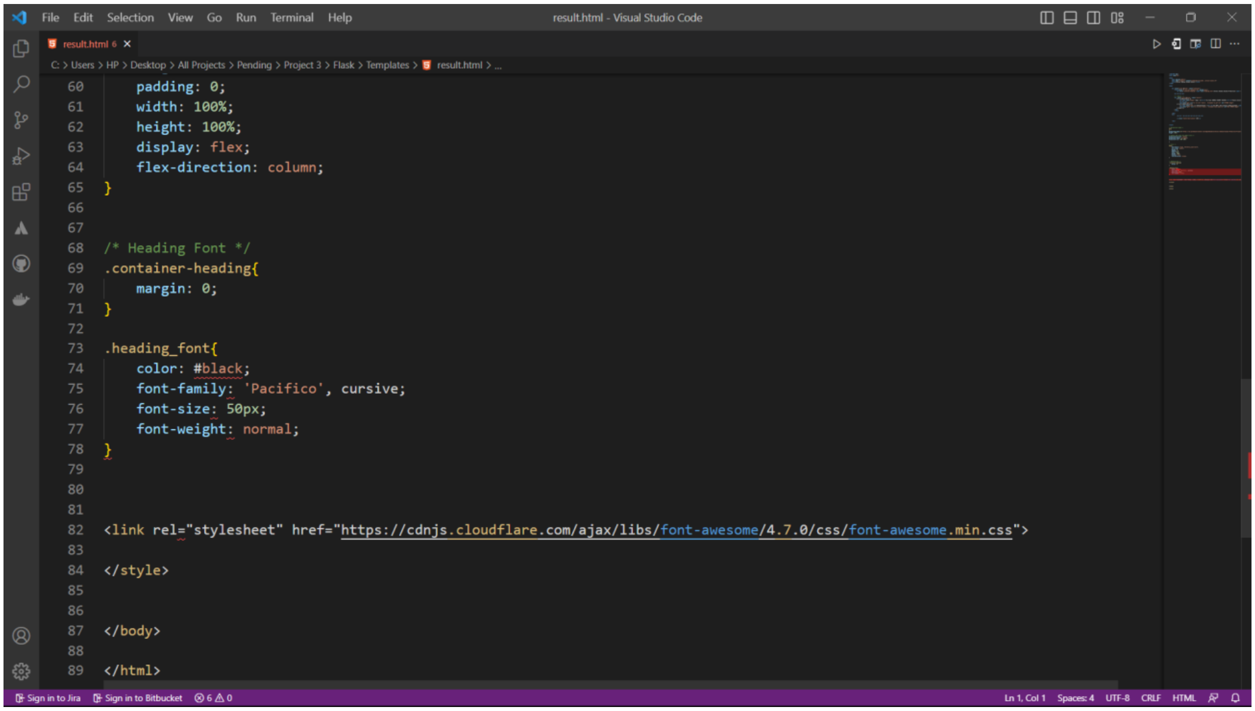
Result.html


File
Edit
Selection
View
Go
Run
Terminal
Help
result.html - Visual Studio Code
result.html 6
C: > Users > HP > Desktop > All Projects > Pending > Project 3 > Flask > Templates > result.html > ...
Sign in to Jira
Sign in to Bitbucket
6 0
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Chronic Kidney Disease Result</title>
</head>

<body>

    <div style="color:black;" class="container">
        <form action="{{ url_for('predict')}}" method="post">
            <h2 class='container-heading'><span class="heading_font">Chronic Kidney Disease Prediction</span></h2>

        <br><br><br><br>

        <!-- Result -->
        <div style="color:black;" class="results">
            {% if prediction==1 %}
                <h1><span class='danger'>Oops! 🙁<br><br>You have CHRONIC KIDNEY DISEASE.<br><br>Please Consult Doctor.</span></h
                <br><br><br>
                <img class="gif" src="{{ url_for('static', filename='no.gif')}}" alt="STROKE Image">
            {% elif prediction==0 %}
                <h1><span class='safe'>🎉 Congratulation! 🎉<br><br>You DON'T have Chronic Kidney Disease.</span></h1>
                <img class="gif1" src="{{ url_for('static', filename='yes2.webp')}}" alt="Not STROKE Image">
            {% endif %}
        </div>
        </form>

    </div>
```

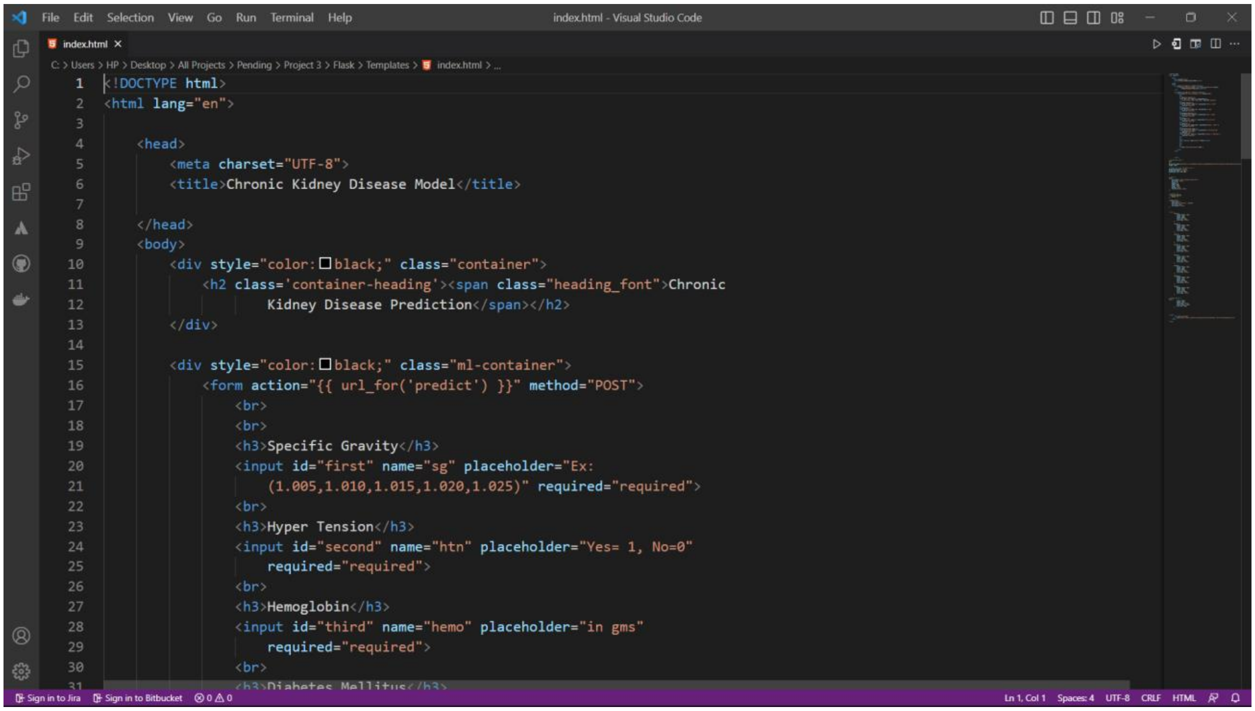

File
Edit
Selection
View
Go
Run
Terminal
Help
result.html - Visual Studio Code
result.html 6
C: > Users > HP > Desktop > All Projects > Pending > Project 3 > Flask > Templates > result.html > ...
Sign in to Jira
Sign in to Bitbucket
⊗ 6 ⚠ 0
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
HTML

```
    </div>
    <div>
            <br><br> <br><br><br><br><br><br><br><br>

            <p class='footer-description'>CKD</p>

    </div>


<style>

/* Background Image */
body
{
background-image:url("https://raw.githubusercontent.com/SagarDhandare/Chronic-Kidney-Disease-Prediction-Project/main/kidney.jpg");
height: 100%;

/* Center and scale the image nicely */
background-position: center;
background-repeat: no-repeat;
background-size: 100% 100%;

}

/* Color */
body{
    font-family: Arial, Helvetica,sans-serif;
    text-align: center;
    margin: 0;
    padding: 0;
```

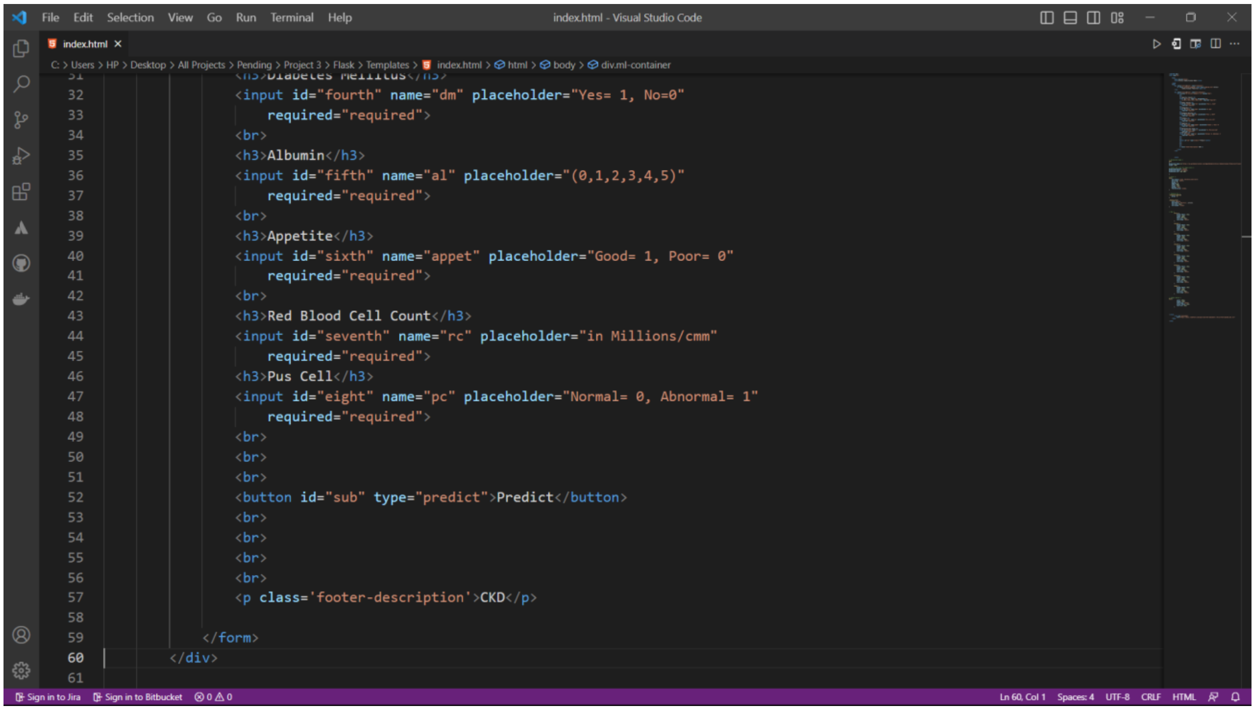

File
Edit
Selection
View
Go
Run
Terminal
Help
result.html - Visual Studio Code
result.html 6
C: > Users > HP > Desktop > All Projects > Pending > Project 3 > Flask > Templates > result.html > ...
Sign in to Jira
Sign in to Bitbucket
⊗ 6 ⚠ 0
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
HTML

```
    padding: 0;
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
}


/* Heading Font */
.container-heading{
    margin: 0;
}

.heading_font{
    color: #black;
    font-family: 'Pacifico', cursive;
    font-size: 50px;
    font-weight: normal;
}



<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

</style>


</body>

</html>
```

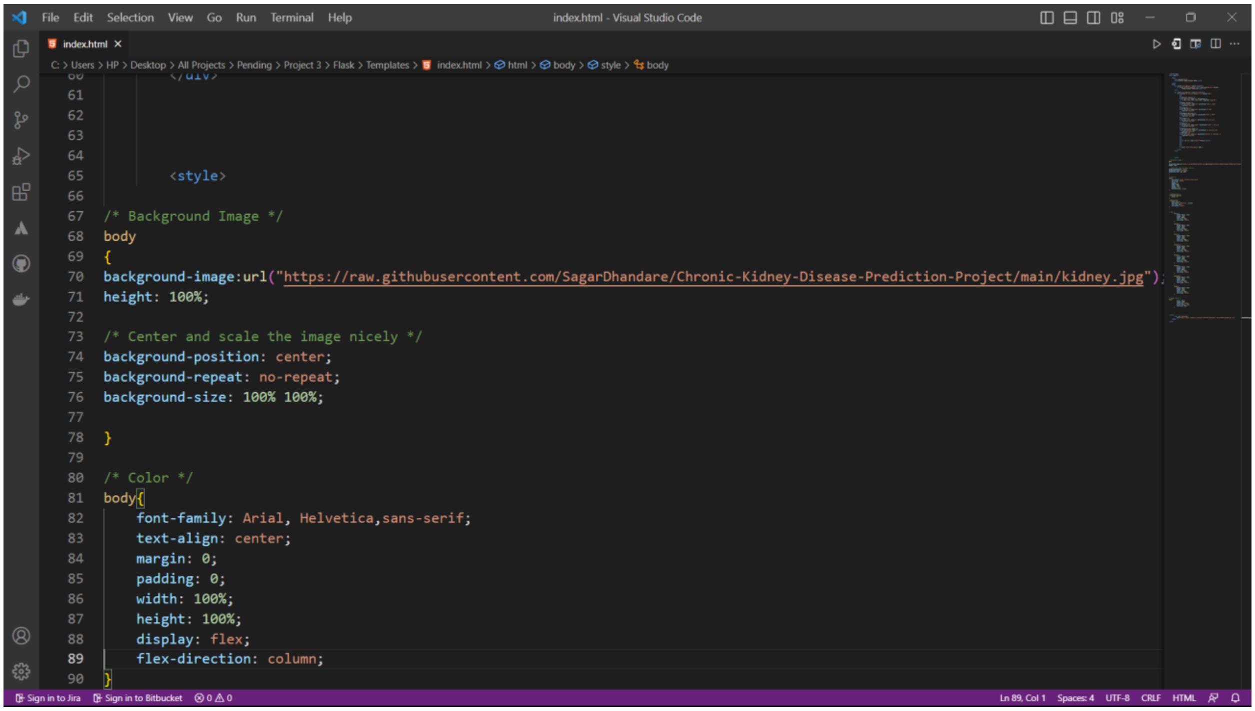
Index.html


File Edit Selection View Go Run Terminal Help
index.html - Visual Studio Code
index.html
C: > Users > HP > Desktop > All Projects > Pending > Project 3 > Flask > Templates > index.html > ...
Sign in to Jira
Sign in to Bitbucket
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML

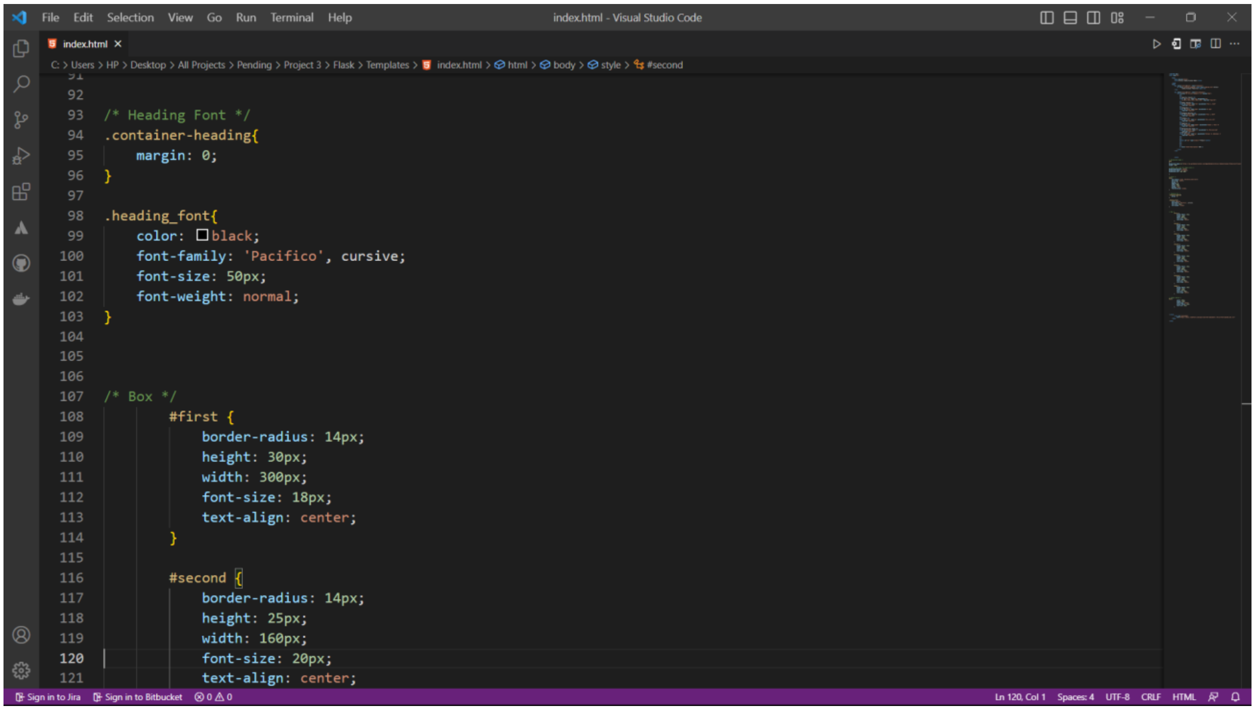
```html
<!DOCTYPE html>
<html lang="en">

    <head>
        <meta charset="UTF-8">
        <title>Chronic Kidney Disease Model</title>

    </head>
    <body>
        <div style="color:black;" class="container">
            <h2 class='container-heading'><span class="heading_font">Chronic
                    Kidney Disease Prediction</span></h2>
        </div>

        <div style="color:black;" class="ml-container">
            <form action="{{ url_for('predict') }}" method="POST">
                <br>
                <br>
                <h3>Specific Gravity</h3>
                <input id="first" name="sg" placeholder="Ex:
                    (1.005,1.010,1.015,1.020,1.025)" required="required">
                <br>
                <h3>Hyper Tension</h3>
                <input id="second" name="htn" placeholder="Yes= 1, No=0"
                    required="required">
                <br>
                <h3>Hemoglobin</h3>
                <input id="third" name="hemo" placeholder="in gms"
                    required="required">
                <br>
                <h3>Diabetes Mellitus</h3>
```


File Edit Selection View Go Run Terminal Help
index.html - Visual Studio Code
index.html
C: > Users > HP > Desktop > All Projects > Pending > Project 3 > Flask > Templates > index.html > html > body > div.ml-container
Sign in to Jira
Sign in to Bitbucket
0 0
Ln 60, Col 1 Spaces: 4 UTF-8 CRLF HTML

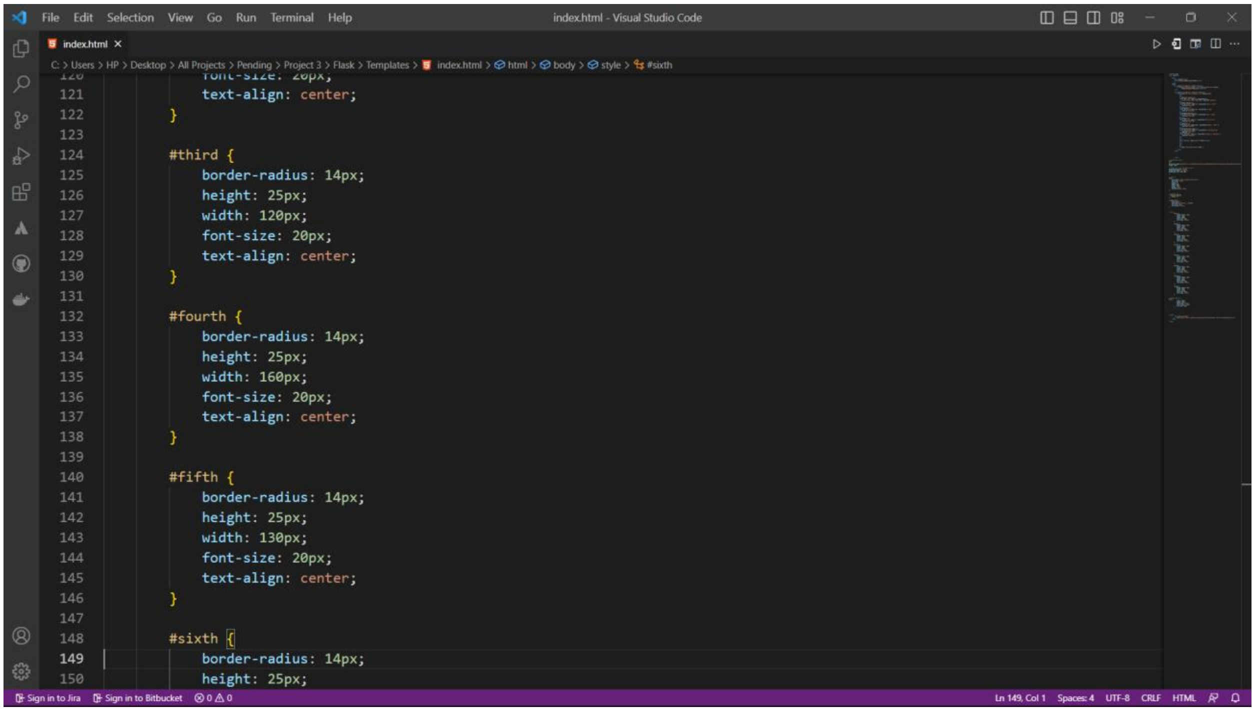
```html
                <input id="fourth" name="dm" placeholder="Yes= 1, No=0"
                    required="required">
                <br>
                <h3>Albumin</h3>
                <input id="fifth" name="al" placeholder="(0,1,2,3,4,5)"
                    required="required">
                <br>
                <h3>Appetite</h3>
                <input id="sixth" name="appet" placeholder="Good= 1, Poor= 0"
                    required="required">
                <br>
                <h3>Red Blood Cell Count</h3>
                <input id="seventh" name="rc" placeholder="in Millions/cmm"
                    required="required">
                <h3>Pus Cell</h3>
                <input id="eight" name="pc" placeholder="Normal= 0, Abnormal= 1"
                    required="required">
                <br>
                <br>
                <br>
                <button id="sub" type="predict">Predict</button>
                <br>
                <br>
                <br>
                <br>
                <p class='footer-description'>CKD</p>

            </form>
        </div>

```


File Edit Selection View Go Run Terminal Help
index.html - Visual Studio Code
index.html ×
C: > Users > HP > Desktop > All Projects > Pending > Project 3 > Flask > Templates > index.html > html > body > style > body
Sign in to Jira
Sign in to Bitbucket
⊗ 0 ⚠ 0
Ln 89, Col 1
Spaces: 4
UTF-8
CRLF
HTML

```
        </div>




        <style>

/* Background Image */
body
{
background-image:url("https://raw.githubusercontent.com/SagarDhandare/Chronic-Kidney-Disease-Prediction-Project/main/kidney.jpg")
height: 100%;

/* Center and scale the image nicely */
background-position: center;
background-repeat: no-repeat;
background-size: 100% 100%;

}

/* Color */
body{
    font-family: Arial, Helvetica,sans-serif;
    text-align: center;
    margin: 0;
    padding: 0;
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
}
```

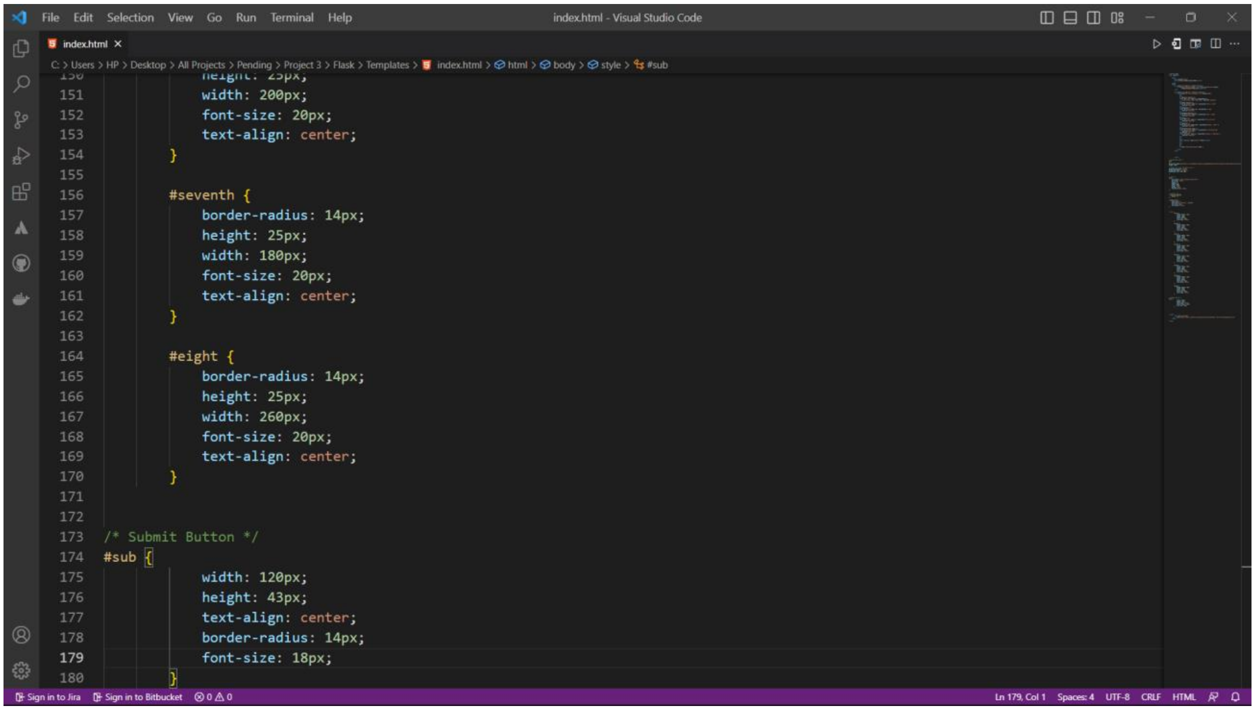

File Edit Selection View Go Run Terminal Help
index.html - Visual Studio Code
index.html ×
C: > Users > HP > Desktop > All Projects > Pending > Project 3 > Flask > Templates > index.html > html > body > style > #second
Sign in to Jira
Sign in to Bitbucket
⊗ 0 ⚠ 0
Ln 120, Col 1
Spaces: 4
UTF-8
CRLF
HTML

```

/* Heading Font */
.container-heading{
    margin: 0;
}

.heading_font{
    color: □black;
    font-family: 'Pacifico', cursive;
    font-size: 50px;
    font-weight: normal;
}



/* Box */
        #first {
            border-radius: 14px;
            height: 30px;
            width: 300px;
            font-size: 18px;
            text-align: center;
        }

        #second {
            border-radius: 14px;
            height: 25px;
            width: 160px;
            font-size: 20px;
            text-align: center;
```



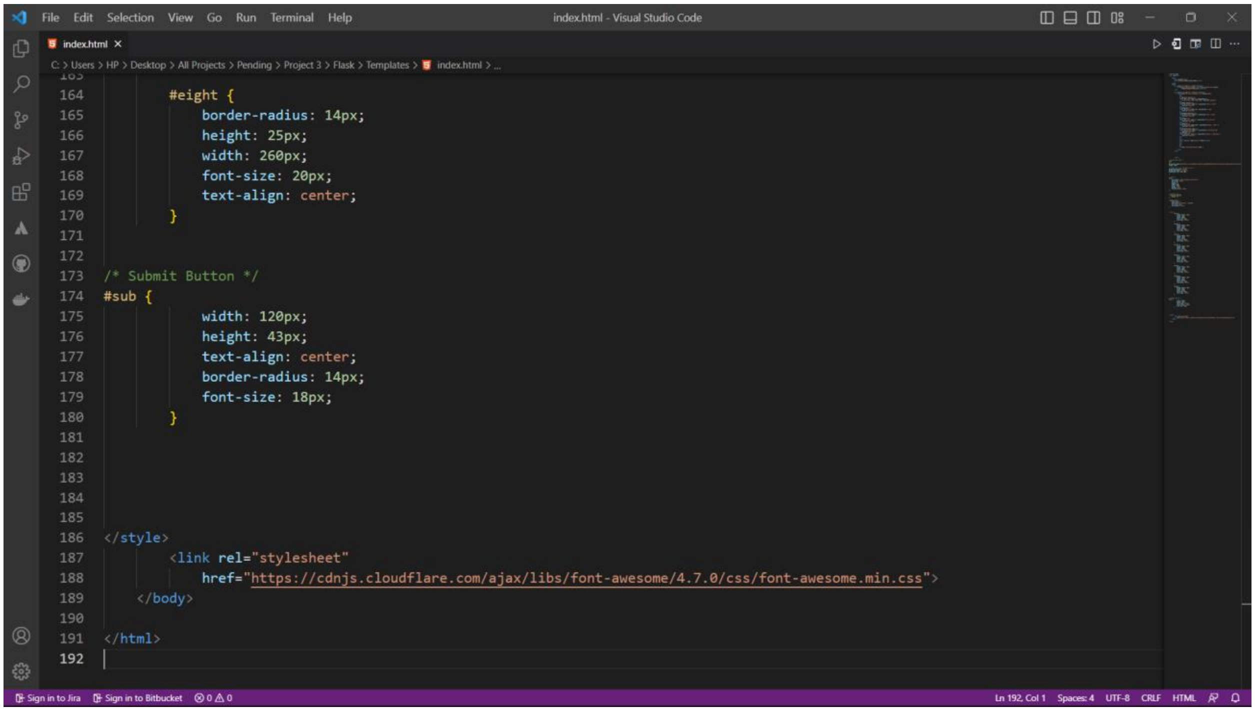
File Edit Selection View Go Run Terminal Help
index.html - Visual Studio Code
index.html
C: > Users > HP > Desktop > All Projects > Pending > Project 3 > Flask > Templates > index.html > html > body > style > #sixth
Sign in to Jira
Sign in to Bitbucket
⊗ 0 ⚠ 0
Ln 149, Col 1 Spaces: 4 UTF-8 CRLF HTML

```
            font-size: 20px;
            text-align: center;
        }

        #third {
            border-radius: 14px;
            height: 25px;
            width: 120px;
            font-size: 20px;
            text-align: center;
        }

        #fourth {
            border-radius: 14px;
            height: 25px;
            width: 160px;
            font-size: 20px;
            text-align: center;
        }

        #fifth {
            border-radius: 14px;
            height: 25px;
            width: 130px;
            font-size: 20px;
            text-align: center;
        }

        #sixth {
            border-radius: 14px;
            height: 25px;
```



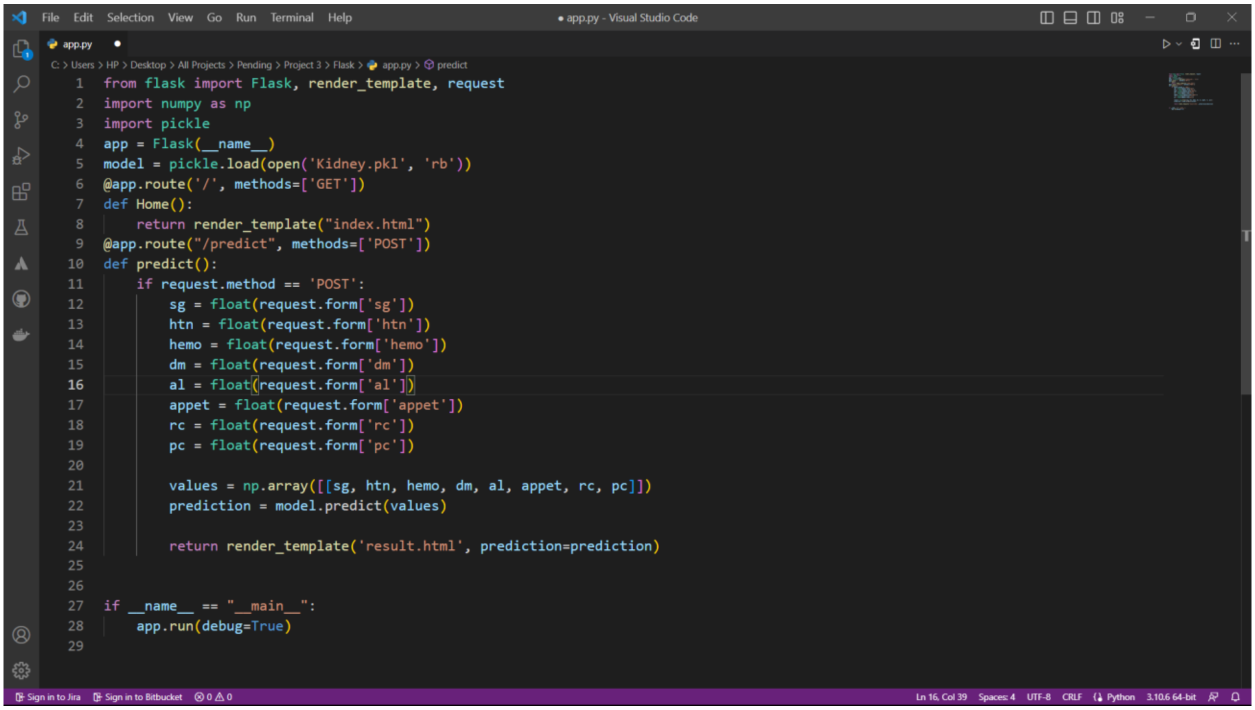
File Edit Selection View Go Run Terminal Help
index.html - Visual Studio Code
index.html
C: > Users > HP > Desktop > All Projects > Pending > Project 3 > Flask > Templates > index.html > html > body > style > #sub
Sign in to Jira
Sign in to Bitbucket
⊗ 0 ⚠ 0
Ln 179, Col 1 Spaces: 4 UTF-8 CRLF HTML

```
            height: 25px;
            width: 200px;
            font-size: 20px;
            text-align: center;
        }

        #seventh {
            border-radius: 14px;
            height: 25px;
            width: 180px;
            font-size: 20px;
            text-align: center;
        }

        #eight {
            border-radius: 14px;
            height: 25px;
            width: 260px;
            font-size: 20px;
            text-align: center;
        }


/* Submit Button */
#sub {
            width: 120px;
            height: 43px;
            text-align: center;
            border-radius: 14px;
            font-size: 18px;
        }
```


File Edit Selection View Go Run Terminal Help
index.html - Visual Studio Code
index.html
C: > Users > HP > Desktop > All Projects > Pending > Project 3 > Flask > Templates > index.html > ...
Sign in to Jira
Sign in to Bitbucket
0 0
Ln 192, Col 1 Spaces: 4 UTF-8 CRLF HTML

```
        #eight {
            border-radius: 14px;
            height: 25px;
            width: 260px;
            font-size: 20px;
            text-align: center;
        }


/* Submit Button */
#sub {
            width: 120px;
            height: 43px;
            text-align: center;
            border-radius: 14px;
            font-size: 18px;
        }





</style>
        <link rel="stylesheet"
            href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    </body>

</html>

```

App.py


File Edit Selection View Go Run Terminal Help
● app.py - Visual Studio Code
app.py
C: > Users > HP > Desktop > All Projects > Pending > Project 3 > Flask > app.py > predict
Sign in to Jira
Sign in to Bitbucket
0 0
Ln 16, Col 39 Spaces: 4 UTF-8 CRLF Python 3.10.6 64-bit

```python
from flask import Flask, render_template, request
import numpy as np
import pickle
app = Flask(__name__)
model = pickle.load(open('Kidney.pkl', 'rb'))
@app.route('/', methods=['GET'])
def Home():
    return render_template("index.html")
@app.route("/predict", methods=['POST'])
def predict():
    if request.method == 'POST':
        sg = float(request.form['sg'])
        htn = float(request.form['htn'])
        hemo = float(request.form['hemo'])
        dm = float(request.form['dm'])
        al = float(request.form['al'])
        appet = float(request.form['appet'])
        rc = float(request.form['rc'])
        pc = float(request.form['pc'])

        values = np.array([[sg, htn, hemo, dm, al, appet, rc, pc]])
        prediction = model.predict(values)

        return render_template('result.html', prediction=prediction)


if __name__ == "__main__":
    app.run(debug=True)

```