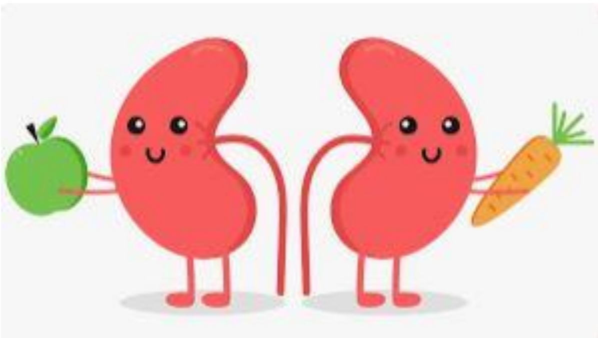
Chronic Kidney Disease Classifier

Project Report



Team Members

Nilothpal Bhattacharya A0269637L
e1113631@u.nus.edu

Contents

Change History

Author Name	**Date**	**Version**	**Change details**
Nilothpal Bhattacharya	10-May-2023	0.1	First Draft of Project report
Nilothpal Bhattacharya	23-May-2023	0.2	Draft changes
Nilothpal Bhattacharya	31-May-2023	1.0	Final draft

1. Executive Summary:

Chronic kidney disease (CKD) means the kidneys are damaged.

The following **STRAITS TIMES news article dated 27th Jan 2022** will help you understand the gravity of the problem in a country like Singapore where people lead a decent lifestyle

https://www.straitstimes.com/singapore/health/chronic-kidney-disease-on-the-rise-in-singapore-nkf-medical-director

National Kidney Foundation (NKF) said that there has been a worrying increase in patients with chronic kidney disease here in recent years.

A person is said to have this condition when they have had kidney disease for more than three months, he said. In Singapore, this is typically the result of hypertension, diabetes, or a combination of the two.

"Both of these conditions are very prevalent in Singapore," said Dr Behram, adding that based on several studies, about two out of three Singaporeans are at risk of developing chronic kidney disease in their lifetime.

Some key facts as per https://nkfs.org/about-us/key-statistics/ :

- Singapore ranks 1st in the world for diabetes-induced kidney failure
- Singapore ranks 6th in the world for prevalence (existing cases) of kidney failure
- Singapore ranks 3rd in the world for incidence (new cases) of kidney failure
- More than 300,000 people in Singapore suffer from CKD, many remain undiagnosed
- About 6 patients are diagnosed daily
- There are currently more than 9000 dialysis patients in Singapore
- About 2 in 3 new cases of kidney failure in Singapore are due to diabetes

Kidney failure is a significant economic burden. **$230 million is spent annually** on dialysis treatment.

Chronic kidney disease, also known as **chronic kidney failure**, describes the slow loss of kidney function. There are five stages of chronic kidney disease as described below.

- **Stage 1** and **2** are the mildest stages and indicate that the kidney is not working at full capacity.
- **Stage 3** occurs when the kidney function is at 50%, causing symptoms such as high blood pressure or bone problems.
- **Stage 4** is when severe kidney damage has occurred. At this point, the treatment plan is to preserve function while managing the symptoms.
- **Stage 5** is kidney failure where the kidney is unable to filter and remove waste, electrolytes and excess fluid on its own. Dialysis or a kidney transplant is the only viable life-sustaining options

Monitor your test results particularly for these two indicators:

- ACR (Albumin to Creatinine Ratio) is detected through a urine test to show how much albumin (a type of protein) is in your urine. Too much albumin is an indicator of early kidney damage.

- GFR (glomerular filtration rate) is found using a blood test which measures which stage of the 5 stages of chronic kidney disease you are at.

In Singapore, more than 50% of patients with stage 5 chronic kidney disease who were undergoing dialysis in 2015, have diabetes-related kidney damage (diabetic nephropathy) as the main underlying cause of their condition[1]. (**Source: Mount Elizabeth Medical Centre**)

2. Business Problem Background:

A hospital network wants to reduce the number of readmissions for patients with chronic kidney disease (CKD). They are always looking for solution that can help identify patients who are at a higher risk of readmission and provide personalized interventions to improve their outcomes. Hospitals want to leverage their existing patient data, including demographic information, medical history, and clinical data, **to develop a predictive model that can identify patients at-risk** and recommend appropriate interventions. The goal is to reduce readmissions, improve patient outcomes, and lower healthcare costs associated with CKD management.

By analyzing various patient data such as medical history, family history, laboratory results, and lifestyle factors:

- A predictive model can help to classify if the person is a CKD patient or not and based on further investigations can generate a risk score for each patient.
- These kind of patients need to go through certain lab tests quite often and this data can be put into some kind of AI model that can continuously learn to classify the patients based on the available data
- The predictive approach can help healthcare providers optimize their resources and provide more personalized care to each patient.
- Also, a predictive model can help identify trends and patterns in CKD progression, which can aid in the development of new treatments and interventions
- CKD can end up for a patient to get dialysis done which could be an expensive affair for a low income family and early detection could be useful for such categories.

With the advancement of technology, tools, mathematics and science, such Intelligent AI systems can be built to assist the doctors or medical practitioners to make proper judgement of a case.

Damaged kidneys are not able to keep you healthy. They cannot filter your blood well enough, and they cannot do their other jobs as well as they should. Kidney disease does not happen overnight. It happens slowly, and in stages. Most people in the early stages do not have any symptoms. They may not know that anything is wrong. But if it is found and treated, kidney disease can often be slowed or stopped. CKD patients can live for longer duration if the disease is detected at an early or intermediate stage.

If kidney disease gets worse, wastes can build to high levels in your blood and make you feel sick. You may get other problems like high blood pressure, a low red blood cell count (anaemia), weak bones, poor nutrition, and nerve damage. You will also have a higher chance of getting heart and blood vessel disease. **CKD** is a long-term condition where the kidneys are unable to function properly, resulting in a gradual loss of kidney function over time. The condition can lead to a build-up of waste products and excess fluids in the body, causing various complications such as high blood pressure, anaemia, bone disease, and nerve damage. **CKD** is often caused by other health conditions, such as diabetes and high blood pressure, and it can progress to end-stage kidney failure, which requires dialysis or a kidney transplant to manage. If it keeps getting worse, it can lead to kidney failure. This means your kidneys no longer work well enough to keep you alive, and you need a treatment like dialysis or a kidney transplant

Chronic kidney disease (CKD) interferes with the body's physiological and biological mechanisms, such as fluid electrolyte and pH balance, blood pressure regulation, excretion of toxins and waste, vitamin D metabolism, and hormonal regulation. Many CKD patients are at risk of hyperkalaemia, hyperphosphatemia, chronic metabolic acidosis, bone deterioration, blood pressure abnormalities, and edema. These risks may be minimized, and the disease's progression may be slowed through careful monitoring of protein, phosphorus, potassium, sodium, and calcium, relieving symptoms experienced by CKD patients.

3. Project Objective:

The project has two main aims:

- Based on the publicly available data, it aims to gain insight and understanding of the raw data. To understand the variation and correlation between the parameters been recorded for multiple patients
- The second aim is to classify the patients whether the patient is a CKD or non-CKD patient based on supervised learning technique. Medical practitioners can use this to feed individual data points for a patient and can check the CKD status of patients.

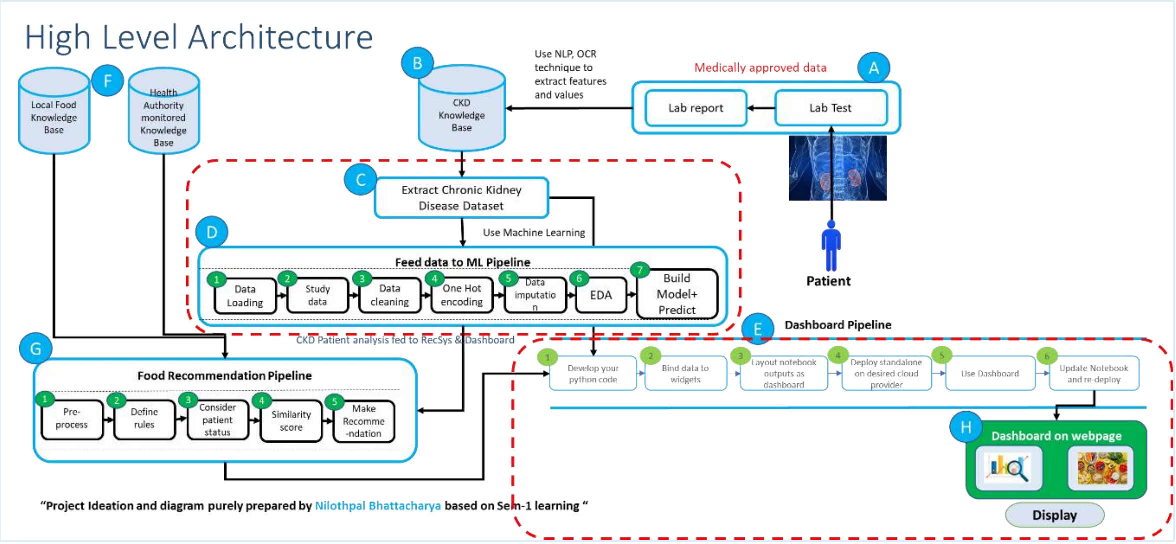
4. Scope of project:

In Scope activities: The project aims to cover the initial data imputation, initial exploratory data analysis and apply machine learning techniques to identify and classify CKD patients. Please refer to item C, D, E, H in the high-level architecture diagram to understand the scope better.

Out of Scope: The feedback mechanism to re-train the model is currently kept out of scope since the the main focus is to gain as much insight possible based on the available data. Definitely the feedback mechanism can help to fine tune the models and improve the accuracy of classification further. But in the interest of time and effort, the scope of analysis is limited and will be taken as an improvement strategy in the next stage. Also the project doesn't aim to calculate a risk score for any patient, such a kind of exercise would require more data, knowledge and interaction with medical practitioners and researchers who are dealing with CKD patients.

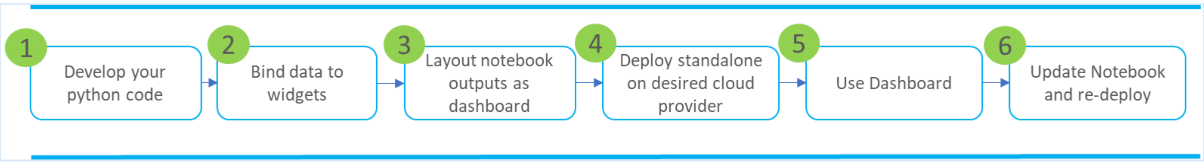
5. Architecture

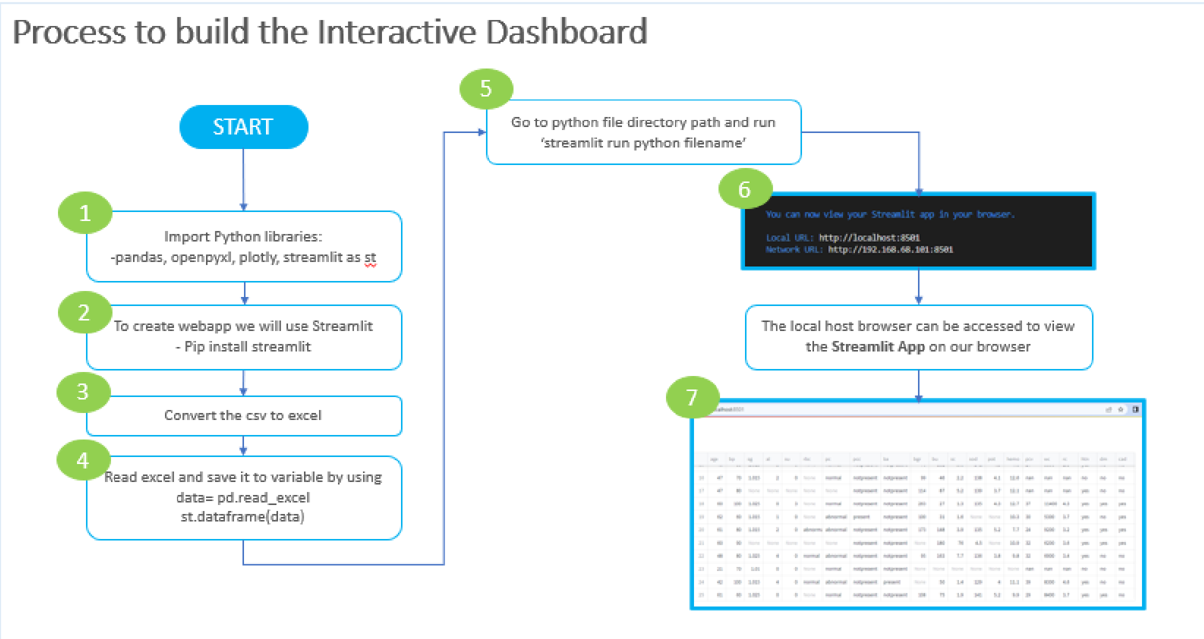
Client-server architecture been used for the web development part here. In this architecture there could multiple clients and server connected over the internet which is nowadays the usual case when any user is trying to access any website/webapp. The client is the browser that user access and the client and server interact using some protocol (the rules that both sides agree to communicate with each other) generated locally through a localhost. On Client-side, HTML scripting is usually followed. Website or webapp should be secured to avoid malicious access to the webapp/website. We are going to use Python for server-side scripting. There are two major frameworks used for server-side scripting in Python these days– one is Django and the other one is Flask. We will use Flask framework in this project. Python will also be used for the analysis of data collected for the project.



Technical

Dashboard App development cycle





6. Dataset details:

The dataset of CKD has been taken from the publicly available and accessible dataset Irvine ML Repository https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease ,which included 400 instances. The dataset captures multiple features that can be used to assess the condition of a patient or subject whether the have CKD or No CKD. Such datasets are not made publicly available due to the nature of it's sensitivity and as per regulations of Healthcare and Lifesciences worldwide. This dataset is assumed to be an unreal data but is trying to mimick similar data-points as observed by the medical practitioners.

Data Set Characteristics:	**Multivariate**	Number of Instances:	**400**	**Name**	**kidney_disease.csv**
Attribute Characteristics:	Real	**Number of Attributes:**	25	**Date Donated**	03-07-2015
Associated Tasks:	Classification	**Missing Values?**	Yes	**Number of Web Hits:**	269640
AI Reasoning Category:	Analytics Task	**Numeric features**	11	**Nominal features**	14

7. Dataset Schema:

Attribute Name	Attribute Type	Attribute Values	Attribute Code
Age	numeric	years	age - age
Blood Pressure	numeric	mm/Hg	bp - blood pressure
Spesific Gravity	numeric	1.005, 1.010, 1.015, 1.020, 1.025	sg - specific gravity
Albumin	numeric	0, 1, 2, 3, 4, 5	al - albumin
Sugar	numeric	0, 1, 2, 3, 4, 5	su - sugar
Red Blood Cells	nominal	normal, abnormal	rbc - red blood cells
Plus Cell	nominal	normal, abnormal	pc - pus cell
Plus Cell Clumps	nominal	present, notpresent	pcc - pus cell clumps
Bacteria	nominal	present, notpresent	ba - bacteria
Blood Glucosa Random	numeric	mgs/dl	bgr - blood glucose random
Blood Urea	numeric	mgs/dl	bu - blood urea
Serum Creatine	numeric	mgs/dl	sc - serum creatinine
Sodium	numeric	mEq/l	sod - sodium
Potassium	numeric	mEq/l	pot - potassium
Hemoglobin	numeric	gms	hemo - hemoglobin
Packed Cell Volume	numeric	pcv	pcv - packed cell volume

White Blood Cell Count	numeric	cells/cumm	wc - white blood cell count
Red Blood Cell Count	numeric	millions/cumm	rc - red blood cell count
Hypertension	numeric	yes, no	htn - hypertension
Diabetes Mellitus	numeric	yes, no	dm - diabetes mellitus
Coronary Artery Disease	nominal	yes, no	cad - coronary artery disease
Appetite	nominal	good, poor	appet - appetite
Pedal Edema	nominal	yes, no	pe - pedal edema
Anemia	nominal	yes, no	ane - anemia
Class	nominal	ckd, notckd	class - class

Description of Fields:

Each row represents an observation or instance, while each column represents a feature or attribute. Here is a brief description of each column:

Field/Attribute/Feature	Field/Attribute/Feature Description
id	**id:** A unique identifier for each observation.
gravity	**gravity:** Specific gravity of urine, which is a measure of the density of urine compared to the density of water.
ph	**ph:** The acidity or basicity of urine, measured on a scale of 0 to 14, where 7 is neutral.
osmo	**osmo:** Osmolality of urine, which is a measure of the concentration of particles in urine.
cond	**cond:** Conductivity of urine, which is a measure of how well urine conducts electricity.
urea	**urea:** Concentration of urea in urine, which is a waste product produced by the liver during protein metabolism.
calc	**calc:** Concentration of calcium in urine, which is an essential mineral that plays a role in many bodily functions.
target:	**target:** The target variable or label, which indicates whether the observation belongs to a certain class or category. In this case, it is a binary variable indicating whether the observation represents a healthy (0) or unhealthy (1) patient.
	The context of data is related to medical diagnosis or analysis of urine samples

8. Description of Fields:

The data provided is a table with several columns and rows. Each row represents an observation or instance, while each column represents a feature or attribute. Here is a brief description of each column:

Field/Attribute/Feature	Field/Attribute/Feature Description
id	A unique identifier for each observation.
age	Age of the patient in years.
bp	Blood pressure of the patient in mm/Hg. **High blood pressure** (hypertension) is a common cause of chronic kidney disease, so it is important to measure blood pressure in patients with kidney problems.
sg	Specific gravity of the patient's urine. Specific gravity is a measure of the concentration of particles in the urine, and can be used to detect kidney problems such as dehydration or impaired kidney function.
al	Amount of **albumin** in the patient's urine. **Albumin** is a protein that is normally present in the blood, but should not be present in significant amounts in the urine. The presence of albumin in the urine (proteinuria) can be a sign of kidney damage.
su**:**	Amount of sugar in the patient's urine. The presence of sugar in the urine (glycosuria) can be a sign of **diabetes mellitus**, which is a common cause of chronic kidney disease.
rbc**:**	Presence or absence of red blood cells in the patient's urine. The presence of red blood cells in the urine (haematuria) can be a sign of kidney damage or other problems in the urinary tract.
pc**:**	Presence or absence of pus cells in the patient's urine. The presence of pus cells in the urine (pyuria) can be a sign of infection or inflammation in the urinary tract.
pcc**:**	Presence or absence of urinary casts in the patient's urine. Urinary casts are cylindrical structures that can form in the kidneys or other parts of the urinary tract. The presence of casts in the urine can be a sign of kidney disease.
ba**:**	Presence or absence of bacteria in the patient's urine. The presence of bacteria in the urine (bacteriuria) can be a sign of infection in the urinary tract
bgr**:**	**Blood glucose** random measurement in mg/dL. Elevated blood glucose levels are a hallmark of diabetes mellitus, which is a common cause of chronic kidney disease.
bu**:**	**Blood urea** measurement in mg/dL. **Urea** is a waste product that is normally removed from the body by the kidneys. Elevated blood urea levels can be a sign of impaired kidney function
sc**:**	Serum creatinine measurement in mg/dL. **Creatinine** is a waste product that is normally removed from the body by the kidneys. Elevated serum creatinine levels can be a sign of impaired kidney function.

sod:	**Sodium (Na)** concentration in mEq/L. Sodium is an electrolyte that is normally regulated by the kidneys. Abnormal sodium levels can be a sign of kidney dysfunction.
pot:	**Potassium** (K) concentration in mEq/L. Potassium is an electrolyte that is normally regulated by the kidneys. Abnormal potassium levels can be a sign of kidney dysfunction.
hemo:	Haemoglobin measurement in g/dL. Haemoglobin is a protein in red blood cells that carries oxygen to the body's tissues. Anaemia (low haemoglobin levels) can be a complication of chronic kidney disease.
pcv:	Packed cell volume (PCV) measurement as a percentage. Packed cell volume (PCV) measurement as a percentage. PCV is a measure of the proportion of blood that is made up of red blood cells. Anaemia (low PCV levels) can be a complication of chronic kidney disease
wc:	White blood cell (WBC) count in cells/cumm. White blood cell (WBC) count in cells/cumm. White blood cells are a component of the immune system that help to fight infection. Elevated WBC counts can be a sign of infection or inflammation in the body.
rc:	Red blood cell (RBC) count in millions/cumm. This is a measure of the number of red blood cells in the patient's blood, expressed in millions per cubic millimetre (cumm)
htn:	Presence or absence of hypertension in the patient. Hypertension is a medical condition characterized by high blood pressure, which can increase the risk of heart disease, stroke, and other health problems.
dm:	Presence or absence of diabetes mellitus in the patient. Presence of diabetes mellitus, indicated as "yes" or "no". Diabetes mellitus is a chronic disease in which the body is unable to properly use and store glucose (a type of sugar), leading to high levels of sugar in the blood
cad:	Presence or absence of coronary artery disease in the patient. Presence of coronary artery disease, indicated as "yes" or "no". Coronary artery disease is a condition in which the arteries that supply blood to the heart become narrowed or blocked, increasing the risk of heart attack and other heart problems
appet:	Patient's appetite, indicated as "good" or "poor". This is a subjective measure of the patient's appetite, which can be affected by a range of factors such as nausea, vomiting, and other symptoms.
pe:	Presence of pedal edema, indicated as "yes" or "no". Pedal edema is swelling in the feet and ankles that can be caused by a range of medical conditions
ane:	Presence or absence of anaemia in the patient. of anaemia, indicated as "yes" or "no". Anaemia is a condition in which the body does not have enough red blood cells or haemoglobin to carry oxygen to the body's tissues, leading to fatigue, weakness, and other symptoms
classification:	The target variable indicating the presence of chronic kidney disease (CKD), indicated as "**ckd**" or "**notckd**". Chronic kidney disease is a condition in which the kidneys are damaged and are unable to properly filter waste products from the blood, leading to a build-up of toxins and other substances in the body. CKD can lead to a range of health problems, including high blood pressure, anaemia, and bone disease, among others

9. Data pre-processing pipeline development:

The data that is used for data processing work has been clearly stated in section 10 of this project report. The initial glance at the data of 400 points showed some gaps in the data before it can be put for further analysis. The dataset was found to be comprised of two types of data mainly : numerical and categorical data. The data processing was done with the help of programming language Python where certain software packages like Numpy and Pandas make the task of data pre-processing much easier. The shape of the data reflected it contained 400 data points across 25 features. Before starting with this data pre-processing step, it was also necessary to refer to some medical references as identified in the Bibliography and Appendices. Without the understanding of some medical terms it is not fair to do such kind of analysis and may result the risk of running incorrect interpretation of data in the given dataset.

On applying some initial checks it was observed the data was lagging in following terms:

- Well a rough initial prompt for me was to just get an initial impression of the data by running few basic commands from pandas package and that helped me to understand what I have in front of me

```
data.head(10)
```

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	NaN	no	no	no	good	no	no	ckd
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	NaN	no	yes	no	poor	no	yes	ckd
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	4.6	no	no	no	good	no	no	ckd
5	5	60.0	90.0	1.015	3.0	0.0	NaN	NaN	notpresent	notpresent	...	39	7800	4.4	yes	yes	no	good	yes	no	ckd
6	6	68.0	70.0	1.010	0.0	0.0	NaN	normal	notpresent	notpresent	...	36	NaN	NaN	no	no	no	good	no	no	ckd
7	7	24.0	NaN	1.015	2.0	4.0	normal	abnormal	notpresent	notpresent	...	44	6900	5	no	yes	no	good	yes	no	ckd
8	8	52.0	100.0	1.015	3.0	0.0	normal	abnormal	present	notpresent	...	33	9600	4.0	yes	yes	no	good	no	yes	ckd
9	9	53.0	90.0	1.020	2.0	0.0	abnormal	abnormal	present	notpresent	...	29	12100	3.7	yes	yes	no	poor	no	yes	ckd

10 rows × 26 columns

- The first concern as a reader it was difficult to understand the abbreviated column name so the column names were elaborated after checking for them in the provided description so I felt the need to update the feature names for all of them for better readability, understanding and interpretation. On getting this kind of display will now prompt you to refer to the medical terms to seek understanding between them.

```
feature_names=['id','Age (yrs)','Blood Pressure (mm/Hg)','Specific Gravity','Albumin (g/dL)','Sugar','Red Blood Cells',
               'Pus Cells','Pus Cell Clumps','Bacteria','Blood Glucose Random (mgs/dL)','Blood Urea (mgs/dL)',
               'Serum Creatinine (mgs/dL)','Sodium (mEq/L)','Potassium (mEq/L)','Hemoglobin (gms)','Packed Cell Volume',
               'White Blood Cells (cells/cmm)','Red Blood Cells (millions/cmm)','Hypertension','Diabetes Mellitus',
               'Coronary Artery Disease','Appetite','Pedal Edema','Anemia','Chronic Kidney Disease Label']
data.columns=feature_names
```

```
data.head(5)
```

	id	Age (yrs)	Blood Pressure (mm/Hg)	Specific Gravity	Albumin (g/dL)	Sugar	Red Blood Cells	Pus Cells	Pus Cell Clumps	Bacteria	...	Packed Cell Volume	White Blood Cells (cells/cmm)	Red Blood Cells (millions/cmm)	Hypertension	Diabetes Mellitus	Coronary Artery Disease	Appetite	Pedal Edema	Anemia	Chroni Kidne Diseas Labe
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	5.2	yes	yes	no	good	no	no	ck
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	NaN	no	no	no	good	no	no	ck
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	NaN	no	yes	no	poor	no	yes	ck
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	3.9	yes	no	no	poor	yes	yes	ck
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	4.6	no	no	no	good	no	no	ck

5 rows × 26 columns

Just an overview to showcase how some of these parameters are linked to each other and their relevance to CKD:

Age (yrs): Age is a significant factor in the development and progression of CKD. As individuals age, the risk of kidney damage and dysfunction increases.

Blood Pressure (mm/Hg): High blood pressure, also known as hypertension, is a common cause and consequence of CKD. Elevated blood pressure can damage the blood vessels in the kidneys and impair their function.

Specific Gravity: Specific gravity is a measure of urine concentration. Abnormal values may indicate impaired kidney function or dehydration, which can be associated with CKD.

Albumin (g/dL): Albumin is a protein found in the blood. Elevated levels of albumin in urine (albuminuria) indicate kidney damage, as the kidneys normally filter out albumin from the urine.

Sugar: Presence of sugar in the urine may suggest uncontrolled diabetes, which is a common cause of CKD.

Red Blood Cells: The presence of red blood cells in the urine (hematuria) may indicate kidney damage or other conditions affecting the urinary system.

Pus Cells: Pus cells in the urine may indicate infection or inflammation in the urinary tract, which can affect kidney health.

Pus Cell Clumps: Clumps of pus cells in the urine may be indicative of a more severe infection or inflammation in the urinary tract.

Bacteria: The presence of bacteria in the urine suggests a urinary tract infection, which can potentially affect the kidneys if left untreated.

Blood Glucose Random (mgs/dL): Random blood glucose levels help assess blood sugar control and detect diabetes, which is a leading cause of CKD.

Blood Urea (mgs/dL) and Serum Creatinine (mgs/dL): Blood urea and serum creatinine are markers of kidney function. Elevated levels indicate impaired kidney function and are used to diagnose and monitor CKD.

Sodium (mEq/L) and Potassium (mEq/L): Electrolyte imbalances, such as abnormal sodium and potassium levels, can occur in CKD due to impaired kidney regulation. These imbalances can have various health implications.

Hemoglobin (gms) and Packed Cell Volume: Hemoglobin and packed cell volume (hematocrit) measurements are indicators of anemia, a common complication of CKD.

White Blood Cells (cells/cmm) and Red Blood Cells (millions/cmm): White blood cell and red blood cell counts may be influenced by infections, inflammation, or other conditions associated with CKD.

Hypertension, Diabetes Mellitus, Coronary Artery Disease: These are comorbid conditions that frequently coexist with CKD and can contribute to its development and progression.

Appetite, Pedal Edema, Anemia: These symptoms can be associated with CKD and indicate potential kidney dysfunction and related complications.

Chronic Kidney Disease Label: This parameter represents the presence or absence of CKD, serving as the outcome or label variable for classification purposes.

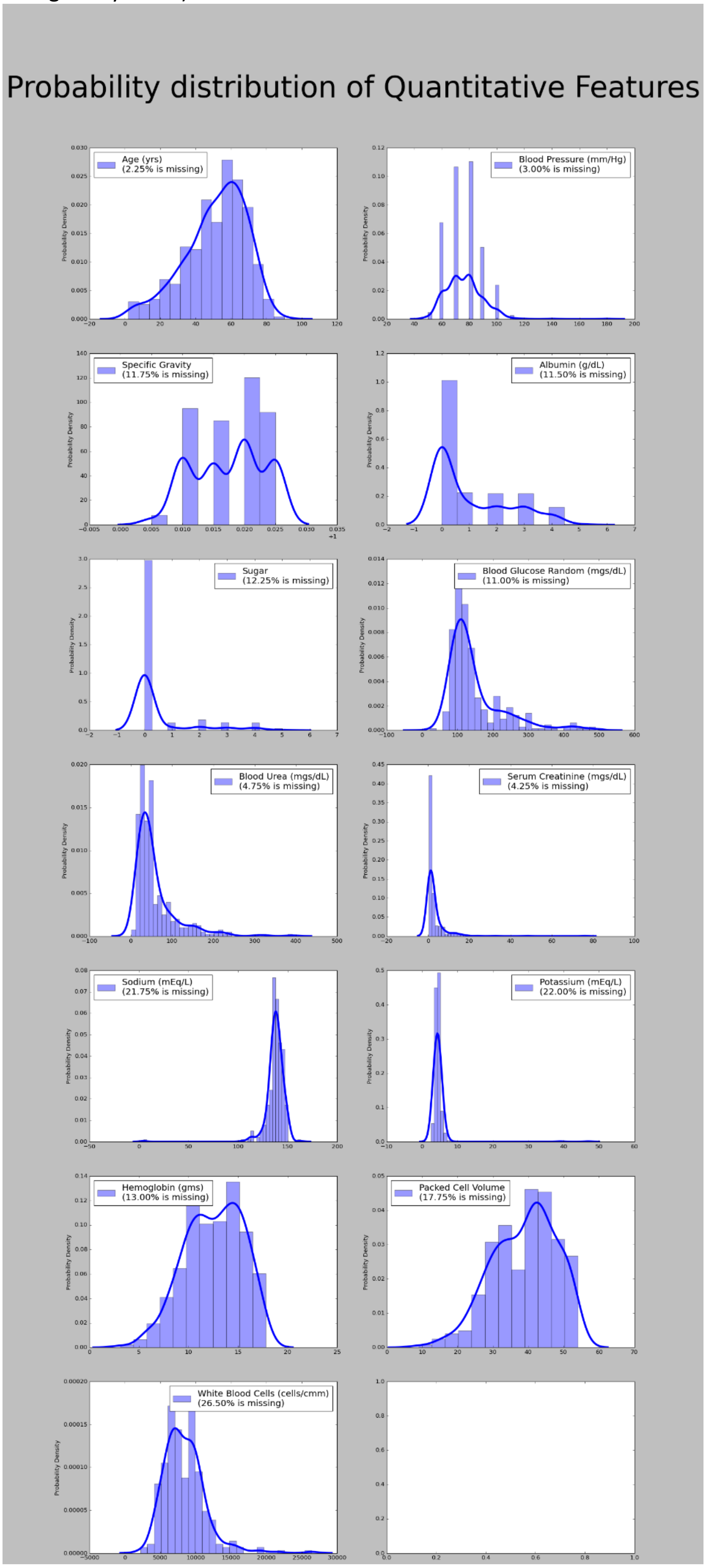
It's important to note that the relationships between these parameters are complex, and the presence of certain markers does not necessarily indicate the presence or severity of CKD. A comprehensive evaluation by healthcare professionals, including medical history, diagnostic tests, and clinical assessment, is crucial for accurate diagnosis and management of CKD.

- The second concern observed was that not all features contained exact count of 400 data points. At many places the data were found to be missing. So this posed a challenge in terms of how to cover this kind of gap in the dataset before we can think of deciding the split for test data and train data. The approach I took to look into this aspect of data was as follows:
    - There exist some industry practices to approach this kind of scenario and I will come on that shortly. But logically as a person from domains that look alike such as AI, Analytics, Machine learning specialist or as a data scientist, the first convincing approach is to look in the amount of data that is missing and to generate the data distribution pattern.
    - As it can be seen here that a lot of data is missing (**400 – Non_null count**) for most of the columns/feature/variable in the given dataset and after reading about the disease on different available sources of information I realized that the data for some of very important parameters were missing such as for Albumin and Serum Creatinine

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 26 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   id                             400 non-null    int64  
 1   Age (yrs)                      391 non-null    float64
 2   Blood Pressure (mm/Hg)         388 non-null    float64
 3   Specific Gravity               353 non-null    float64
 4   Albumin (g/dL)                 354 non-null    float64
 5   Sugar                          351 non-null    float64
 6   Red Blood Cells                248 non-null    object 
 7   Pus Cells                      335 non-null    object 
 8   Pus Cell Clumps                396 non-null    object 
 9   Bacteria                       396 non-null    object 
 10  Blood Glucose Random (mgs/dL)  356 non-null    float64
 11  Blood Urea (mgs/dL)            381 non-null    float64
 12  Serum Creatinine (mgs/dL)      383 non-null    float64
 13  Sodium (mEq/L)                 313 non-null    float64
 14  Potassium (mEq/L)              312 non-null    float64
 15  Hemoglobin (gms)               348 non-null    float64
 16  Packed Cell Volume             330 non-null    object 
 17  White Blood Cells (cells/cmm)  295 non-null    object 
 18  Red Blood Cells (millions/cmm) 270 non-null    object 
 19  Hypertension                   398 non-null    object 
 20  Diabetes Mellitus              398 non-null    object 
 21  Coronary Artery Disease        398 non-null    object 
 22  Appetite                       399 non-null    object 
 23  Pedal Edema                    399 non-null    object 
 24  Anemia                         399 non-null    object 
 25  Chronic Kidney Disease Label   400 non-null    object 
dtypes: float64(11), int64(1), object(14)
memory usage: 81.4+ KB
```

- I executed few commands that help to show the shape of data in terms of their counts for each feature or variable, the amount of data missing (usually observed in terms of the total data points in dataset versus the total count of observed values for a particular column/variable/feature. Sharing a glance of it here :
- The initial pattern of this data for each of the variables can be seen below in the form of probability distribution generated


Probability distribution of Quantitative Features
Age (yrs) (2.25% is missing)
Blood Pressure (mm/Hg) (3.00% is missing)
Specific Gravity (11.75% is missing)
Albumin (g/dL) (11.50% is missing)
Sugar (12.25% is missing)
Blood Glucose Random (mgs/dL) (11.00% is missing)
Blood Urea (mgs/dL) (4.75% is missing)
Serum Creatinine (mgs/dL) (4.25% is missing)
Sodium (mEq/L) (21.75% is missing)
Potassium (mEq/L) (22.00% is missing)
Hemoglobin (gms) (13.00% is missing)
Packed Cell Volume (17.75% is missing)
White Blood Cells (cells/cmm) (26.50% is missing)
Probability Density

- Also to notice that some features may be irrelevant to be looked more into such as the ID which doesn't reflect any scientific or mathematical nature of data so it can be dropped from the given dataset

```
data
```

	Age (yrs)	Blood Pressure (mm/Hg)	Specific Gravity	Albumin (g/dL)	Sugar	Red Blood Cells	Pus Cells	Pus Cell Clumps	Bacteria	Blood Glucose Random (mgs/dL)	...	Packed Cell Volume	White Blood Cells (cells/cmm)	Red Blood Cells (millions/cmm)	Hypertension	Diabetes Mellitus	Coronary Artery Disease	Appetite	Pedal Edema	Anemia	Chronic Kidney Disease Label
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	...	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	...	38	6000	NaN	no	no	no	good	no	no	ckd
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	...	31	7500	NaN	no	yes	no	poor	no	yes	ckd
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	...	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	...	35	7300	4.6	no	no	no	good	no	no	ckd
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	140.0	...	47	6700	4.9	no	no	no	good	no	no	notckd
396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	75.0	...	54	7800	6.2	no	no	no	good	no	no	notckd
397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	100.0	...	49	6600	5.4	no	no	no	good	no	no	notckd
398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	114.0	...	51	7200	5.9	no	no	no	good	no	no	notckd
399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	131.0	...	53	6800	6.1	no	no	no	good	no	no	notckd

400 rows × 25 columns

Checking for the missing values in data:

```
[71] # Check for missing values in the original data
missing_rows_original = data[data.isnull().any(axis=1)]

# Print the number of missing rows
print("Missing rows in original data:", len(missing_rows_original))
```

```
Missing rows in original data: 242
```

Applying One Hot encoding to convert the categorical data to numerical data

Also when I tried imputation after applying encoding , still there were errors in dataset such as `ValueError: could not convert string to float: '\t?' so this data had to be cleaned.`

- Since the model doesn't understand categorical data for processing so we need to convert them into numerical form and this process is called **One Hot Encoding** and this applies to all the features including the supervised Label which is '**`Chronic Kidney Disease Label`**' in our case. Here the supervised label represents different disease classes, such as "CKD" (Chronic Kidney Disease) and "Non-CKD," we would typically **apply one-hot encoding** to convert this categorical variable into binary columns like "CKD" and "Non-CKD" with values of 0 or 1 **One Hot encoding has to be done before applying any form of data imputation on the dataset.**

Applying One Hot encoding to convert the categorical data to numerical data

```
# Define the list of columns to be one-hot encoded including the supervised classification label if it is also categorical.
# One hot encoding is needed for any variable that is categorical in nature. In our case the supervised label 'Chronic Kidney Disease Label' is categorical so need to convert to one hot code.
categorical_columns = ['Red Blood Cells',
                'Pus Cells','Pus Cell Clumps','Bacteria','Hypertension','Diabetes Mellitus',
                'Coronary Artery Disease','Appetite','Pedal Edema','Anemia','Chronic Kidney Disease Label']
# categorical_columns = ["rbc", "pc", "pcc", "ba", "htn", "dm", "cad", "appet", "pe", "ane", "classification"]

# Apply one-hot encoding to the categorical columns
data_encoded = pd.get_dummies(data, columns=categorical_columns)

# Display the encoded dataset
print(data_encoded)
```

```
     Age (yrs)  Blood Pressure (mm/Hg)  Specific Gravity  Albumin (g/dL)  \
0         48.0                    80.0             1.020             1.0
1          7.0                    50.0             1.020             4.0
2         62.0                    80.0             1.010             2.0
3         48.0                    70.0             1.005             4.0
4         51.0                    80.0             1.010             2.0
..         ...                     ...               ...             ...
395       55.0                    80.0             1.020             0.0
396       42.0                    70.0             1.025             0.0
397       12.0                    80.0             1.020             0.0
398       17.0                    60.0             1.025             0.0
399       58.0                    80.0             1.025             0.0

     Sugar  Blood Glucose Random (mgs/dL)  Blood Urea (mgs/dL)  \
0      0.0                          121.0                 36.0
1      0.0                            NaN                 18.0
2      3.0                          423.0                 53.0
```

✓ 0s completed at 4:47 AM

```
     Sugar  Blood Glucose Random (mgs/dL)  Blood Urea (mgs/dL)  \
0      0.0                          121.0                 36.0   
1      0.0                            NaN                 18.0   
2      3.0                          423.0                 53.0   
3      0.0                          117.0                 56.0   
4      0.0                          106.0                 26.0   
..     ...                            ...                  ...   
395    0.0                          140.0                 49.0   
396    0.0                           75.0                 31.0   
397    0.0                          100.0                 26.0   
398    0.0                          114.0                 50.0   
399    0.0                          131.0                 18.0   

     Serum Creatinine (mgs/dL)  Sodium (mEq/L)  Potassium (mEq/L)  ...  \
0                          1.2             NaN                NaN  ...   
1                          0.8             NaN                NaN  ...   
2                          1.8             NaN                NaN  ...   
3                          3.8           111.0                2.5  ...   
4                          1.4             NaN                NaN  ...   
..                         ...             ...                ...  ...   
395                        0.5           150.0                4.9  ...   
396                        1.2           141.0                3.5  ...   
397                        0.6           137.0                4.4  ...   
398                        1.0           135.0                4.9  ...   
399                        1.1           141.0                3.5  ...   

     Coronary Artery Disease_yes Appetite_good Appetite_poor Pedal Edema_no  \
0                              0             1             0              1   
1                              0             1             0              1   
2                              0             0             1              1   
3                              0             0             1              0   
4                              0             1             0              1   
..                           ...           ...           ...            ...   
395                            0             1             0              1   
396                            0             1             0              1
```

```
     Pedal Edema_yes  Anemia_no  Anemia_yes  Chronic Kidney Disease Label_ckd  \
0                  0          1           0                                 1   
1                  0          1           0                                 1   
2                  0          0           1                                 1   
3                  1          0           1                                 1   
4                  0          1           0                                 1   
..               ...        ...         ...                               ...   
395                0          1           0                                 0   
396                0          1           0                                 0   
397                0          1           0                                 0   
398                0          1           0                                 0   
399                0          1           0                                 0   

     Chronic Kidney Disease Label_ckd\t  Chronic Kidney Disease Label_notckd  
0                                     0                                    0  
1                                     0                                    0  
2                                     0                                    0  
3                                     0                                    0  
4                                     0                                    0  
..                                  ...                                  ...  
395                                   0                                    1  
396                                   0                                    1  
397                                   0                                    1  
398                                   0                                    1  
399                                   0                                    1  

[400 rows x 41 columns]
```

- Some industry techniques that were used for **data imputation** and they are as follows for my and reader's reference. Okay before that let me clarify what is the meaning of data imputation. **Data imputation in simple terms means technique that is used to fill in missing values in a dataset.** Please note that each of these techniques below has its own strengths and limitations. For a brief overview of their general applicability sharing them as follows:
- **Mean or Median Imputation**: This technique is simple and fast, making it useful when the missing data is randomly distributed and missingness is low. However, it doesn't account for the relationships between variables, potentially this has a tendency to lead to biased estimates so we need to be careful here. The mean imputation assumes that the missing values are missing at random and that the mean is a suitable estimate for the missing values.

Applying Data Imputation

```
#  Using Mean Imputation for initial check, using data.fillna(data.mean()), the function replaces missing values with the mean value of the corresponding column

data_imputed_mean = data_encoded.fillna(data_encoded.mean())
```

```
<ipython-input-77-1eadfc7baa7c>:3: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None
  data_imputed_mean = data_encoded.fillna(data_encoded.mean())
```

```
# If the printed data_imputed DataFrame has fewer rows than the original data DataFrame, it suggests that some rows had missing values in all or most of their columns.
# When using data.fillna(data.mean()), the function replaces missing values with the mean value of the corresponding column. However, if an entire row has missing values,
# it cannot be imputed with the mean, resulting in a row with all NaN values

data_imputed_mean
```

	Age (yrs)	Blood Pressure (mm/Hg)	Specific Gravity	Albumin (g/dL)	Sugar	Blood Glucose Random (mgs/dL)	Blood Urea (mgs/dL)	Serum Creatinine (mgs/dL)	Sodium (mEq/L)	Potassium (mEq/L)	...	Coronary Artery Disease_yes	Appetite_good	Appetite_poor	Pedal Edema_no	Pedal Edema_yes	Anemia_no	Anemia_yes	Chronic Kidney Disease Label_ckd
0	48.0	80.0	1.020	1.0	0.0	121.000000	36.0	1.2	137.528754	4.627244	...	0	1	0	1	0	1	0	1
1	7.0	50.0	1.020	4.0	0.0	148.036517	18.0	0.8	137.528754	4.627244	...	0	1	0	1	0	1	0	1
2	62.0	80.0	1.010	2.0	3.0	423.000000	53.0	1.8	137.528754	4.627244	...	0	0	1	1	0	0	1	1
3	48.0	70.0	1.005	4.0	0.0	117.000000	56.0	3.8	111.000000	2.500000	...	0	0	1	0	1	0	1	1
4	51.0	80.0	1.010	2.0	0.0	106.000000	26.0	1.4	137.528754	4.627244	...	0	1	0	1	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

✓ 0s completed at 4:59 AM

```
# To determine if there are any rows that have missing values even after the imputation process.Mostly with Categorical data

# Check for missing values in the original data
missing_rows_original = data[data.isnull().any(axis=1)]

# Check for missing values in the imputed data
missing_rows_imputed = data_imputed_mean[data_imputed_mean.isnull().any(axis=1)]

# Print the number of missing rows
print("Missing rows in original data:", len(missing_rows_original))
print("Missing rows in imputed data:", len(missing_rows_imputed))
```

```
Missing rows in original data: 242
Missing rows in imputed data: 135
```

- **Hot Deck Imputation**: This technique can be effective when there is a pattern or similarity among data points and missingness is not completely random. It preserves the structure of the dataset, but it may not be appropriate if there are no clear patterns or if variables are not related. This is where we can generate the frequency distribution as well so observe some pattern but would not necessarily mean causality between variables would exist.
- **Regression Imputation:** Regression-based imputation is beneficial when there are strong relationships between variables and missingness is not completely random. It leverages the relationships to estimate missing values, but it assumes linearity and may introduce bias if the relationships are not accurately captured.
- **Multiple Imputation:** Multiple imputation provides more accurate estimates by considering the uncertainty associated with missing values. It is suitable when missingness is not completely random and when there is substantial missing data. However, it requires more computational resources and may be complex to implement.
- **K-Nearest Neighbour (KNN) Imputation**: KNN imputation is useful when there is similarity or clustering among data points. It takes into account the relationships between variables and can

handle both continuous and categorical data. However, it can be computationally expensive, especially with large datasets. But since our dataset here is relatively small so I preferred to use this and this addresses for both numerical and categorical. The numerical data here is more of continuous type data and not discrete data. In KNN imputation, the n_neighbors parameter determines the number of nearest neighbors used to impute missing values. Choosing an appropriate value for n_neighbors is important as it can impact the imputation results. A few considerations to help decide on the n_neighbors parameter:

- **Size of the dataset:** If you have a large dataset, you can consider using a larger value for n_neighbors as you have more data points to find nearest neighbors from. However, keep in mind that a very large n_neighbors value can increase the computational complexity.
- **Sparsity of data:** If dataset has a low density of data points or if missing values are widespread, a larger n_neighbors value may be beneficial. This can help capture a broader range of data points and improve the imputation accuracy.
- **Similarity of instances:** Consider the inherent similarity between instances in dataset. If the instances are similar and missing values are expected to be imputed using nearby neighbours, a smaller n_neighbors value can be sufficient. On the other hand, if instances are diverse and the missing values may need information from distant neighbors, a larger n_neighbors value may be more appropriate.
- **Cross-validation:** can perform cross-validation to evaluate different n_neighbors values and choose the one that results in the best imputation performance. This helps in assessing the impact of different n_neighbors values on imputation accuracy and generalization to unseen data.
- It is recommended to experiment with different values of n_neighbors and observe the imputation results in terms of imputation accuracy and the ability to preserve the underlying patterns and relationships in the data.

10. Building the prediction model using Supervised Binary classification technique such as Logistic Regression

Logistic Regression is a statistical algorithm used for binary classification tasks. As an AI engineer , I find Logistic Regression to be a powerful and commonly used method for predicting the probability of a binary outcome.

The concept behind Logistic Regression is to model the relationship between the independent variables, also known as features or predictors, and the dependent variable, which represents the binary outcome we want to predict. This algorithm is particularly suitable when the dependent variable follows a binomial distribution.

To train a Logistic Regression model, I start by collecting a labeled dataset where each observation is labeled with the corresponding binary outcome. Then, I divide the data into a training set and a test set to evaluate the model's performance.

During the training phase, I use an optimization algorithm called Maximum Likelihood Estimation to estimate the parameters of the logistic function. The logistic function, also known as the sigmoid function, maps the linear combination of the features and parameters onto a range between 0 and 1, representing the probability of the positive outcome.

Once the model is trained, I can use it to make predictions on new, unseen data. By applying the learned weights and biases to the input features, I obtain a probability score. By applying a threshold to this score, typically 0.5, I can classify the observation into one of the two classes.

What makes Logistic Regression so useful is its interpretability. The model provides insights into the importance of each feature by estimating the coefficients associated with them. These coefficients indicate the direction and strength of the relationship between the feature and the probability of the positive outcome.

However, Logistic Regression assumes a linear relationship between the features and the log-odds of the outcome. If the relationship is non-linear, I may need to consider using more complex models or applying transformations to the features.

Overall, Logistic Regression is a versatile algorithm that allows me to predict binary outcomes based on a set of features such as in this case I used it for CKD patient classification. Its interpretability and simplicity make it a popular choice in various domains, including healthcare, finance, and social sciences.

11. Hardware:

Using Windows 10 Operating system on 64-bit

12. Software:

- **Windows 10 OS**
- **Visual Studio** Editor for programming
- **Google Chrome for web browsing**
- **Flask** Framework for Python based application
- **Python** language
    - **Packages used for Backend development:**

```
from flask import Flask, render_template,request,jsonify,redirect,url_for
# from jinja2 import escape
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.style as style
style.use('classic')
import seaborn as sns
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.impute import KNNImputer
from sklearn.metrics import accuracy_score
import warnings
# from sklearn.externals import joblib
import joblib
```

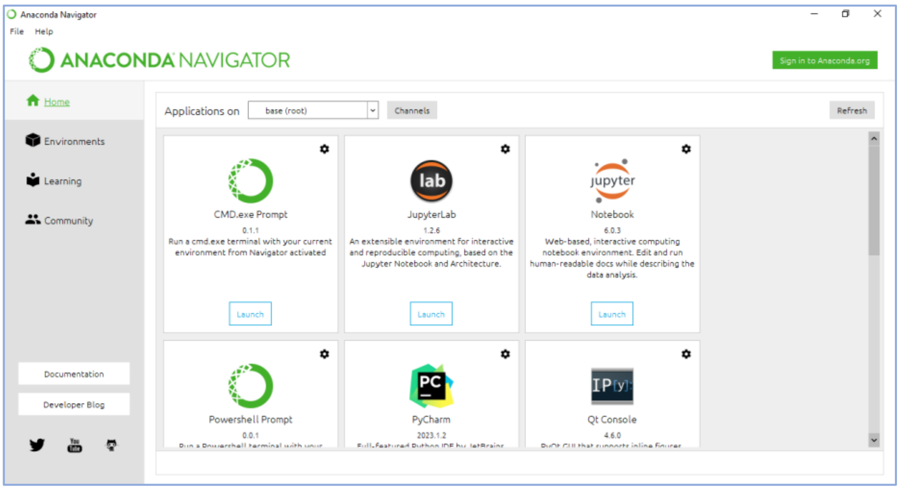
- **Packages used for Frontend development:**
- **Setting up Streamlit environment for web app using Anaconda Navigator and Conda Prompt**

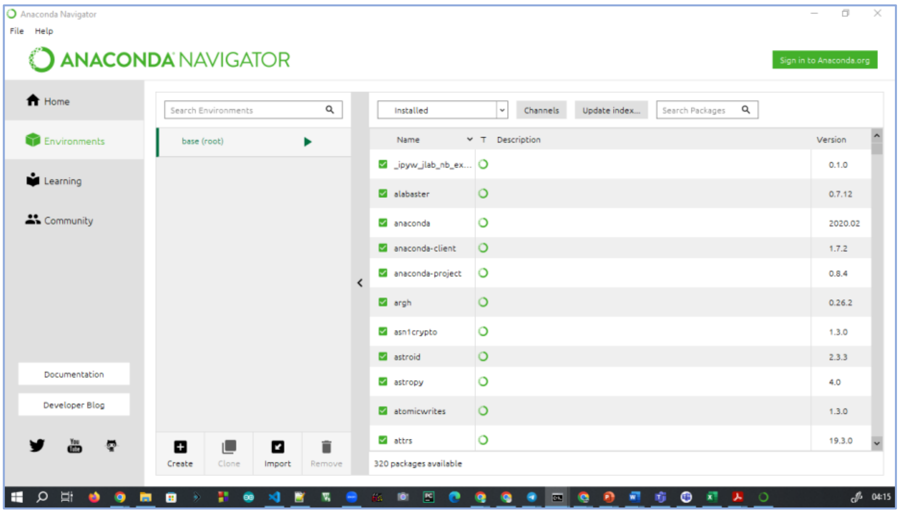
```
o   import streamlit as st
o   import pandas as pd
o   import numpy as np
o   import seaborn as sns
o   import matplotlib.pyplot as plt
o   import matplotlib.style as style
```

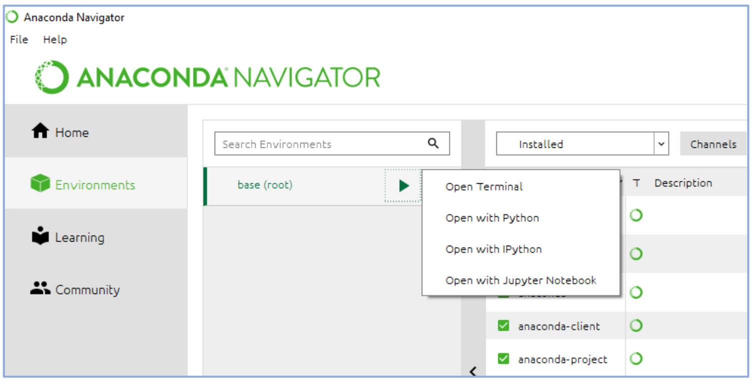
- **HTML** scripting language
  - **For frontend HTML Page scripting**

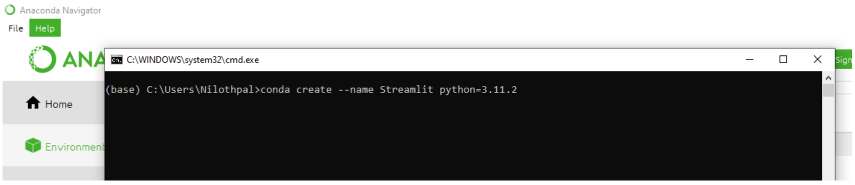
StreamLit setup process using Anaconda Navigator:

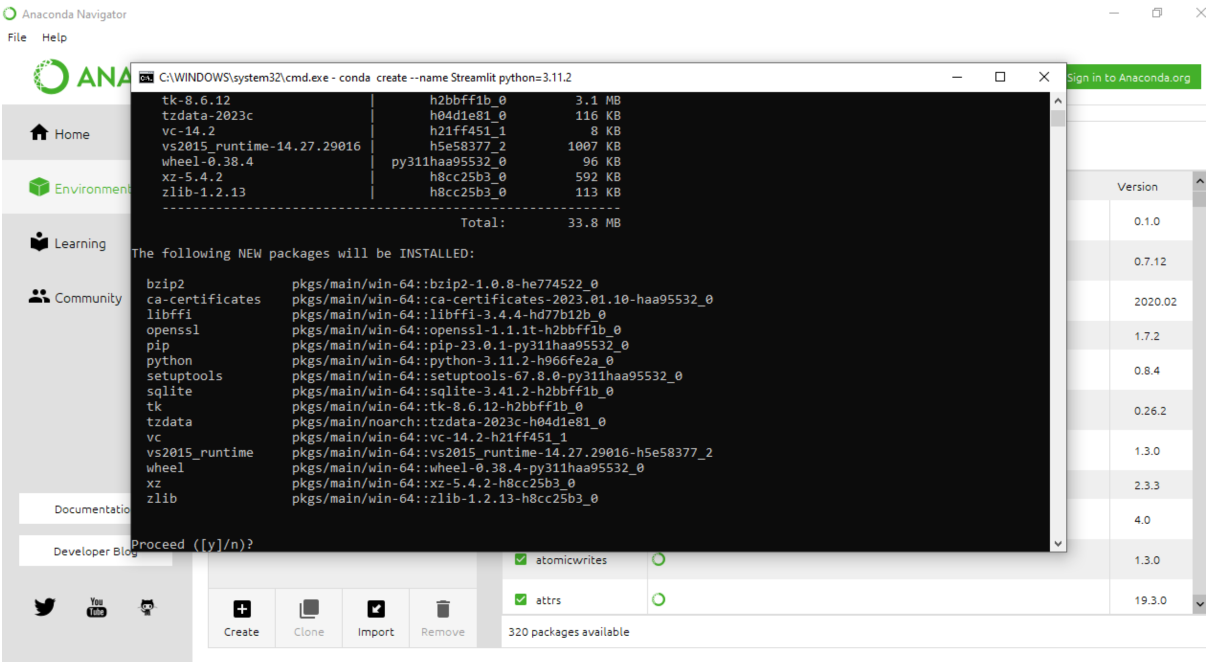
Setting up environment for Streamlit- Dashboard based Projects

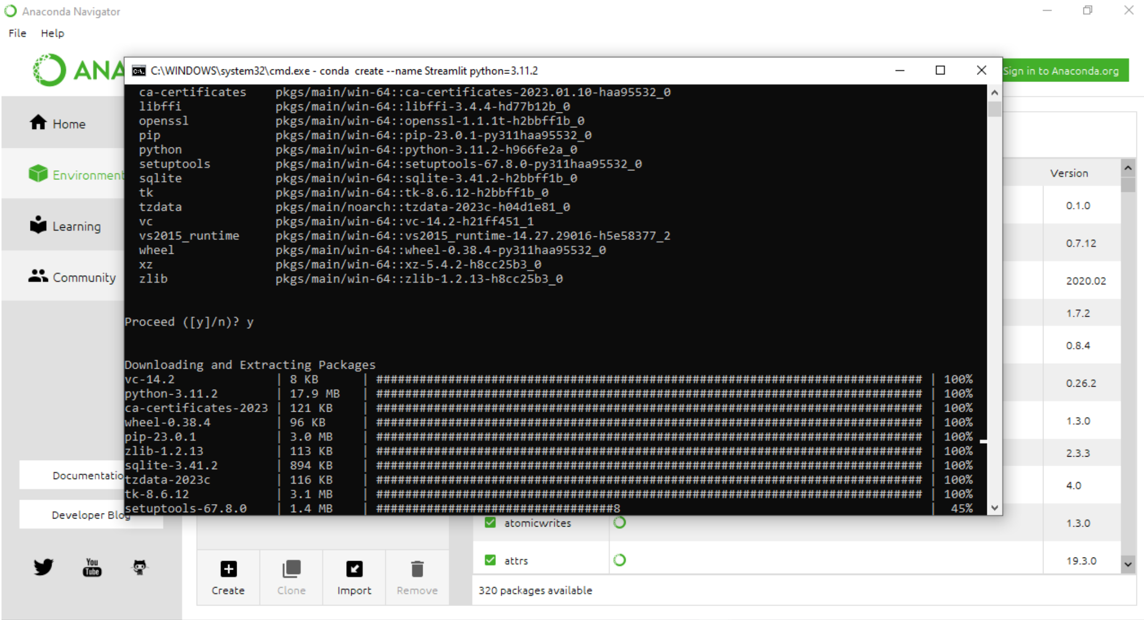




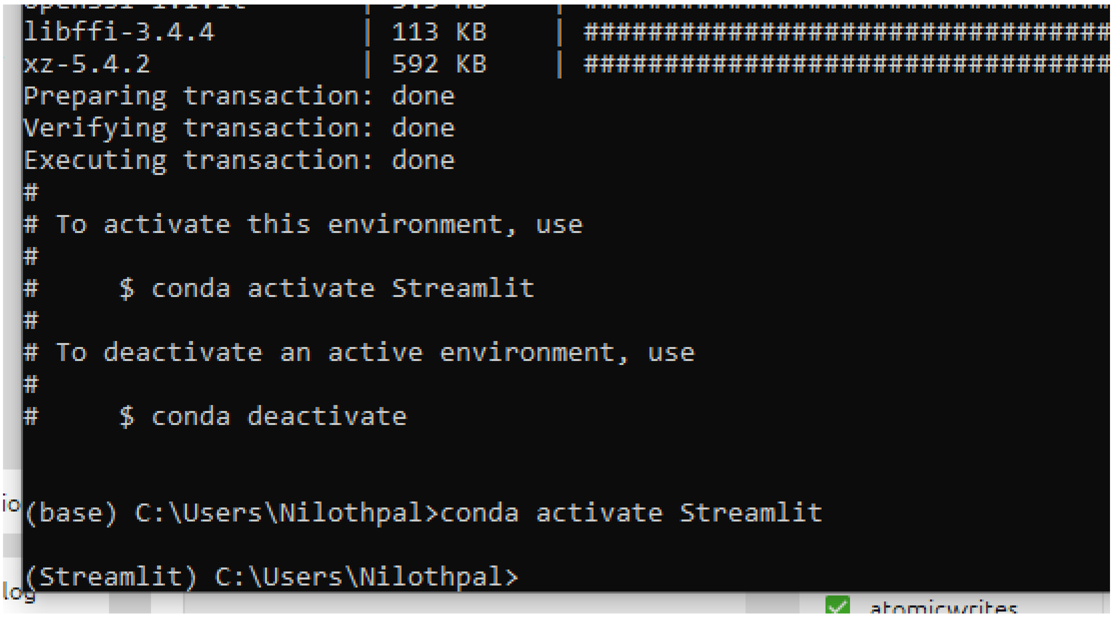

Anaconda Navigator
File Help
ANACONDA NAVIGATOR
Home
Environments
Learning
Community
Search Environments
base (root)
Installed
Channels
Open Terminal
Open with Python
Open with IPython
Open with Jupyter Notebook
T Description
anaconda-client
anaconda-project


C:\WINDOWS\system32\cmd.exe
(base) C:\Users\Nilothpal>conda create --name Streamlit python=3.11.2

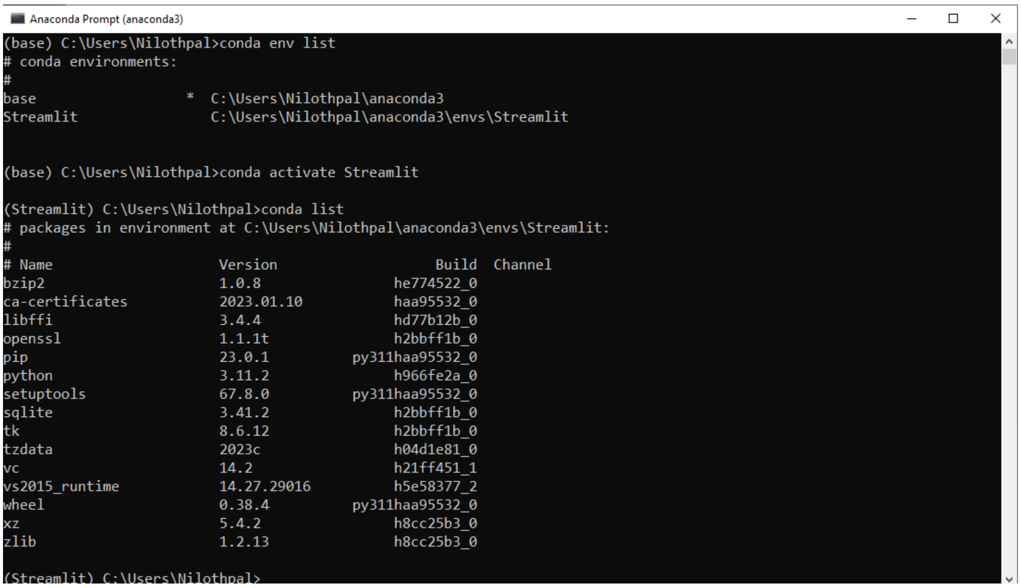

Anaconda Navigator
File Help
C:\WINDOWS\system32\cmd.exe - conda create --name Streamlit python=3.11.2
tk-8.6.12 | h2bbff1b_0 3.1 MB
tzdata-2023c | h04d1e81_0 116 KB
vc-14.2 | h21ff451_1 8 KB
vs2015_runtime-14.27.29016 | h5e58377_2 1007 KB
wheel-0.38.4 | py311haa95532_0 96 KB
xz-5.4.2 | h8cc25b3_0 592 KB
zlib-1.2.13 | h8cc25b3_0 113 KB
Total: 33.8 MB
The following NEW packages will be INSTALLED:
bzip2 pkgs/main/win-64::bzip2-1.0.8-he774522_0
ca-certificates pkgs/main/win-64::ca-certificates-2023.01.10-haa95532_0
libffi pkgs/main/win-64::libffi-3.4.4-hd77b12b_0
openssl pkgs/main/win-64::openssl-1.1.1t-h2bbff1b_0
pip pkgs/main/win-64::pip-23.0.1-py311haa95532_0
python pkgs/main/win-64::python-3.11.2-h966fe2a_0
setuptools pkgs/main/win-64::setuptools-67.8.0-py311haa95532_0
sqlite pkgs/main/win-64::sqlite-3.41.2-h2bbff1b_0
tk pkgs/main/win-64::tk-8.6.12-h2bbff1b_0
tzdata pkgs/main/noarch::tzdata-2023c-h04d1e81_0
vc pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel pkgs/main/win-64::wheel-0.38.4-py311haa95532_0
xz pkgs/main/win-64::xz-5.4.2-h8cc25b3_0
zlib pkgs/main/win-64::zlib-1.2.13-h8cc25b3_0
Proceed ([y]/n)?
Sign in to Anaconda.org
Home
Learning
Community
Version
0.1.0
0.7.12
2020.02
1.7.2
0.8.4
0.26.2
1.3.0
2.3.3
4.0
1.3.0
19.3.0
atomicwrites
attrs
Create
Clone
Import
Remove
320 packages available



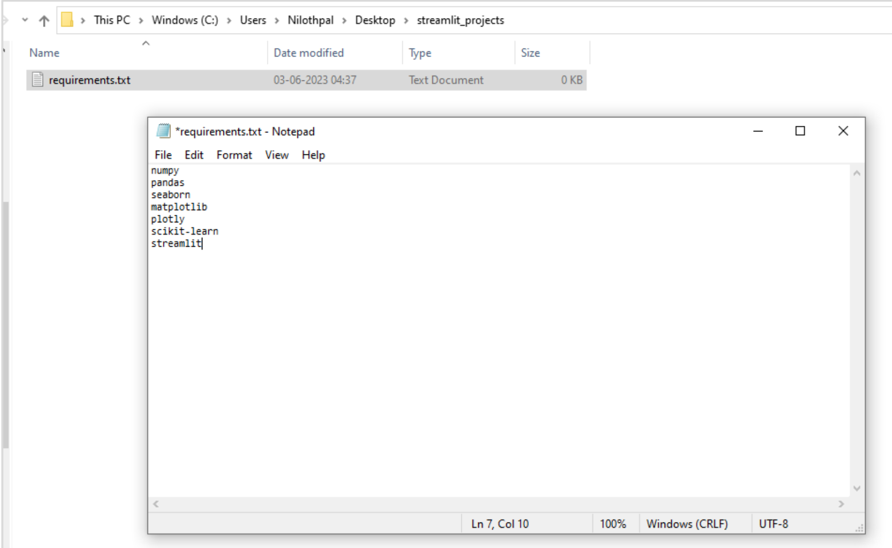

```
  ca-certificates    pkgs/main/win-64::ca-certificates-2023.01.10-haa95532_0
  libffi             pkgs/main/win-64::libffi-3.4.4-hd77b12b_0
  openssl            pkgs/main/win-64::openssl-1.1.1t-h2bbff1b_0
  pip                pkgs/main/win-64::pip-23.0.1-py311haa95532_0
  python             pkgs/main/win-64::python-3.11.2-h966fe2a_0
  setuptools         pkgs/main/win-64::setuptools-67.8.0-py311haa95532_0
  sqlite             pkgs/main/win-64::sqlite-3.41.2-h2bbff1b_0
  tk                 pkgs/main/win-64::tk-8.6.12-h2bbff1b_0
  tzdata             pkgs/main/noarch::tzdata-2023c-h04d1e81_0
  vc                 pkgs/main/win-64::vc-14.2-h21ff451_1
  vs2015_runtime     pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
  wheel              pkgs/main/win-64::wheel-0.38.4-py311haa95532_0
  xz                 pkgs/main/win-64::xz-5.4.2-h8cc25b3_0
  zlib               pkgs/main/win-64::zlib-1.2.13-h8cc25b3_0


Proceed ([y]/n)? y


Downloading and Extracting Packages
vc-14.2              | 8 KB      | ############################################################################ | 100%
python-3.11.2        | 17.9 MB   | ############################################################################ | 100%
ca-certificates-2023 | 121 KB    | ############################################################################ | 100%
wheel-0.38.4         | 96 KB     | ############################################################################ | 100%
pip-23.0.1           | 3.0 MB    | ############################################################################ | 100%
zlib-1.2.13          | 113 KB    | ############################################################################ | 100%
sqlite-3.41.2        | 894 KB    | ############################################################################ | 100%
tzdata-2023c         | 116 KB    | ############################################################################ | 100%
tk-8.6.12            | 3.1 MB    | ############################################################################ | 100%
setuptools-67.8.0    | 1.4 MB    | ##################################                                           |  45%
```

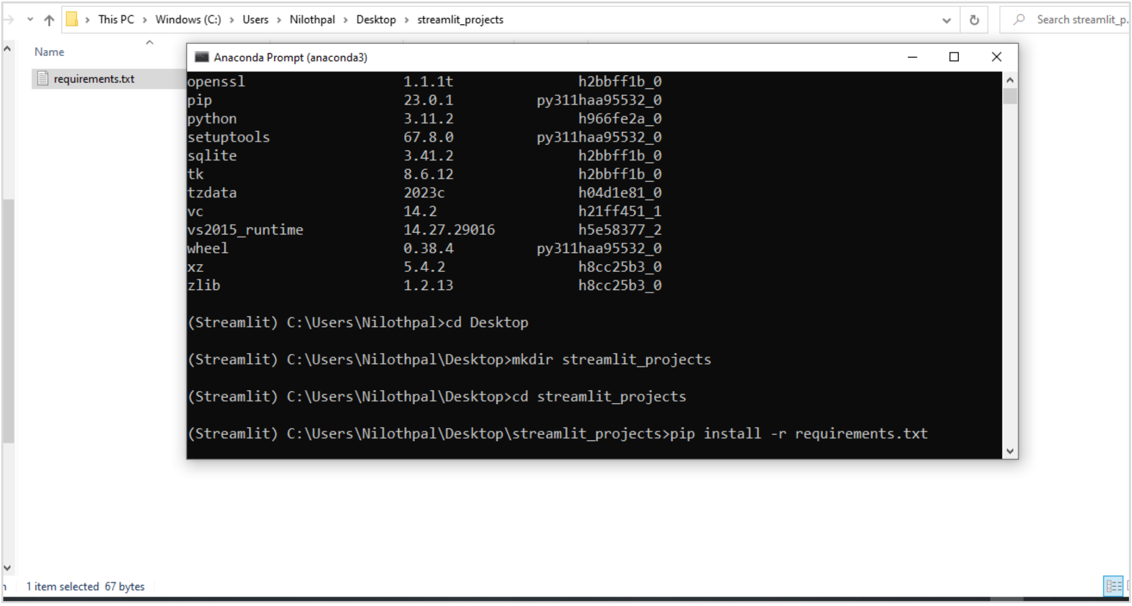


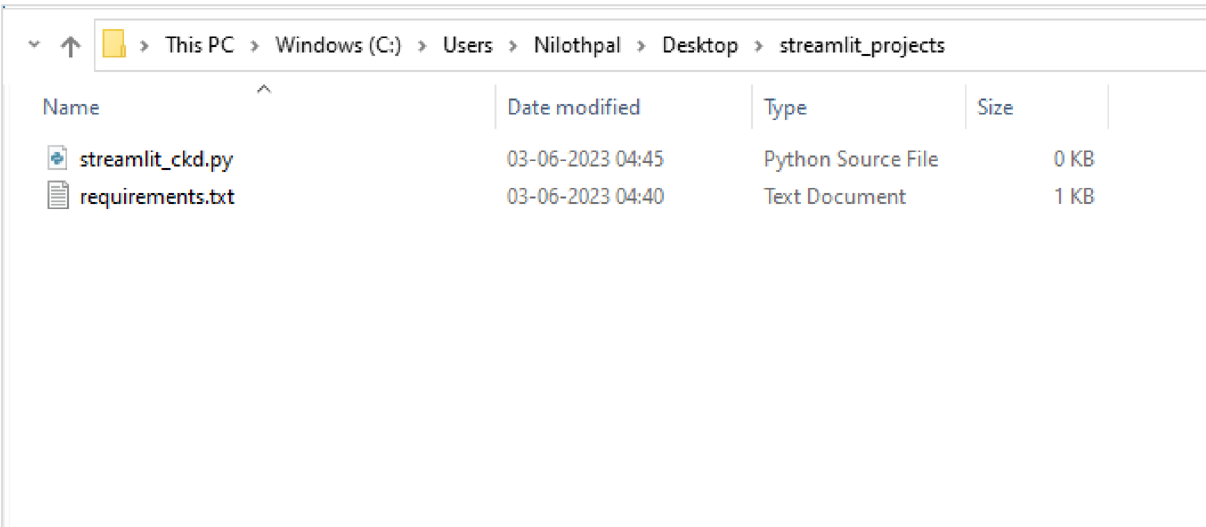
```
libffi-3.4.4         | 113 KB    | ##################################
xz-5.4.2             | 592 KB    | ##################################
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate Streamlit
#
# To deactivate an active environment, use
#
#     $ conda deactivate


(base) C:\Users\Nilothpal>conda activate Streamlit

(Streamlit) C:\Users\Nilothpal>
```

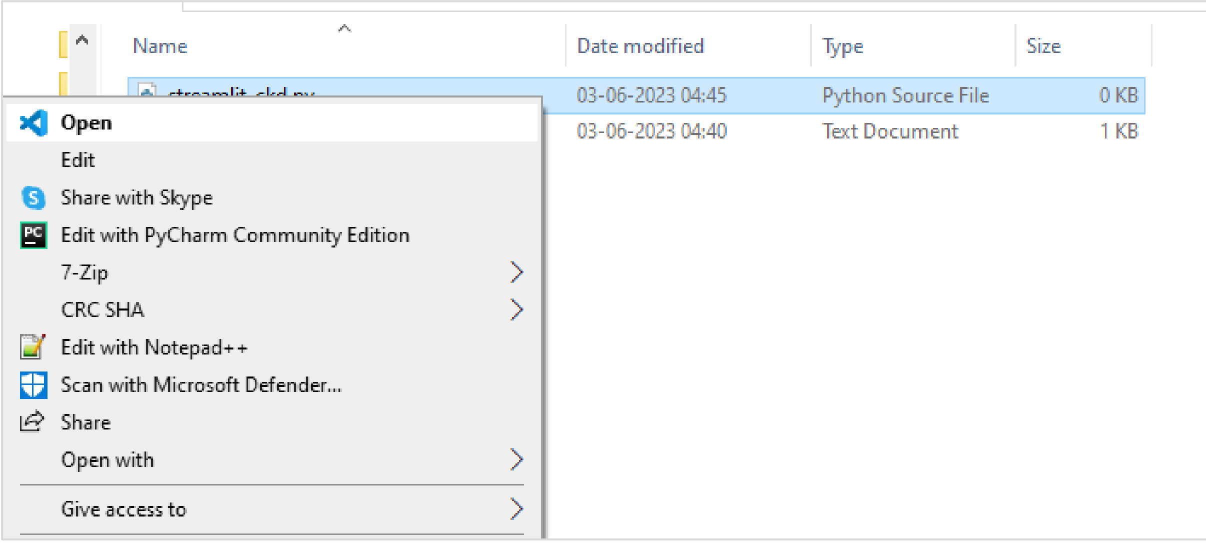



```
(base) C:\Users\Nilothpal>conda env list
# conda environments:
#
base                  *  C:\Users\Nilothpal\anaconda3
Streamlit                C:\Users\Nilothpal\anaconda3\envs\Streamlit


(base) C:\Users\Nilothpal>conda activate Streamlit

(Streamlit) C:\Users\Nilothpal>conda list
# packages in environment at C:\Users\Nilothpal\anaconda3\envs\Streamlit:
#
# Name                    Version                   Build  Channel
bzip2                     1.0.8                he774522_0
ca-certificates           2023.01.10           haa95532_0
libffi                    3.4.4                hd77b12b_0
openssl                   1.1.1t               h2bbff1b_0
pip                       23.0.1          py311haa95532_0
python                    3.11.2               h966fe2a_0
setuptools                67.8.0          py311haa95532_0
sqlite                    3.41.2               h2bbff1b_0
tk                        8.6.12               h2bbff1b_0
tzdata                    2023c                h04d1e81_0
vc                        14.2                 h21ff451_1
vs2015_runtime            14.27.29016          h5e58377_2
wheel                     0.38.4          py311haa95532_0
xz                        5.4.2                h8cc25b3_0
zlib                      1.2.13               h8cc25b3_0

(Streamlit) C:\Users\Nilothpal>
```

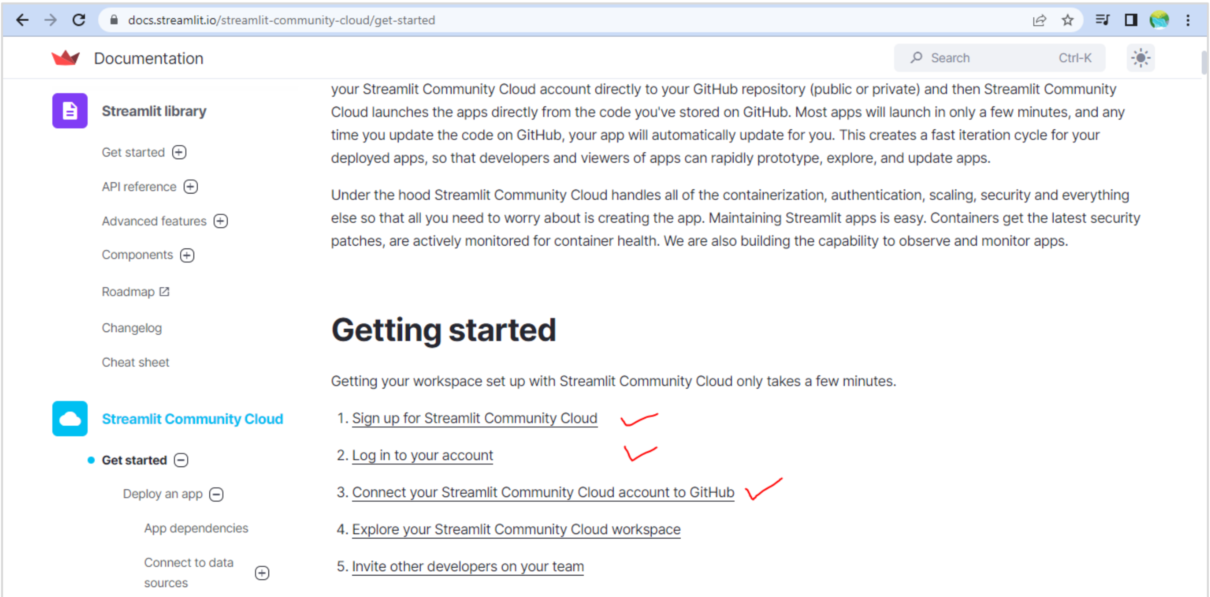

This PC > Windows (C:) > Users > Nilothpal > Desktop > streamlit_projects
Name
Date modified
Type
Size
requirements.txt
03-06-2023 04:37
Text Document
0 KB
*requirements.txt - Notepad
File Edit Format View Help
numpy
pandas
seaborn
matplotlib
plotly
scikit-learn
streamlit
Ln 7, Col 10
100%
Windows (CRLF)
UTF-8

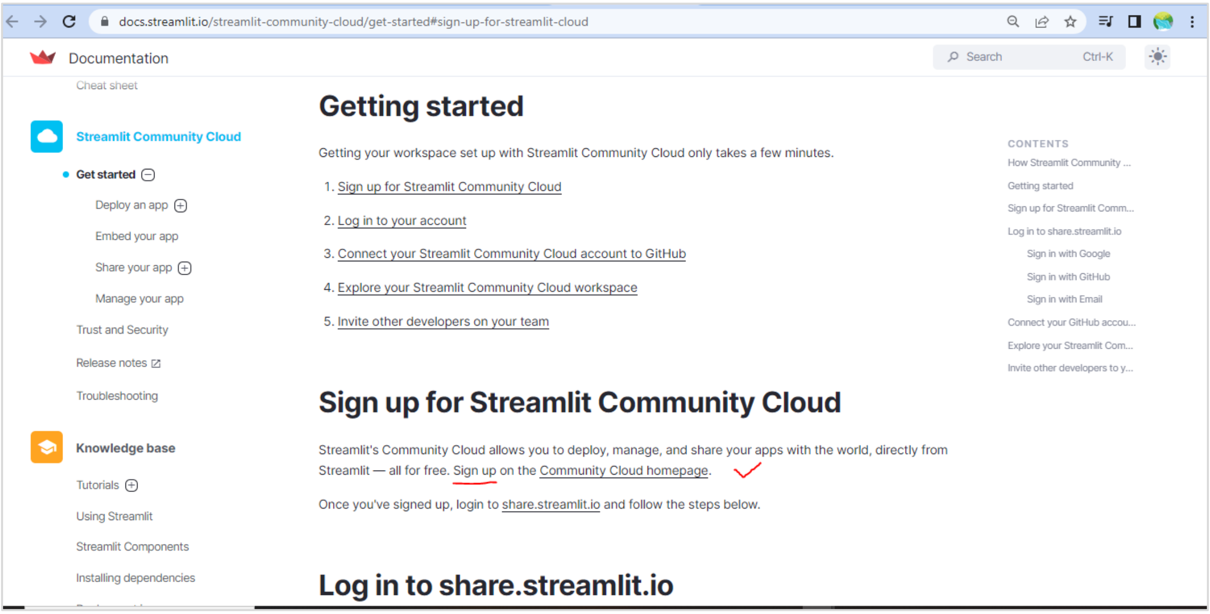

This PC > Windows (C:) > Users > Nilothpal > Desktop > streamlit_projects
Search streamlit_p...
Name
requirements.txt
Anaconda Prompt (anaconda3)
openssl 1.1.1t h2bbff1b_0
pip 23.0.1 py311haa95532_0
python 3.11.2 h966fe2a_0
setuptools 67.8.0 py311haa95532_0
sqlite 3.41.2 h2bbff1b_0
tk 8.6.12 h2bbff1b_0
tzdata 2023c h04d1e81_0
vc 14.2 h21ff451_1
vs2015_runtime 14.27.29016 h5e58377_2
wheel 0.38.4 py311haa95532_0
xz 5.4.2 h8cc25b3_0
zlib 1.2.13 h8cc25b3_0
(Streamlit) C:\Users\Nilothpal>cd Desktop
(Streamlit) C:\Users\Nilothpal\Desktop>mkdir streamlit_projects
(Streamlit) C:\Users\Nilothpal\Desktop>cd streamlit_projects
(Streamlit) C:\Users\Nilothpal\Desktop\streamlit_projects>pip install -r requirements.txt
1 item selected 67 bytes

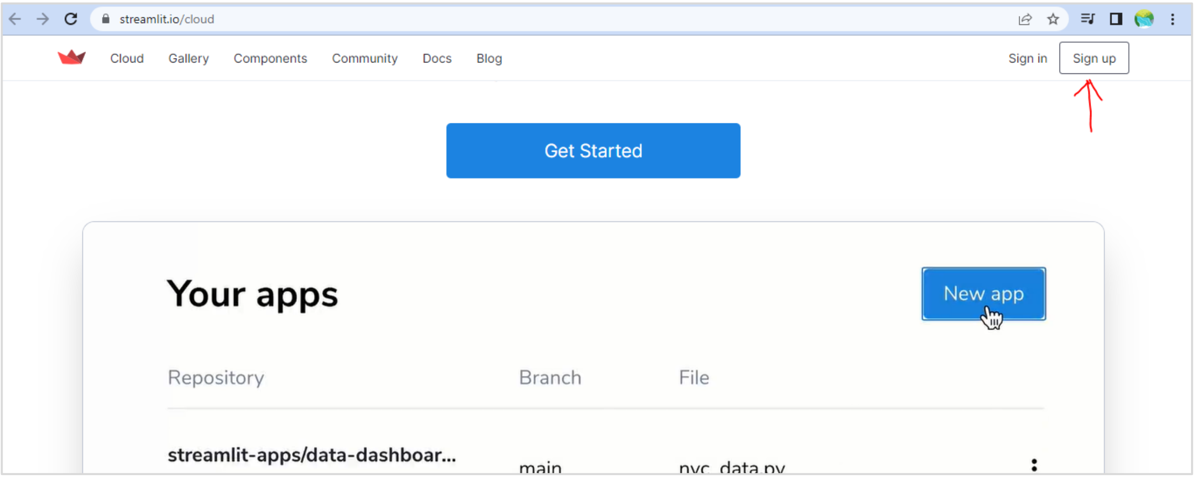

This PC > Windows (C:) > Users > Nilothpal > Desktop > streamlit_projects
Name
Date modified
Type
Size
streamlit_ckd.py
03-06-2023 04:45
Python Source File
0 KB
requirements.txt
03-06-2023 04:40
Text Document
1 KB

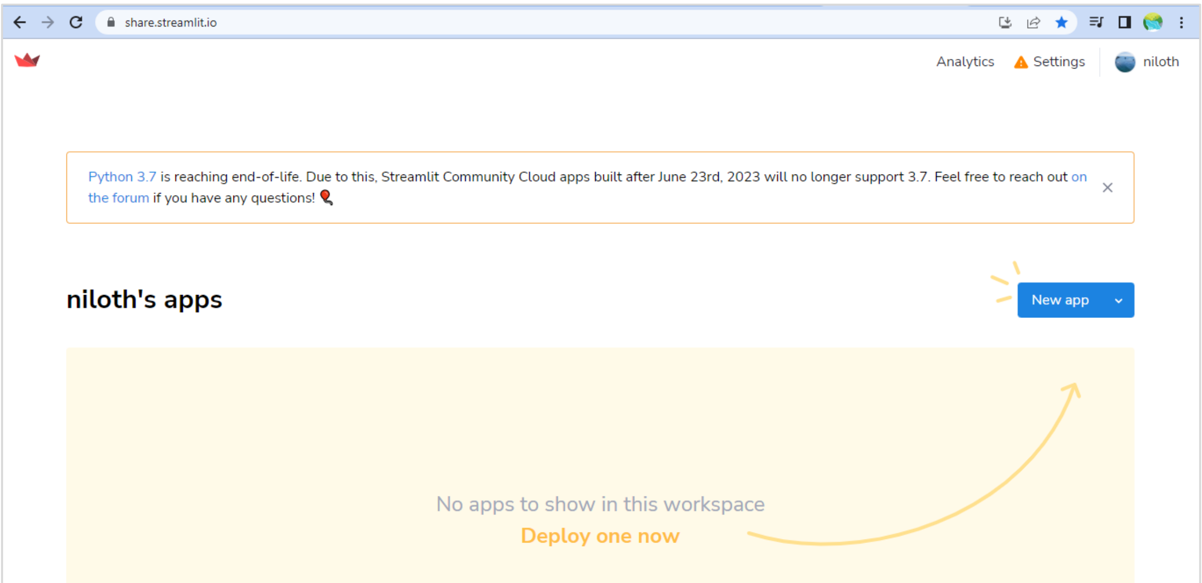


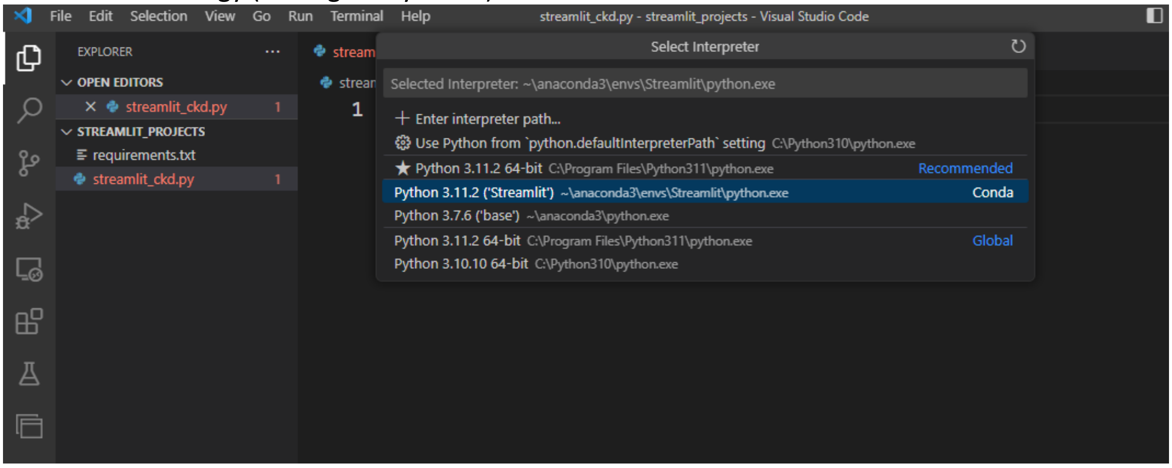
Setting up Streamlit Web App process

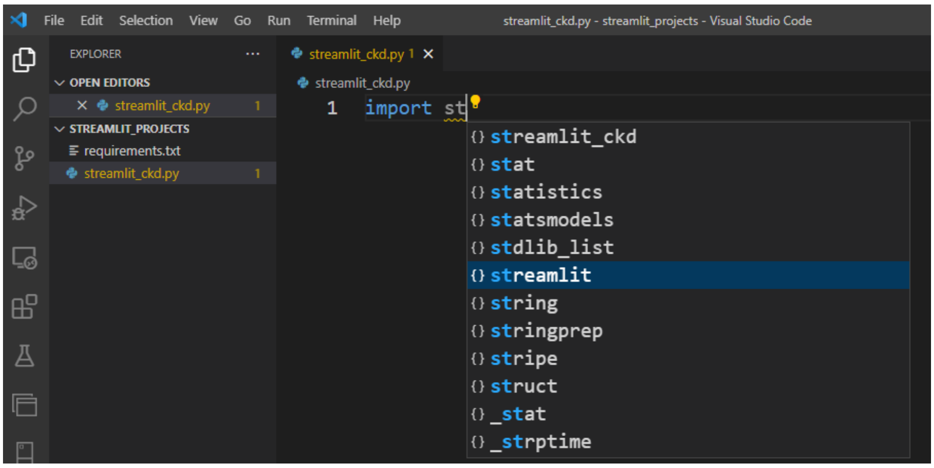


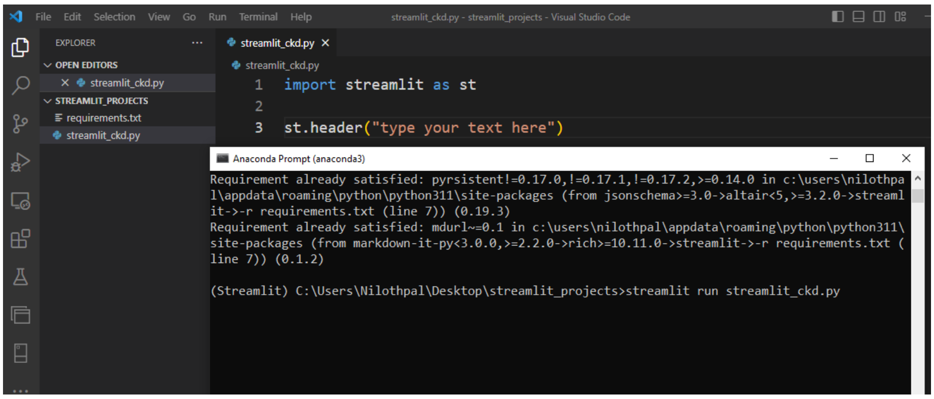

docs.streamlit.io/streamlit-community-cloud/get-started#sign-up-for-streamlit-cloud
Documentation
Search
Ctrl-K
Cheat sheet
Streamlit Community Cloud
Get started
Deploy an app
Embed your app
Share your app
Manage your app
Trust and Security
Release notes
Troubleshooting
Knowledge base
Tutorials
Using Streamlit
Streamlit Components
Installing dependencies
Getting started
Getting your workspace set up with Streamlit Community Cloud only takes a few minutes.
1. Sign up for Streamlit Community Cloud
2. Log in to your account
3. Connect your Streamlit Community Cloud account to GitHub
4. Explore your Streamlit Community Cloud workspace
5. Invite other developers on your team
Sign up for Streamlit Community Cloud
Streamlit's Community Cloud allows you to deploy, manage, and share your apps with the world, directly from Streamlit — all for free. Sign up on the Community Cloud homepage.
Once you've signed up, login to share.streamlit.io and follow the steps below.
Log in to share.streamlit.io
CONTENTS
How Streamlit Community ...
Getting started
Sign up for Streamlit Comm...
Log in to share.streamlit.io
Sign in with Google
Sign in with GitHub
Sign in with Email
Connect your GitHub accou...
Explore your Streamlit Com...
Invite other developers to y...

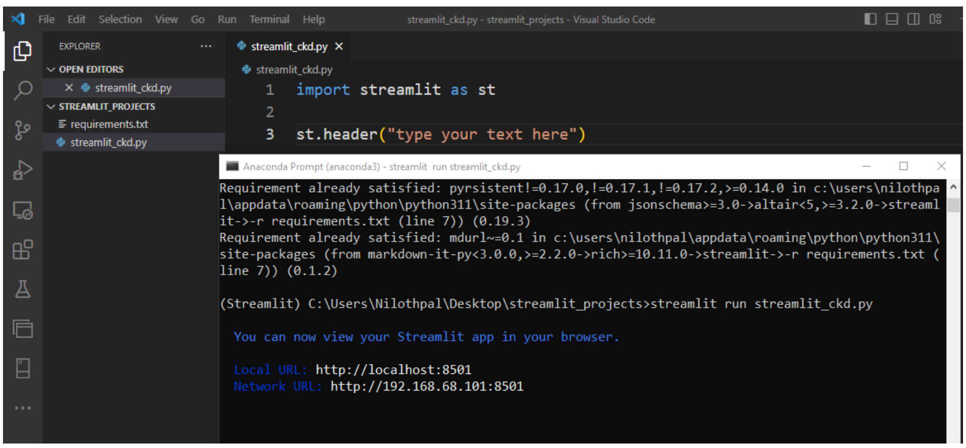

streamlit.io/cloud
Cloud
Gallery
Components
Community
Docs
Blog
Sign in
Sign up
Get Started
Your apps
New app
Repository
Branch
File
streamlit-apps/data-dashboar...
main
nyc_data.py

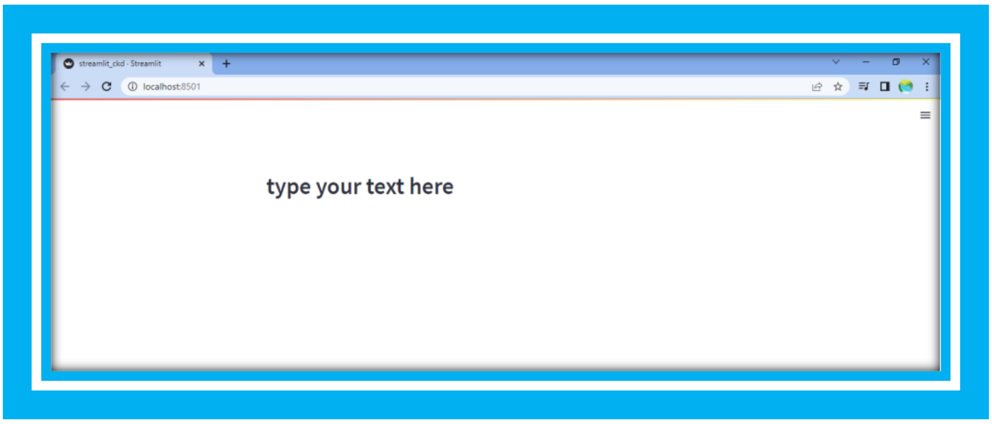

share.streamlit.io
Analytics
Settings
niloth
Python 3.7 is reaching end-of-life. Due to this, Streamlit Community Cloud apps built after June 23rd, 2023 will no longer support 3.7. Feel free to reach out on the forum if you have any questions!
niloth's apps
New app
No apps to show in this workspace
Deploy one now


File Edit Selection View Go Run Terminal Help
streamlit_ckd.py - streamlit_projects - Visual Studio Code
EXPLORER
OPEN EDITORS
streamlit_ckd.py 1
STREAMLIT_PROJECTS
requirements.txt
streamlit_ckd.py 1
Select Interpreter
Selected Interpreter: ~\anaconda3\envs\Streamlit\python.exe
+ Enter interpreter path...
Use Python from `python.defaultInterpreterPath` setting C:\Python310\python.exe
Python 3.11.2 64-bit C:\Program Files\Python311\python.exe
Recommended
Python 3.11.2 ('Streamlit') ~\anaconda3\envs\Streamlit\python.exe
Conda
Python 3.7.6 ('base') ~\anaconda3\python.exe
Python 3.11.2 64-bit C:\Program Files\Python311\python.exe
Global
Python 3.10.10 64-bit C:\Python310\python.exe


File Edit Selection View Go Run Terminal Help
streamlit_ckd.py - streamlit_projects - Visual Studio Code
EXPLORER
OPEN EDITORS
streamlit_ckd.py 1
STREAMLIT_PROJECTS
requirements.txt
streamlit_ckd.py 1
1 import st
streamlit_ckd
stat
statistics
statsmodels
stdlib_list
streamlit
string
stringprep
stripe
struct
_stat
_strptime


File Edit Selection View Go Run Terminal Help
streamlit_ckd.py - streamlit_projects - Visual Studio Code
EXPLORER
OPEN EDITORS
streamlit_ckd.py
STREAMLIT_PROJECTS
requirements.txt
streamlit_ckd.py
1 import streamlit as st
2
3 st.header("type your text here")
Anaconda Prompt (anaconda3)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in c:\users\nilothpal\appdata\roaming\python\python311\site-packages (from jsonschema>=3.0->altair<5,>=3.2.0->streamlit->-r requirements.txt (line 7)) (0.19.3)
Requirement already satisfied: mdurl~=0.1 in c:\users\nilothpal\appdata\roaming\python\python311\site-packages (from markdown-it-py<3.0.0,>=2.2.0->rich>=10.11.0->streamlit->-r requirements.txt (line 7)) (0.1.2)
(Streamlit) C:\Users\Nilothpal\Desktop\streamlit_projects>streamlit run streamlit_ckd.py


File Edit Selection View Go Run Terminal Help
streamlit_ckd.py - streamlit_projects - Visual Studio Code
EXPLORER
OPEN EDITORS
streamlit_ckd.py
STREAMLIT_PROJECTS
requirements.txt
streamlit_ckd.py
1 import streamlit as st
2
3 st.header("type your text here")
Anaconda Prompt (anaconda3) - streamlit run streamlit_ckd.py
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in c:\users\nilothpal\appdata\roaming\python\python311\site-packages (from jsonschema>=3.0->altair<5,>=3.2.0->streamlit->-r requirements.txt (line 7)) (0.19.3)
Requirement already satisfied: mdurl~=0.1 in c:\users\nilothpal\appdata\roaming\python\python311\site-packages (from markdown-it-py<3.0.0,>=2.2.0->rich>=10.11.0->streamlit->-r requirements.txt (line 7)) (0.1.2)
(Streamlit) C:\Users\Nilothpal\Desktop\streamlit_projects>streamlit run streamlit_ckd.py
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8501
Network URL: http://192.168.68.101:8501



Your **Streamlit Web Application** has started by its own on Google Chrome web-browser

streamlit_ckd.py

```
import streamlit as st

st.header("type your text here")
st.text(" Now I am about to start developing the CKD Analysis Dashboard")
```

13. References:

- https://flask.palletsprojects.com/en/2.3.x/ -use as a reference for Flask documentation, tutorials, templates, testing flask application, handling/debugging application errors, CLI, Working with Shell etc.
- https://nkfs.org/about-us/key-statistics/
- https://www.straitstimes.com/singapore/health/chronic-kidney-disease-on-the-rise-in-singapore-nkf-medical-director
- https://www.memc.com.sg/specialty-areas/renal-medicine/chronic-kidney-disease-treatment/

14. Data Pre-processing Stage:

- The dataset is sparse with lot of missing values

- The dataset is not completely numerical in nature – it has nominal , numeric data type
- The field names cannot be directly understood by looking at the short hand field names identified in the dataset
- Requirement for hot encoding
- One or two more derived quantities need to be identified such as Albumin to Creatinine ratio

15. APPENDIX

- Appendix of **report**: Project Proposal

A0269637L_Project Proposal.docx

GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)

PRACTICE MODULE: Project Proposal

Date of proposal: 26 April 2023
Project Title: • **Project1 - ClimateBot** for raising awareness on Climate change , this is based on DialogFlow hosted on Telegram • **Project2 - Chronic Kidney Disease Classifier** based on Logistic Regression classification technique to classify the CKD patients whether they are CKD or CKD not
Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)* Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore NATIONAL UNIVERSITY OF SINGAPORE (NUS) Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg
Background/Aims/Objectives: Project 2: The proposed intelligent system will make use of various advanced analysis and learning techniques to train itself in a way that it can predict whether a person has CKD or CKD Not based on certain input parameters of patient.
Requirements Overview: Project2: Chronic Kidney Disease Classifier

- Research ability
  - To build the CKD classifier, team would need to understand the nature of this problem since there are multiple medical parameters that need to be understood. Also to build the predictive system need to understand what model to be used and how the pipeline to be designed to assist the medical practitioners.
- Programming ability
  - Need to know Python , NLP techniques, ML techniques
  - Need to know HTML5
  - Need to know about REST APIs
  - Need to know about Streamlit for Dashboard building
- System integration ability
  - The backend will be Python based for modeling, used for learning, analysis and reasoning task
  - Flask App for hosting the application on Web
  - Front End is HTML based and also Streamlit based App rendering for Dashboard
  - The data is restricted in this area so currently don't have access to any such database. The knowledge base used for this is a publicly available dataset that was fed for Model training and prediction.

Resource Requirements (please list Hardware, Software and any other resources)

Hardware proposed for consideration:

- I5 and i7 etc
- My system configuration is Processor Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz, 1896 Mhz, 4 Core(s), 8 Logical Processor(s).

Software proposed for consideration:

- Pertained machine learning models- Logistic Regression Binary classifier
- Deep learning tools - Python Scikit Learn
- Streamlit for building Web App

Number of Learner Interns required: (Please specify their tasks if possible)

Only 1 person

Methods and Standards:

Procedures	Objective	Key Activities
Requirement Gathering and Analysis	The team should meet with ISS to scope the details of project and ensure the achievement of business objectives.	1. Gather & Analyze Requirements 2. Define internal and External Design 3. Prioritize & Consolidate Requirements 4. Establish Functional Baseline
Technical Construction	· To develop the source code in accordance to the design.	1. Setup Development Environment 2. Understand the System Context, Design

	· To perform unit testing to ensure the quality before the components are integrated as a whole project	3. Perform Coding 4. Conduct Unit Testing
Integration Testing and acceptance testing	To ensure interface compatibility and confirm that the integrated system hardware and system software meets requirements and is ready for acceptance testing.	1. Prepare System Test Specifications 2. Prepare for Test Execution 3. Conduct System Integration Testing 4. Evaluate Testing 5. Establish Product Baseline
Acceptance Testing	To obtain ISS user acceptance that the system meets the requirements.	1. Plan for Acceptance Testing 2. Conduct Training for Acceptance Testing 3. Prepare for Acceptance Test Execution 4. ISS Evaluate Testing 5. Obtain Customer Acceptance Sign-off
Delivery	To deploy the system into production (ISS standalone server) environment.	1. Software must be packed by following ISS's standard 2. Deployment guideline must be provided in ISS production (ISS standalone server) format 3. Production (ISS standalone server) support and troubleshooting process must be defined.

Team Formation & Registration

Team Name: **Qiánzhān** **(**the idea of being at the forefront)
Project Title (repeated): • **Project2 - Chronic Kidney Disease Classifier** based on Logistic Regression classification technique to classify the CKD patients whether they are CKD or CKD not
System Name (if decided): **Project 2- Med Analytica**
Team Member 1 Name: Nilothpal Bhattacharya
Team Member 1 Matriculation Number: e1113631@u.nus.edu
Team Member 1 Contact (Mobile/Email): 83204831

For ISS Use Only		
Programme Name:	**Project No:**	**Learner Batch:**
Accepted/Rejected/KIV:		
Learners Assigned:		
Advisor Assigned: Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg		

- Appendix of **report**: Mapped System Functionalities against knowledge, techniques and skills of modular courses: MR, RS, CGS

  The NLP techniques were used to clean the data, data analysis and logistic algorithm were used for machine learning and prediction

- Appendix of **report**: Installation and User Guide
- Appendix of **report**: 1 or 2 pages **individual project report** per **project member**, including:
- **Individual reflection of project journey:**
    - (1) personal **contribution** to group project

    I have built the entire solution by myself starting from ideation till all project documentation and presentation for the project.

    - (2) what learnt is most **useful for you**

    In this semester, we were introduced to different concepts and approach in areas such as Machine Reasoning, Reasoning and Cognitive Systems to help us understand how AI systems are built at the most basic level. This exercise of project work allowed me to freely think on my ideation and implement it. It wasn't an easy journey though but the best part was that I learnt a lot in the process of building a solution all by myself and I have taken down the challenges that I encountered in entire process. Programming is not what I do in my daily work but when you get into the shoes of developer than you can realize the real challenges of building a product from scratch, starting from requirements gathering, designing the architecture and then realizing the skills needed to accomplish each of those components, I had to study separately about lot of options to build a prototype or MVP solution. I explored multiple options. I realized that backend logic building, setting up a knowledge database, the concept of new knowledge derived from an existing one, how the prediction models are built and implemented for prediction. The data analysis part wasn't so straightforward and easy. It consumes a lot of effort.

    - (3) how you can apply the knowledge and skills in **other situations or your workplaces**

    We are currently having a competition on Generative AI in our company worldwide where we have been asked to propose ideas that will be assessed to check if they are feasible. I am working to propose some use cases with my Senior Manager in Banking domain. NLP, Knowledge graph, Recommendation systems are some skills that we developed in this semester so that knowledge can be implemented to help build some useful tools in order to assist our banking clients. I will share more details once we start working on it but mainly the concept here would be to develop some kind of application with the help of which the teams can generate the architecture systems. So it has to be assessed how the existing knowledge base need to be organized and reshaped to assist the generative app building process.