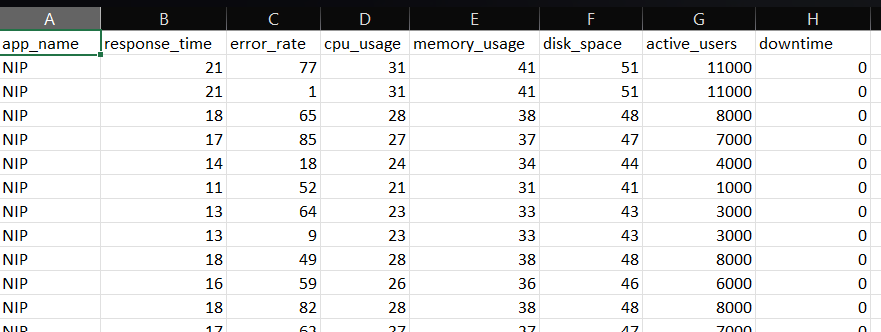
**Application Incident Prediction**

**1.DATA COLLECTION**



**About Dataset**

**A dataset to predict if an application will be down in the next 3 months**

**app\_name:** This feature represents the name of the application.

**response\_time:** It measures the time taken by the application to respond to a request or an event. It is a crucial metric for performance evaluation.

**error\_rate:** This feature indicates the rate of errors that occur in the application. It's a measure of how often errors or issues are encountered.

**cpu\_usage:** Represents the percentage of CPU (Central Processing Unit) capacity used by the application. It provides insights into the computational load.

**memory\_usage:** Indicates the amount of RAM (Random Access Memory) used by the application. It reflects the memory consumption.

**disk\_space:** Represents the amount of disk space utilized by the application. It measures the storage requirements.

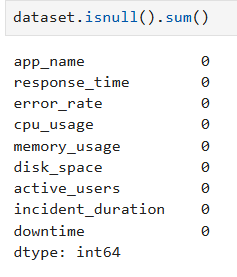
**active\_users:** This feature denotes the number of users currently interacting with the application. It reflects the user activity.

**downtime:** This is the target variable, binary in nature, indicating whether an incident resulting in downtime occurred (1) or not (0).

**timestamp:** This likely represents the timestamp or date/time associated with the data entry, providing a temporal dimension to the dataset.

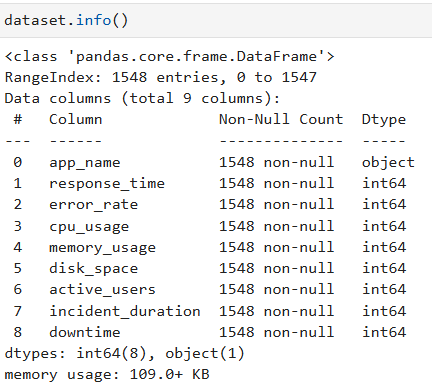
**2.DATA PREPROCESSING**

* Write the libraries and load the dataset.
* Check for the null values.



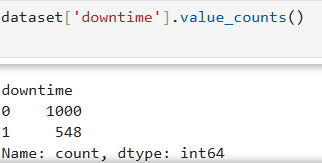
There are no null values.

* Finding the information about the dataset (dtype) object or integer that column contains.



Object means qual and int64 means quan

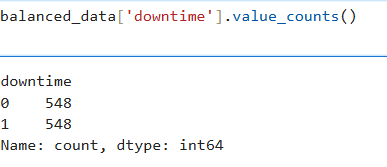
* Dataset shape for downtime



The dataset is imbalanced so we want to convert the dataset into balanced or downsampling the dataset

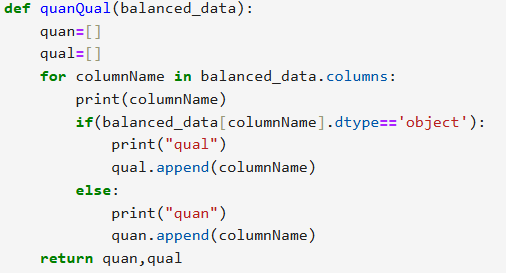
Downsampling which means we have to remove some rows to balance the dataset. It helps us to predict the correct output otherwise it will mostly choose the option to display the output as which one have the majority of samples it leads to failure of our project and its wrong prediction. In classification, we have to check the dataset balanced or imbalanced.

After downsampling the dataset,



It is now balanced

* separating the quan and qual



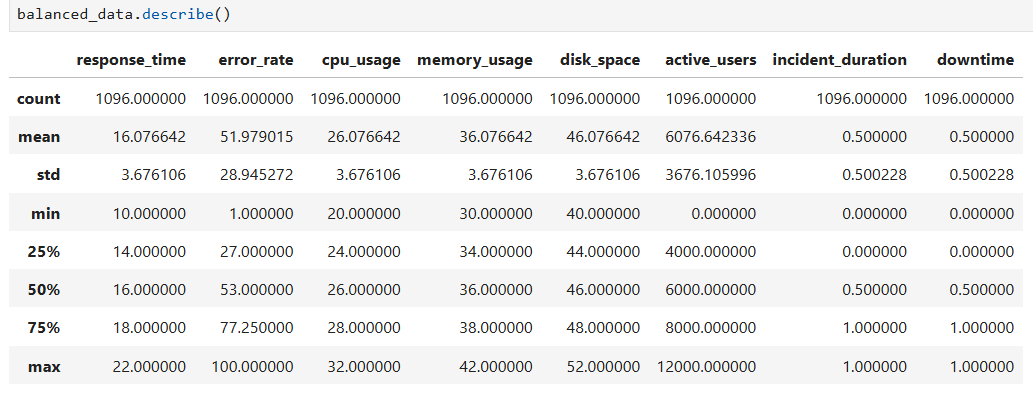
* Finding outliers and replacing them

It was very important step to data preprocessing because of the lesser and greater outlier are formed. It will make our dataset values wrong.

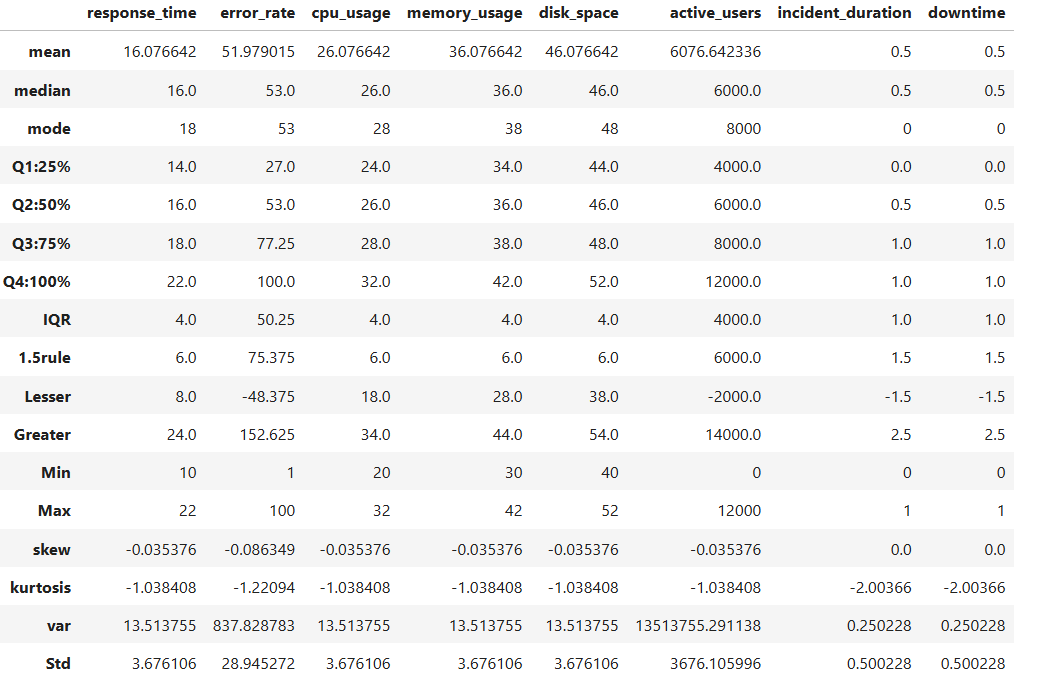
**3.UNIVARIATE AND BIVARIATE ANALYSIS**

**UNIVARIATE**

* Write the libraries and load the dataset
* Basic information of dataset by Dataframe to view



* Additionally added some univariate information like quartile, 1.5 rule, min, max, greater, lesser, skewness, kurtosis, variance, and standard deviation



**Mean:** It will calculate the average value of the all columns in the dataset

**Median:** It also the same process of mean but it can eliminate the outliers, it will take the middle value and perform the average of dividing them

**Mode:** Displaying the most repeated values from the column

**Quartile:** To divide data into four parts to understand its spread. They show how values are distributed around the median.

**IQR:** The Interquartile Range (IQR) is a measure of statistical dispersion, representing the range between the first quartile (Q1, 25th percentile) and the third quartile (Q3, 75th percentile). It is calculated as:

**IQR = Q3 - Q1**.

It shows the middle 50% of the data.

**1.5 rule:** The **1.5 IQR Rule** is used to detect outliers in a data set. An observation is considered an outlier if it is:

* Below **Q1 - 1.5 × IQR**
* Above **Q3 + 1.5 × IQR**

Where Q1 and Q3 are the first and third quartiles, and IQR is the interquartile range.

**Lesser and Greater:** The lesser and greater values identified using the 1.5 IQR rule help detect **outliers**. These values mark the boundaries:

* **Lesser value**: Q1 - 1.5 × IQR (below which data points are outliers).
* **Greater value**: Q3 + 1.5 × IQR (above which data points are outliers).

**Minimum:** It was the minimum value in the column

**Maximum:** It was the maximum value in the column

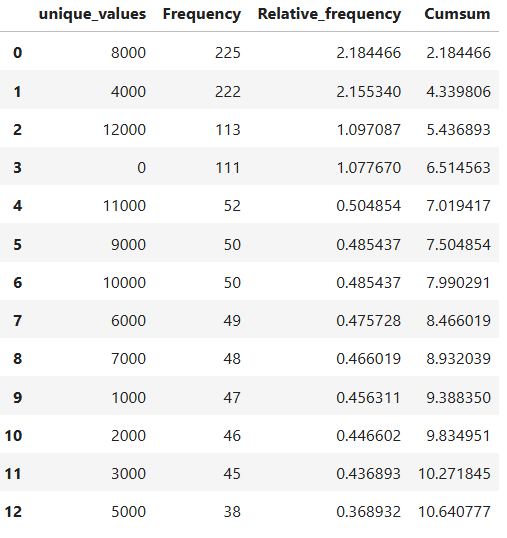
**Skewness:** It will measure the data distribution and it have two types (positive and negative)

**Kurtosis:** It will measure about the shape of the data distribution at the end

**Variance:** The displayed values from the above are calculated as the average of the squared differences between each data point and the mean from the respected column

**Standard deviation:** The square root of variance. The above values give a clearer idea of data spread in the same units from the respected column

* Frequency



**Unique\_values:-**

This column says the unique values obtained

**Frquency:-**

This column says the repeated values obtained

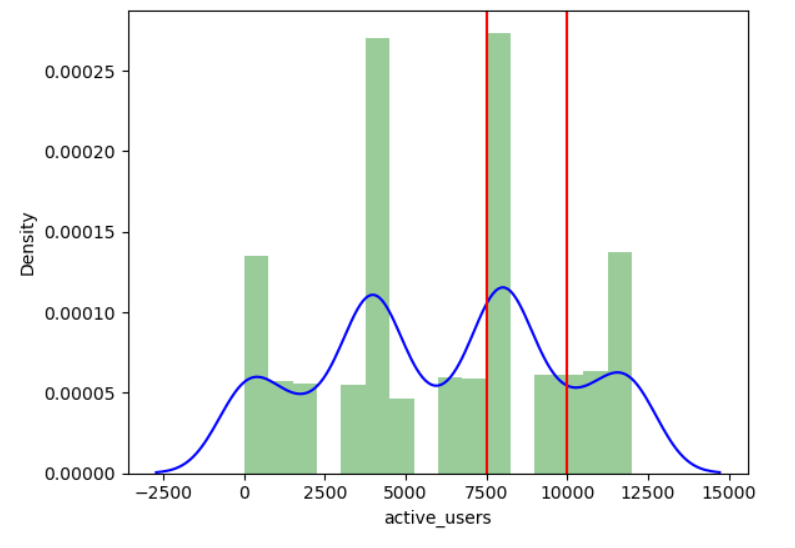
**Relative\_frequency:-**

This column says the repeated values that are in percentage (%)

**Cumsum:-**

It adds each frequency to the sum of the previous ones

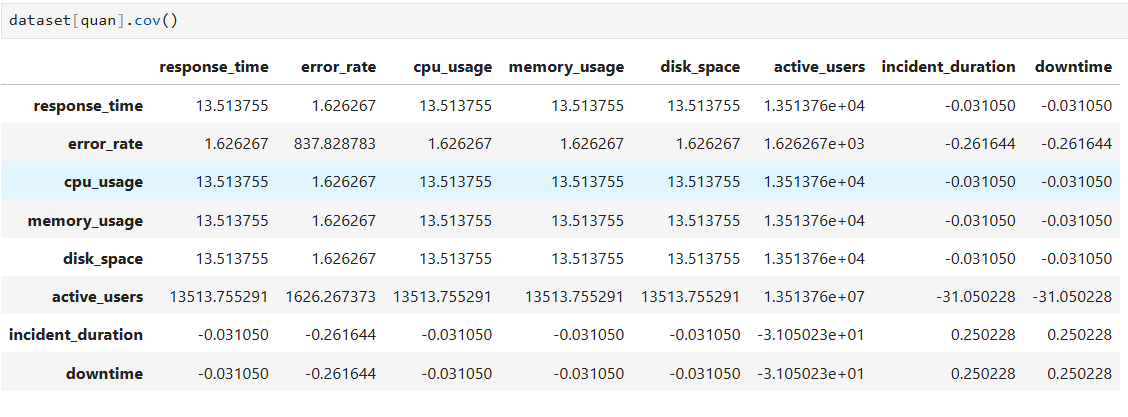
* probability density function



The red vertical line shows the probability of particular range about (mean and standard deviation) also we can calculate other calculations as we need.

**BIVARIATE**

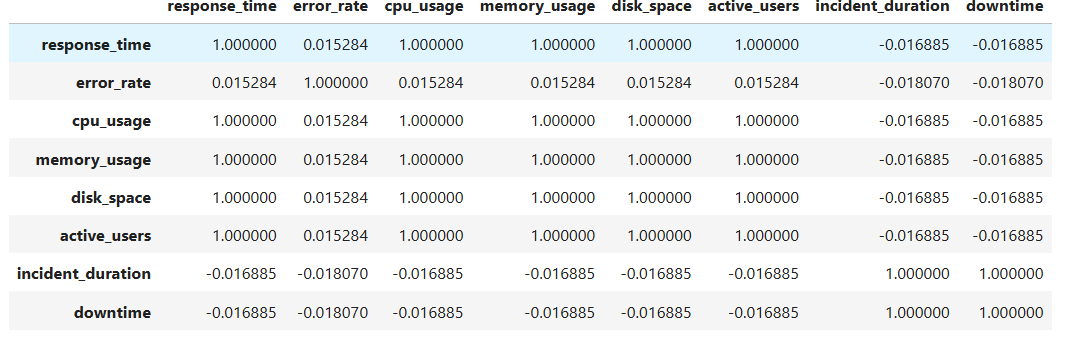
* Write the libraries and load the dataset
* Analysis for the covariance



There are three types of covariance:

1. **Positive**-Positive values
2. **Negative**-Negative values
3. **Zero** -Zero values

* Analysis for the correlation

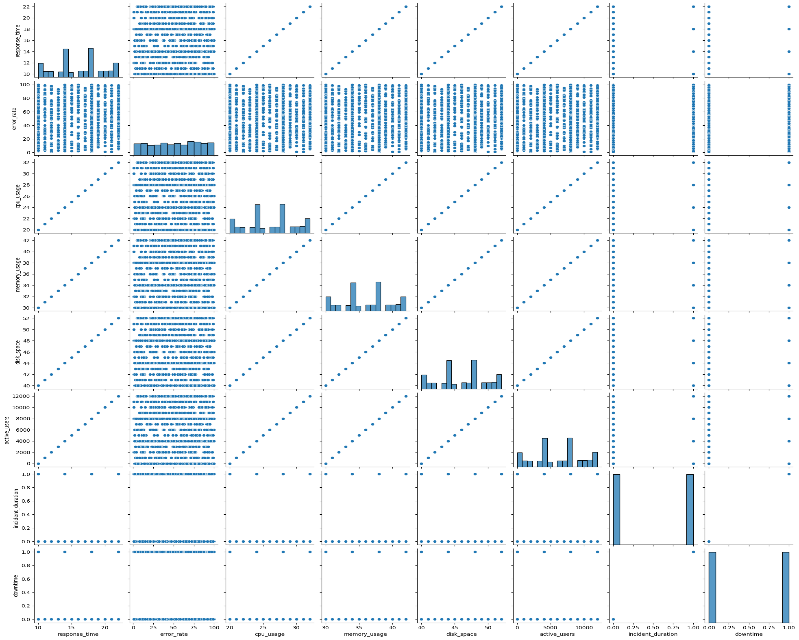


There are three types of correlation

1. **Positive**: directly proportional (The value is positive then the column value increases another column that are compared to same column that will also increase)
2. **Negative**: indirectly proportional (The value is positive then the column value decreases another column that are compared to same column that will also decrease)
3. **Zero**: Zero proportional

**Example:**

* The relation between error\_rate and downtime:
* The value is -0.018070, it is negative correlation. It is indirect propostionality
* when error\_rate increase downtime is drecrease or error\_rate decreases downtime will increase
* **SEABORN**



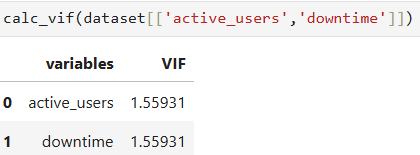
Here, the graph will show the correlation between all the columns from dataset

Incident\_duration and downtime will have the zero correlation

Error\_rate have the more correlation values

The other columns are positive correlation (the value increases the related column also increase or the value decreases the related column also decrease)

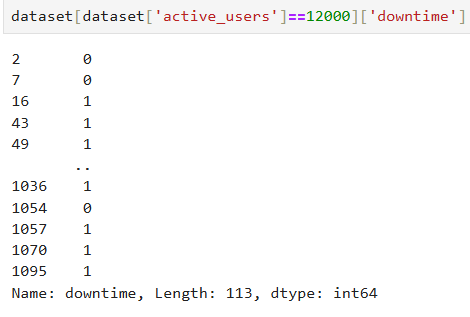
* VARIANCE INFLATOR FACTOR



VIF=1.55931

Not mostly correlated

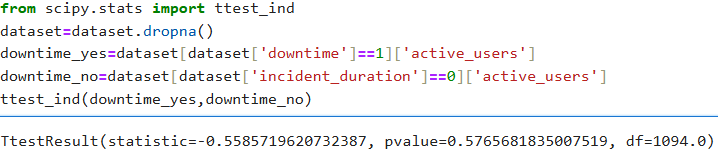
* Getting the values active\_users=12000 that respect with downtime



This will display the result as active\_users=12000 also having the downtime (0 or 1)

* Test the similarity between the downtime(1) and incident\_duration(0) with respect to active\_users at significance level of 5%.(Make decision using Hypothesis Testing)

It is independent sample (different group(0,1) but same condition(active\_users))



Null: There is no difference between downtime and incident\_duration

Alternative: There is difference between downtime and incident\_duration

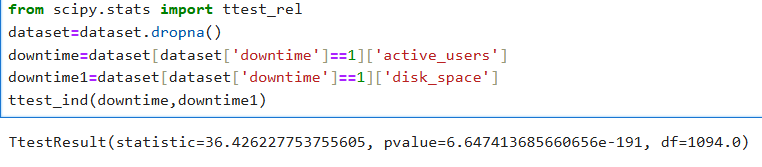
solution:

The pvalue is 0.5765681835

It will reject the Alternative statement and accept the Null statement

* Test the similarity between the downtime(1) with respect to active\_users and disk\_space at significance level of 5. (Make decision using Hypothesis Testing)

It is dependent sample (same group (downtime) but different condition (active\_users, disk\_space))



Null: There is no difference between downtime with respect to active\_users and disk\_space

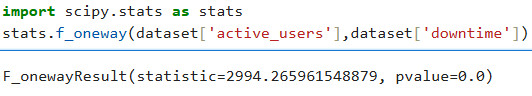
Alternative: There is difference between downtime with respect to active\_users and disk\_space

Solution:

The pvalue is 6.6474136

It will reject the Alternative statement and accept the Null statement

* ANOVA one way classification



null statement: there is no similarities between downtime and active\_users

alterative statement: there is similarities between downtime and active\_users

solution:

The pvalue is 0.

This result is mathematically valid and indicates that there is strong evidence to reject the null statement and accept Alternative statement

However, verify the ANOVA:

\*Data normality

\*Equal variances between groups

\*No outliers

* The applications are down for next 3 months

There are 548 applications are down for next 3 months. There have incident\_duration and downtime are quite similar.

REASON: If any incident were recorded there must be down of the application.

All the active\_user=0 is not to downtime\_yes (1) there are remaining rows on active\_users but not for incident\_duration

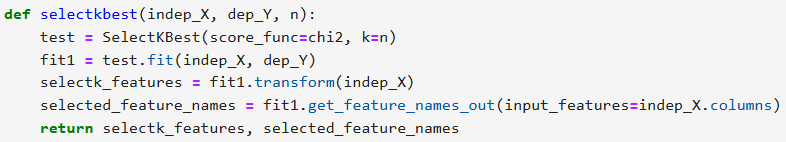
Incident\_duration is yes downtime is also be yes

**DATA VISUALIZATION**

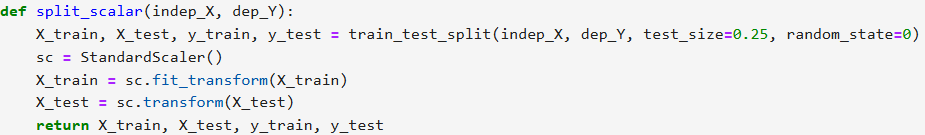
[Application incident](https://lookerstudio.google.com/u/0/reporting/b51e98df-ca47-4b65-ae8d-980bc7066842/page/ZOTaE)

**4.Feature selection and model creation**

* Write the libraries and load the dataset
* Add the function selectkbest:



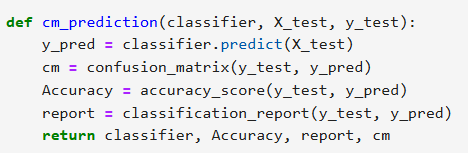
* Add the split scaler method to split for the test set and train set
* Add the standardscaler method to defined range of output



Split scaler will split the dataset to training the model and testing the model

Standardscaler will make the train and test set to defined range of values

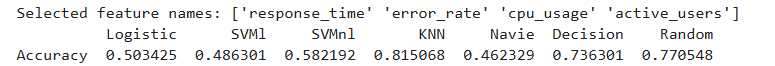
* Add the few classifications metrics



* Add algorithms to predict the applications is down or not in next 3 months

1. Logistic classification
2. Svm linear classification
3. Svm non-linear classification
4. Decision tree classification
5. Random forest classification
6. Knearest neighbour classification

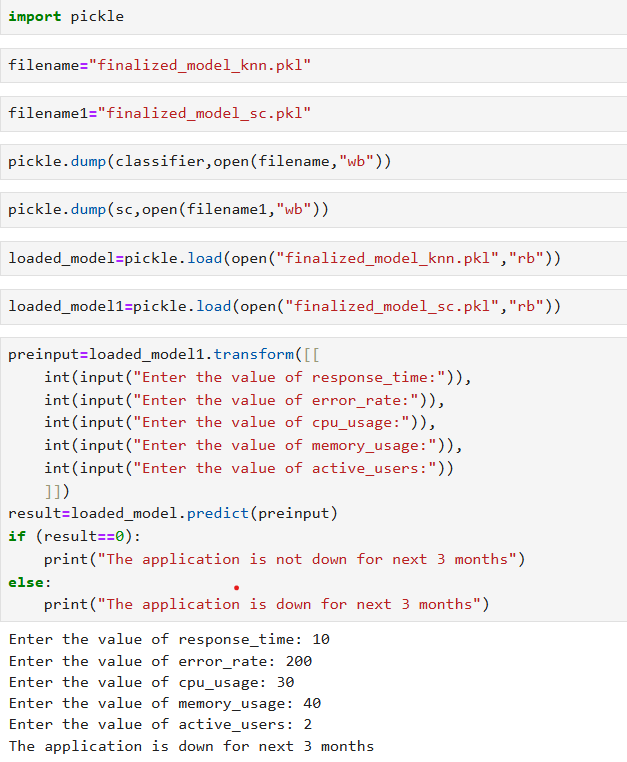
* View like dataframe to know the accuracy of all the algorithms and what the selectkbest method choosed the columns to enter the values



Here, KNN show the best accuracy so we move for the final model to save and deployment phase

**5.KNN CLASSIFICATION**

* Write the libraries and load the dataset
* Add the selectkbest method
* Add the splictscaler and standardscaler method to optimize the result
* Add the classification metrics to know the accuracy of model
* Go for Knn classification and create the model, view the accuracy
* Import pickle to save the model



* Create the new file for the deployment phase, use the saved file to redirect the function from one file to another file (using pickle function). This will help to predict for the deployment phase.

The deployment phase helps to view for users and make them understand easily. It doesn’t show any codes so users can’t face any difficulties on using them.

