**Problem Statement or Requirement:**

A requirement from the Hospital, Management asked us to create a predictive model which will predict the Chronic Kidney Disease (CKD) based on the several parameters. The Client has provided the dataset of the same:

1. Identify your problem statement
2. Tell basic info about the dataset (Total number of rows, columns)
3. Mention the pre-processing method if you’re doing any (like converting string to number – nominal data).
4. Develop a good model with good evaluation metric. You can use any machine learning algorithm; you can create many models. Finally, you have to come up with final model
5. All the research values of each algorithm should be documented. (You can make tabulation or screenshot of the results.
6. Mention your final model, justify why u have chosen the same.

Note: Mentioned points are necessary, kindly mail your document as well as .ipynb (code file) with respective name.

Sub file name also should be properly named for Example (SVM\_Ramisha\_Assi-5.ipynb)

Communication is important (How you are representing the document.)

Kindly uploaded in the Github and Share it with us.

**Identification of problem statement:**

To develop a model which will predict if Chronic Kidney disease exists or not.

**Basic information about dataset:**

This dataset has 400 rows and 25 columns. Out of which 24 columns are dependent variable and one independent variable.

**Preprocessing method:**

We will convert the ordinal data into the nominal data. Field “classification” is the dependent variable and it is changed to nominal data with Yes as 1 and No as 0.

A black text on a white background

Description automatically generated

In total the data has 249 items classified as 1 which is Yes and 150 is classified as 0 which is No.

This data set **is imbalanced dataset.**

After splitting the test and train data **training** data has **266** records and **test** data has **133** records. These 133 records has **51** classified into 0 and 82 as .

**Researching using different algorithms:**

**Logistic Regression Grid Classification**

When using the logistic regression in the grid search CV the best parameters suited are



The confusion matrix for the classification is shown below:

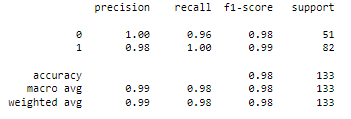


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Classification** |  | **Predicted** | |
|  | **0(No)** | **1(Yes)** |
| **Actual** | **0(No)** |  | 49 | 2 |
| **1(Yes)** |  | 0 | 82 |

In actual 51 record are classified as No but this algorithm classified only 49 items as No and 2 items as Yes.

In actual 82 records are classified as Yes which is correctly matched.

**Classification Report:**



1. Logistic regression has accuracy of 98%
2. Recall of 0 is 96%
3. Recall of 1 is 100%
4. Precision of 0 is 100%
5. Precision of 1 is 98%
6. F1 score of 0 is 98%
7. F1-Score of 1 is 99%

**This algorithm performs good.**

**Naïve Bayes Algorithm**

**GaussianNB**

Confusion\_matrix and Classification report of this algorithm is shown below:

A screenshot of a computer screen

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Classification** |  | **Predicted** | |
|  | **0(No)** | **1(Yes)** |
| **Actual** | **0(No)** |  | 51 | 0 |
| **1(Yes)** |  | 3 | 79 |

1. Naïve Bayes Gaussian NB classification has accuracy of 98%
2. Recall of 0 is 100%
3. Recall of 1 is 96%
4. Precision of 0 is 94%
5. Precision of 1 is 100%
6. F1 score of 0 is 97%
7. F1-Score of 1 is 98%

**This algorithm performs good.**

**Problem Statement or Requirement:**

A requirement from the Hospital, Management asked us to create a predictive model which will predict the Chronic Kidney Disease (CKD) based on the several parameters. The Client has provided the dataset of the same:

1. Identify your problem statement
2. Tell basic info about the dataset (Total number of rows, columns)
3. Mention the pre-processing method if you’re doing any (like converting string to number – nominal data).
4. Develop a good model with good evaluation metric. You can use any machine learning algorithm; you can create many models. Finally, you have to come up with final model
5. All the research values of each algorithm should be documented. (You can make tabulation or screenshot of the results.
6. Mention your final model, justify why u have chosen the same.

Note: Mentioned points are necessary, kindly mail your document as well as .ipynb (code file) with respective name.

Sub file name also should be properly named for Example (SVM\_Ramisha\_Assi-5.ipynb)

Communication is important (How you are representing the document.)

Kindly uploaded in the Github and Share it with us.

**Identification of problem statement:**

To develop a model which will predict if Chronic Kidney disease exists or not.

**Basic information about dataset:**

This dataset has 400 rows and 25 columns. Out of which 24 columns are dependent variable and one independent variable.

**Preprocessing method:**

We will convert the ordinal data into the nominal data. Field “classification” is the dependent variable and it is changed to nominal data with Yes as 1 and No as 0.

A black text on a white background

Description automatically generated

In total the data has 249 items classified as 1 which is Yes and 150 is classified as 0 which is No.

This data set **is imbalanced dataset.**

After splitting the test and train data **training** data has **266** records and **test** data has **133** records. These 133 records has **51** classified into 0 and 82 as .

**Researching using different algorithms:**

**Logistic Regression Grid Classification**

When using the logistic regression in the grid search CV the best parameters suited are



The confusion matrix for the classification is shown below:

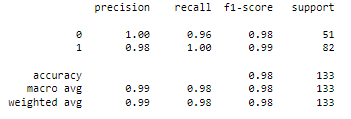


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Classification** |  | **Predicted** | |
|  | **0(No)** | **1(Yes)** |
| **Actual** | **0(No)** |  | 49 | 2 |
| **1(Yes)** |  | 0 | 82 |

In actual 51 record are classified as No but this algorithm classified only 49 items as No and 2 items as Yes.

In actual 82 records are classified as Yes which is correctly matched.

**Classification Report:**



1. Logistic regression has accuracy of 98%
2. Recall of 0 is 96%
3. Recall of 1 is 100%
4. Precision of 0 is 100%
5. Precision of 1 is 98%
6. F1 score of 0 is 98%
7. F1-Score of 1 is 99%

**This algorithm performs good.**

**Naïve Bayes Algorithm**

**GaussianNB:**

Confusion\_matrix and Classification report of this algorithm is shown below:

A screenshot of a computer screen

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Classification** |  | **Predicted** | |
|  | **0(No)** | **1(Yes)** |
| **Actual** | **0(No)** |  | 51 | 0 |
| **1(Yes)** |  | 3 | 79 |

1. Naïve Bayes Gaussian NB classification has accuracy of 98%
2. Recall of 0 is 100%
3. Recall of 1 is 96%
4. Precision of 0 is 94%
5. Precision of 1 is 100%
6. F1 score of 0 is 97%
7. F1-Score of 1 is 78%

**Naïve Bayes Algorithm**

**MultinomialNB:**

Confusion\_matrix and Classification report of this algorithm is shown below:

A screenshot of a computer screen

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Classification** |  | **Predicted** | |
|  | **0(No)** | **1(Yes)** |
| **Actual** | **0(No)** |  | 50 | 1 |
| **1(Yes)** |  | 23 | 59 |

1. it has accuracy of 82%
2. Recall of 0 is 81%
3. Recall of 1 is 83%
4. Precision of 0 is 68%
5. Precision of 1 is 98%
6. F1 score of 0 is 98%
7. F1-Score of 1 is 72%

**Naïve Bayes Algorithm**

**BernoulliNB**

Confusion\_matrix and Classification report of this algorithm is shown below:

A screenshot of a computer screen

Description automatically generated

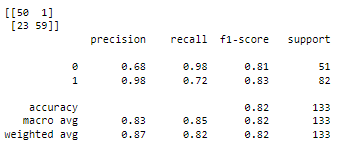
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Classification** |  | **Predicted** | |
|  | **0(No)** | **1(Yes)** |
| **Actual** | **0(No)** |  | 51 | 0 |
| **1(Yes)** |  | 8 | 74 |

1. Naïve Bayes Multinomial NB classification has accuracy of 94%
2. Recall of 0 is 100%
3. Recall of 1 is 90%
4. Precision of 0 is 86%
5. Precision of 1 is 100%
6. F1 score of 0 is 93%
7. F1-Score of 1 is 95%

**Naïve Bayes Algorithm**

**ComplementNB**

Confusion\_matrix and Classification report of this algorithm is shown below:

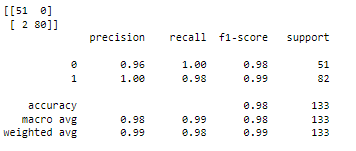


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Classification** |  | **Predicted** | |
|  | **0(No)** | **1(Yes)** |
| **Actual** | **0(No)** |  | 50 | 1 |
| **1(Yes)** |  | 23 | 59 |

1. Naïve Bayes Complement NB classification has accuracy of 82%
2. Recall of 0 is 98%
3. Recall of 1 is 72%
4. Precision of 0 is 68%
5. Precision of 1 is 98%
6. F1 score of 0 is 81%
7. F1-Score of 1 is 83%

**Decision Tree Algorithm**

Confusion\_matrix and Classification report of this algorithm is shown below:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Classification** |  | **Predicted** | |
|  | **0(No)** | **1(Yes)** |
| **Actual** | **0(No)** |  | 51 | 0 |
| **1(Yes)** |  | 2 | 80 |

1. It has accuracy of 98%
2. Recall of 0 is 100%
3. Recall of 1 is 98%
4. Precision of 0 is 96%
5. Precision of 1 is 100%
6. F1 score of 0 is 98%
7. F1-Score of 1 is 99%

**Random Forest Algorithm**

Confusion\_matrix and Classification report of this algorithm is shown below:

A number of numbers on a white background

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Classification** |  | **Predicted** | |
|  | **0(No)** | **1(Yes)** |
| **Actual** | **0(No)** |  | 51 | 0 |
| **1(Yes)** |  | 6 | 76 |

1. It has accuracy of 95%
2. Recall of 0 is 100%
3. Recall of 1 is 93%
4. Precision of 0 is 89%
5. Precision of 1 is 100%
6. F1 score of 0 is 94%
7. F1-Score of 1 is 96%

**Conclusion:**

By analysing the algorithms we could see that Logistic Grid Classification, Decision Tree Algorithm, Naïve Bayes – Gauissian model gives the better results.