

Hope Artificial Intelligence



Classification Assignment

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

Develop a predictive model to identify Chronic Kidney Disease (CKD) based on various parameters provided in the dataset.

2.) Tell basic info about the dataset (Total number of rows, columns)

399 rows × 25 columns

3.) Mention the pre-processing method if you're doing any (like converting string to number – nominal data)

get_dummies -To convert string to number

StandardScaler-Standardization rescales features to have a mean of 0 and a standard deviation of 1,

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.)

Decision tree results

```
In [16]: print("The confusion Matrix:\n",cm)
```

```
The confusion Matrix:
[[49  2]
 [ 4 78]]
```

```
In [17]: print("The report:\n",clf_report)
```

```
The report:
              precision    recall  f1-score   support

     0       0.92      0.96      0.94         51
     1       0.97      0.95      0.96         82

 accuracy          0.95
 macro avg          0.95
 weighted avg       0.96
```

Random Forest Results

```
In [28]: print("The confusion Matrix:\n",cm)
```

```
The confusion Matrix:  
[[50  1]  
 [ 0 82]]
```

```
In [29]: print("The report:\n",clf_report)
```

```
The report:
```

	precision	recall	f1-score	support
0	1.00	0.98	0.99	51
1	0.99	1.00	0.99	82
accuracy			0.99	133
macro avg	0.99	0.99	0.99	133
weighted avg	0.99	0.99	0.99	133

SVM Results

```
In [17]: print("The confusion Matrix:\n",cm)
```

```
The confusion Matrix:  
[[51  0]  
 [ 1 81]]
```

```
In [18]: print("The report:\n",clf_report)
```

```
The report:
```

	precision	recall	f1-score	support
0	0.98	1.00	0.99	51
1	1.00	0.99	0.99	82
accuracy			0.99	133
macro avg	0.99	0.99	0.99	133
weighted avg	0.99	0.99	0.99	133

Logistic Regression_results

```
In [18]: print("The confusion Matrix:\n",cm)
```

```
The confusion Matrix:  
[[51  0]  
 [ 1 81]]
```

```
In [19]: print("The report:\n",clf_report)
```

```
The report:
```

	precision	recall	f1-score	support
0	0.98	1.00	0.99	51
1	1.00	0.99	0.99	82
accuracy			0.99	133
macro avg	0.99	0.99	0.99	133
weighted avg	0.99	0.99	0.99	133

KNN Results

Error

BernoulliNB_results

```
In [14]: from sklearn.naive_bayes import BernoulliNB
classifier = BernoulliNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

	precision	recall	f1-score	support
0	0.86	1.00	0.93	51
1	1.00	0.90	0.95	82
accuracy			0.94	133
macro avg	0.93	0.95	0.94	133
weighted avg	0.95	0.94	0.94	133

```
[[51  0]
 [ 8 74]]
```

categorical NB_results

```
In [15]: from sklearn.naive_bayes import CategoricalNB
classifier = CategoricalNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

```
File ~\anaconda3\Lib\site-packages\sklearn\naive_bayes.py:102, in _BaseNB.predict(self, X)
    100 check_is_fitted(self)
    101 X = self._check_X(X)
--> 102 jll = self._joint_log_likelihood(X)
    103 return self.classes_[np.argmax(jll, axis=1)]

File ~\anaconda3\Lib\site-packages\sklearn\naive_bayes.py:1526, in CategoricalNB._joint_log_likelihood(self, X)
    1524 for i in range(self.n_features_in_):
    1525     indices = X[:, i]
-> 1526     jll += self.feature_log_prob_[i][:, indices].T
    1527 total_ll = jll + self.class_log_prior_
    1528 return total_ll
```

IndexError: index 90 is out of bounds for axis 1 with size 84

ComplementNB_Results

```
In [16]: from sklearn.naive_bayes import ComplementNB
classifier = ComplementNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

	precision	recall	f1-score	support
0	0.68	0.98	0.81	51
1	0.98	0.72	0.83	82
accuracy			0.82	133
macro avg	0.83	0.85	0.82	133
weighted avg	0.87	0.82	0.82	133


```
[[50  1]
 [23 59]]
```

MultinomialNB_results

```
In [13]: from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

	precision	recall	f1-score	support
0	0.68	0.98	0.81	51
1	0.98	0.72	0.83	82
accuracy			0.82	133
macro avg	0.83	0.85	0.82	133
weighted avg	0.87	0.82	0.82	133


```
[[50  1]
 [23 59]]
```

6.) Mention your final model, justify why u have chosen the same.

SVM or Logistic Regression are suitable options due to their low Type I error rates and higher accuracy compared to other models. Additionally, these models demonstrate higher precision and recall values, attributed to their training on unbalanced data.

Best parameter for SVM

The f1_macro value for best parameter {'C': 10, 'gamma': 'auto', 'kernel': 'sigmoid'}: 0.9924946382275899

Best parameter of Logistic Regression

The f1_macro value for best parameter {'penalty': 'l2', 'solver': 'newton-cg'}: 0.9924946382275899

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