

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

Ans:

Problem Statement is Domain Selection is Machine Learning-> Supervised Learning -> Classification

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

Ans:

Basic Info Dataset is Total Number of Rows: 399
Total Number of Columns : 25

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

Ans:

Yes we need to convert string to number using one hot Encoding Method

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.)

Ans:

Grid Search CV

LogisticRegression

```
In [29]: #Learning Model Creation
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

In [26]: from sklearn.linear_model import LogisticRegression

In [30]: from sklearn.model_selection import GridSearchCV

param_grid = {'solver':['newton-cg', 'lbfgs', 'liblinear', 'saga'],
              'penalty':['l2']}
```

```

grid = GridSearchCV(LogisticRegression(), param_grid, refit = True, verbose = 3,n_jobs=-1,scoring='f1_weighted')

# fitting the model for grid search
grid.fit(X_train, y_train)

```

In [33]: `print("The confusion Matrix:\n",cm)`

```

The confusion Matrix:
[[80  2]
 [ 0 51]]

```

In [34]: `print("The report:\n",clf_report)`

```

The report:

```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 1.00 | 0.98 | 0.99 | 82 |
| 1 | 0.96 | 1.00 | 0.98 | 51 |
| accuracy | | | 0.98 | 133 |
| macro avg | 0.98 | 0.99 | 0.98 | 133 |
| weighted avg | 0.99 | 0.98 | 0.99 | 133 |

SVC

In [35]: `from sklearn.svm import SVC`

```

In [36]: from sklearn.model_selection import GridSearchCV

param_grid = {'kernel':['linear','rbf','poly','sigmoid'],
              'gamma':['auto','scale'],
              'C':[10,100,1000,2000,3000]}

grid = GridSearchCV(SVC(), param_grid, refit = True, verbose = 3,n_jobs=-1,scoring='f1_weighted')

```

In [39]: `print("The confusion Matrix for SVC :\n",cm)`

```

The confusion Matrix for SVC :
[[79  3]
 [ 0 51]]

```

In [40]: `print("The report for SVC:\n",clf_report)`

```

The report for SVC:

```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 1.00 | 0.96 | 0.98 | 82 |
| 1 | 0.94 | 1.00 | 0.97 | 51 |
| accuracy | | | 0.98 | 133 |
| macro avg | 0.97 | 0.98 | 0.98 | 133 |
| weighted avg | 0.98 | 0.98 | 0.98 | 133 |

Decision Tree Classifier

```

In [42]: from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier
param_grid = {'criterion':['gini','entropy'],
              'max_features': ['auto','sqrt','log2'],
              'splitter':['best','random']}

|

grid = GridSearchCV(DecisionTreeClassifier(), param_grid, refit = True, verbose = 3,n_jobs=-1,scoring='f1_weighted')

# fitting the model for grid search
grid.fit(X_train, y_train)

Fitting 5 folds for each of 12 candidates, totalling 60 fits

```

```
the f1_macro value for best parameter { criterion : gini , max_features : sqrt , splitter : random } : 0.9105489850/1/
```

```
In [45]: print("The confusion Matrix of Decision Tree Classifier:\n",cm)
```

```
The confusion Matrix of Decision Tree Classifier:  
[[74  8]  
 [ 4 47]]
```

```
In [46]: print("The report for Decision Tree Classifier:\n",clf_report)
```

```
The report for Decision Tree Classifier:  
      precision    recall  f1-score   support  
  
    0       0.95      0.90      0.92         82  
    1       0.85      0.92      0.89         51  
  
 accuracy          0.91         0.91         0.91        133  
  macro avg       0.90      0.91      0.91         133  
 weighted avg     0.91      0.91      0.91         133
```

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

Ans:

Final Model is **SVC because accuracy is 0.98**

F1_score: 0.98

For No(0): 0.98

For Yes(1):0.97