

NEURAL NETWORKS CLASSIFIER CODE

GitHub Repo: [Code Link](#)

Snippet-1: Imports the necessary libraries and functions

```
import numpy as np
from sklearn.svm import SVC.....
```

Snippet-2: Fetches the Breast Cancer Dataset from Uv Irvine website and divides the variables into target(y) and predictor(X) variables.

```
from ucimlrepo import fetch_ucirepo
# fetch dataset
breast_cancer_wisconsin_diagnostic = fetch_ucirepo(id=17)
# data (as pandas dataframes)
X = breast_cancer_wisconsin_diagnostic.data.features
y = breast_cancer_wisconsin_diagnostic.data.targets
```

Snippet-3: Creates histograms and boxplots of the feature variables for skewness and outliers in the data.

```
#Histograms of the predictor variables
X.hist(bins = 30, figsize=(20,15), color='red')
plt.suptitle('Histograms of the Predictors')
plt.show()
#Boxplots for outliers
plt.figure(figsize=(20,15))
sns.boxplot(data=X, orient='h')
plt.title('Boxplots of the Predictors')
plt.show()
```

Snippet-4: The following snippet splits the features and target variables into training and testing sets with an 80/20 ratio. The function `train_and_test_models` accepts 7 parameters and incorporates a pipeline that applies Yeo-Johnson and spatial transformations to reduce skewness and eliminate outliers. Afterward, it builds a model and generates predictions on the test set. To evaluate model performance, it uses accuracy, F1 score, precision, and recall.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=96)
def train_and_test_models(name, model, tuning_grid, pred_train, pred_test, tgt_train,
tgt_test):
    model_pipeline = Pipeline([
        ('yeo_johnson', PowerTransformer(method='yeo-johnson')),
        ('spatial', Normalizer(norm='l2')),
        (name, model)])
    model_grid_search = GridSearchCV(model_pipeline, tuning_grid, cv=scoring='accuracy', n_jobs=-1)
    model_grid_search.fit(pred_train, tgt_train)
    model_predictions = model_grid_search.predict(pred_test)
    display(f"{name} results")
    display("The accuracy is:", accuracy_score(tgt_test, model_predictions))
    display("The Precision is:", precision_score(tgt_test, model_predictions, pos_label='M'))
    display("The Recall is:", recall_score(tgt_test, model_predictions, pos_label='M'))
    display("The F1 Score is:", f1_score(tgt_test, model_predictions, pos_label='M'))
```

Snippet-5: `nn_param_grid` is the tuning grid for Neural Networks model and is used in the function `train_and_test_models` to train and hypertune the Multi Layer Neural Network model.

```
nn_param_grid = {
    'nn_model__hidden_layer_sizes': [(10,), (20,), (30,)],
    'nn_model__activation': ['logistic', 'tanh', 'relu'],
    'nn_model__learning_rate': ['constant', 'adaptive', 'invscaling'],
}

train_and_test_models("nn_model",MLPClassifier(), nn_param_grid, X_train,
X_test, y_train, y_test)
```