7-8-25

Page
Date

LAB-3

Study of the Classifiers with respect to Statistical Parameters

Aim: To Study and evaluate the performance of a Classifier using a Statistical Parameters ~~Confusion, Precision, Recall, F1 Score~~ Accuracy

Objective:

- To train a Logistic Regression classifier on the Iris dataset
- To evaluate the classifier using Statistical metrics.
- To understand the significance of Precision, recall F1 Score, and confusion matrix in classification tasks.

Pseudocode:

1. Import required libraries and modules.
2. Load Iris dataset
3. Split the dataset into training and testing sets.
4. Initialize and train the logistic regression model on the training data, Decision tree, SVM
5. Predict the target values using the trained model on the test set.
6. Evaluate using confusion matrix.
7. print Results

Page
Date

Observation:

- Logistic Regression classified Iris-Setosa perfectly with minor misclassifications b/w Versicolor and Virginica.
- ~~Precision, Recall and F1 Score were high (>0.9) showing strong performance.~~
- SVM achieved near perfect classification which is a strong fit.

Result:

- The Model achived 95-97% accuracy.
- Succesfully Implemented classifier with respect to statistical parameters

Observation Table:

Classifier	Accuracy	Notes
Decision Tree	84.72%	Fast but slightly overfits lowest generalization.
SVM	98.61%	High performance & Performs best.
Logistic Regression	97.5%	Very high accuracy good generalization

- Decision Tree shows lowest performance :)
- SVM achieved near perfect classification Strong fit for database.
- LR also very accurate and Competitive with SVM.

```
[notice] A new release of pip is available: 25.0.1 -> 25.2
[notice] To update, run: pip install --upgrade pip
```

[13]:
```python
import numpy as np
import scipy
import seaborn as sns
import matplotlib.pyplot as plt

print("Libraries are working correctly.")
```

```
Libraries are working correctly.
```

[14]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
```

the appropriate compiler flags.

[15]:
```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'  # 0 = all logs, 3 = only fatal errors
import tensorflow as tf
```

[17]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
```

[1]:
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

[2]: