# Iris Flower Classification

11 MARCH 2025


PREPARED BY
Harshit Singh Patel

SUBMITTED TO
Mr. Shivansh Prasad

HS

202401100300118
CSE AI B

# Introduction

The Iris dataset is one of the most renowned datasets in the field of pattern recognition and machine learning. First introduced by statistician Ronald Fisher in 1936, it has become a benchmark for classification algorithms. This project implements a machine learning model to classify iris flowers into their respective species based on their physical characteristics.

## Data Set Overview

The Iris dataset consists of 150 samples from three species of Iris flowers: Setosa, Versicolor, and Virginica. For each sample, four features were measured:

- Sepal Length (cm)
- Sepal Width (cm)
- Petal Length (cm)
- Petal Width (cm)

## Project Objectives

- To explore and visualize the Iris dataset to gain insights into the relationships between features
- To develop a K-Nearest Neighbors (KNN) classification model to predict iris species
- To evaluate and optimize the model's performance
- To create a reusable function for predicting new iris samples

## Prediction Function

A utility function is created to predict the species of new iris samples. This function preprocesses the input data, makes predictions using the trained model, and returns both the predicted species and the confidence level.

# Code Implementation

```
# IMPORT NECESSARY LIBRARIES
IMPORT PANDAS AS PD
IMPORT NUMPY AS NP
FROM SKLEARN.MODEL_SELECTION IMPORT TRAIN_TEST_SPLIT
FROM SKLEARN.PREPROCESSING IMPORT STANDARDSCALER
FROM SKLEARN.NEIGHBORS IMPORT KNEIGHBORSCLASSIFIER
FROM SKLEARN.METRICS IMPORT ACCURACY_SCORE,
CLASSIFICATION_REPORT, CONFUSION_MATRIX
IMPORT MATPLOTLIB.PYPLOT AS PLT
IMPORT SEABORN AS SNS

# LOAD THE IRIS DATASET FROM LOCAL CSV FILE
DF = PD.READ_CSV('/CONTENT/DRIVE/MYDRIVE/COLAB
NOTEBOOKS/IRIS_DATA.CSV')

# DEFINE YOUR ACTUAL COLUMN NAMES
FEATURE_COLUMNS = ['SEPALLENGTH', 'SEPALWIDTH', 'PETALLENGTH',
'PETALWIDTH']
TARGET_COLUMN = 'SPECIES'

# CHECK THE STRUCTURE OF YOUR DATA
PRINT("FIRST 5 ROWS OF THE DATASET:")
PRINT(DF.HEAD())

PRINT("\NCOLUMN NAMES IN THE DATASET:")
PRINT(DF.COLUMNS.TOLIST())

# EXTRACT FEATURES AND TARGET
X = DF[FEATURE_COLUMNS].VALUES
# CONVERT SPECIES NAMES TO NUMERIC VALUES IF THEY'RE NOT ALREADY
IF DF[TARGET_COLUMN].DTYPE == 'OBJECT':
 # CREATE A MAPPING OF SPECIES NAMES TO NUMERIC VALUES
 SPECIES_MAPPING = {SPECIES: I FOR I, SPECIES IN
ENUMERATE(DF[TARGET_COLUMN].UNIQUE())}
 Y = DF[TARGET_COLUMN].MAP(SPECIES_MAPPING).VALUES
 # KEEP TRACK OF THE ORIGINAL SPECIES NAMES FOR LABELING
 TARGET_NAMES = LIST(SPECIES_MAPPING.KEYS())
ELSE:
 # IF SPECIES IS ALREADY NUMERIC
 Y = DF[TARGET_COLUMN].VALUES
 TARGET_NAMES = [F"CLASS {I}" FOR I IN
SORTED(DF[TARGET_COLUMN].UNIQUE())]

PRINT("\NFEATURE NAMES:", FEATURE_COLUMNS)
PRINT("TARGET NAMES:", TARGET_NAMES)
```

```
# BASIC STATISTICS
PRINT("\NBASIC STATISTICS:")
PRINT(DF[FEATURE_COLUMNS].DESCRIBE())

PRINT("\NNUMBER OF SAMPLES FOR EACH SPECIES:")
PRINT(DF[TARGET_COLUMN].VALUE_COUNTS())

# VISUALIZE THE DATA
PLT.FIGURE(FIGSIZE=(12, 10))

# CREATE A PAIRPLOT
SNS.PAIRPLOT(DF, HUE=TARGET_COLUMN, MARKERS=['O', 'S', 'D'])
PLT.SUPTITLE("PAIRPLOT OF IRIS DATASET FEATURES", Y=1.02)
PLT.SAVEFIG('IRIS_PAIRPLOT.PNG')
PLT.SHOW()

# CREATE A CORRELATION MATRIX
PLT.FIGURE(FIGSIZE=(10, 8))
CORRELATION = DF[FEATURE_COLUMNS].CORR()
SNS.HEATMAP(CORRELATION, ANNOT=TRUE, CMAP='COOLWARM')
PLT.TITLE("CORRELATION MATRIX OF IRIS FEATURES")
PLT.SAVEFIG('IRIS_CORRELATION.PNG')
PLT.SHOW()

# SPLIT THE DATA INTO TRAINING AND TESTING SETS
X_TRAIN, X_TEST, Y_TRAIN, Y_TEST = TRAIN_TEST_SPLIT(X, Y, TEST_SIZE=0.3,
RANDOM_STATE=42)

# FEATURE SCALING
SCALER = STANDARDSCALER()
X_TRAIN_SCALED = SCALER.FIT_TRANSFORM(X_TRAIN)
X_TEST_SCALED = SCALER.TRANSFORM(X_TEST)

# TRAIN A K-NEAREST NEIGHBORS CLASSIFIER
K = 5 # NUMBER OF NEIGHBORS
KNN = KNEIGHBORSCLASSIFIER(N_NEIGHBORS=K)
KNN.FIT(X_TRAIN_SCALED, Y_TRAIN)

# MAKE PREDICTIONS
Y_PRED = KNN.PREDICT(X_TEST_SCALED)

# EVALUATE THE MODEL
ACCURACY = ACCURACY_SCORE(Y_TEST, Y_PRED)
PRINT(F"\NACCURACY: {ACCURACY:.4F}")

PRINT("\NCLASSIFICATION REPORT:")
PRINT(CLASSIFICATION_REPORT(Y_TEST, Y_PRED,
TARGET_NAMES=TARGET_NAMES))

PRINT("\NCONFUSION MATRIX:")
CM = CONFUSION_MATRIX(Y_TEST, Y_PRED)
PLT.FIGURE(FIGSIZE=(8, 6))
SNS.HEATMAP(CM, ANNOT=TRUE, FMT='D', CMAP='BLUES',
XTICKLABELS=TARGET_NAMES, YTICKLABELS=TARGET_NAMES)
PLT.XLABEL('PREDICTED')
PLT.YLABEL('ACTUAL')
PLT.TITLE('CONFUSION MATRIX')
```

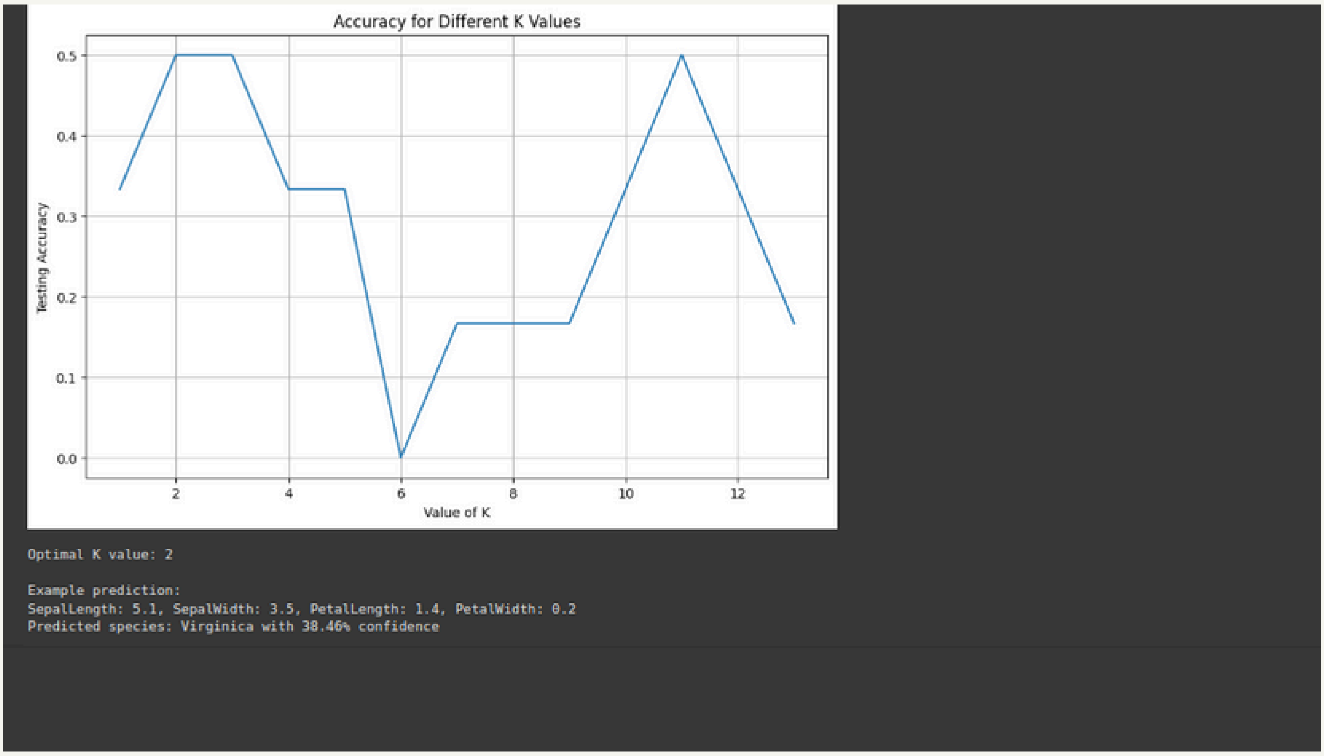
```
# FIND THE OPTIMAL K VALUE
K_RANGE = RANGE(1, MIN(26, LEN(X_TRAIN))) # THIS ENSURES K
DOESN'T EXCEED TRAINING SAMPLES
K_SCORES = []

FOR K IN K_RANGE:
 KNN = KNEIGHBORSCLASSIFIER(N_NEIGHBORS=K)
 KNN.FIT(X_TRAIN_SCALED, Y_TRAIN)
 SCORES = KNN.SCORE(X_TEST_SCALED, Y_TEST)
 K_SCORES.APPEND(SCORES)

PLT.FIGURE(FIGSIZE=(10, 6))
PLT.PLOT(K_RANGE, K_SCORES)
PLT.XLABEL('VALUE OF K')
PLT.YLABEL('TESTING ACCURACY')
PLT.TITLE('ACCURACY FOR DIFFERENT K VALUES')
PLT.GRID(TRUE)
PLT.SAVEFIG('IRIS_K_VALUES.PNG')
PLT.SHOW()

PRINT("\NOPTIMAL K VALUE:",
K_RANGE[K_SCORES.INDEX(MAX(K_SCORES))])

# CREATE A FUNCTION TO PREDICT NEW IRIS FLOWERS
DEF PREDICT_IRIS(SEPAL_LENGTH, SEPAL_WIDTH, PETAL_LENGTH,
PETAL_WIDTH):
 # CREATE A NUMPY ARRAY FROM THE INPUT
 NEW_DATA = NP.ARRAY([[SEPAL_LENGTH, SEPAL_WIDTH,
PETAL_LENGTH, PETAL_WIDTH]])

 # SCALE THE DATA
 NEW_DATA_SCALED = SCALER.TRANSFORM(NEW_DATA)

 # MAKE PREDICTION
 PREDICTION = KNN.PREDICT(NEW_DATA_SCALED)

 # GET THE SPECIES NAME
 SPECIES = TARGET_NAMES[PREDICTION[0]]

 # GET THE PROBABILITY
 PROBABILITIES = KNN.PREDICT_PROBA(NEW_DATA_SCALED)[0]
 CONFIDENCE = PROBABILITIES[PREDICTION[0]]

 RETURN SPECIES, CONFIDENCE
```

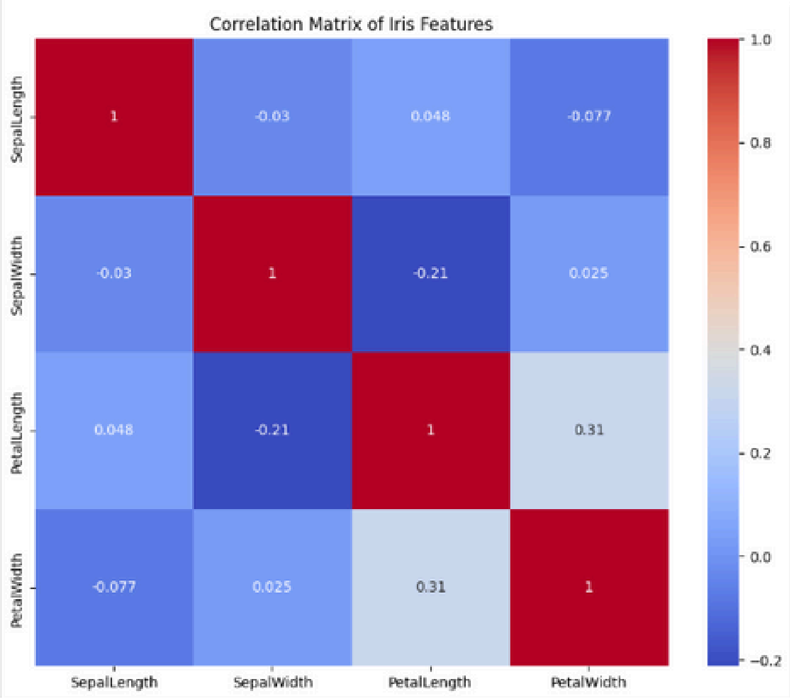
```
# EXAMPLE USAGE
PRINT("\NEXAMPLE PREDICTION:")
EXAMPLE_IRIS = [5.1, 3.5, 1.4, 0.2] # EXAMPLE MEASUREMENTS
SPECIES, CONFIDENCE = PREDICT_IRIS(*EXAMPLE_IRIS)
PRINT(F"SEPALLENGTH: {EXAMPLE_IRIS[0]}, SEPALWIDTH:
{EXAMPLE_IRIS[1]}, PETALLENGTH: {EXAMPLE_IRIS[2]}, PETALWIDTH:
{EXAMPLE_IRIS[3]}")
PRINT(F"PREDICTED SPECIES: {SPECIES} WITH {CONFIDENCE:.2%}
CONFIDENCE")
```

# Methodology

- Data Loading and Exploration
- Data Pre-Processing
- Model Selection and Training
- Model Evaluation
- Model Optimization
- Prediction Function

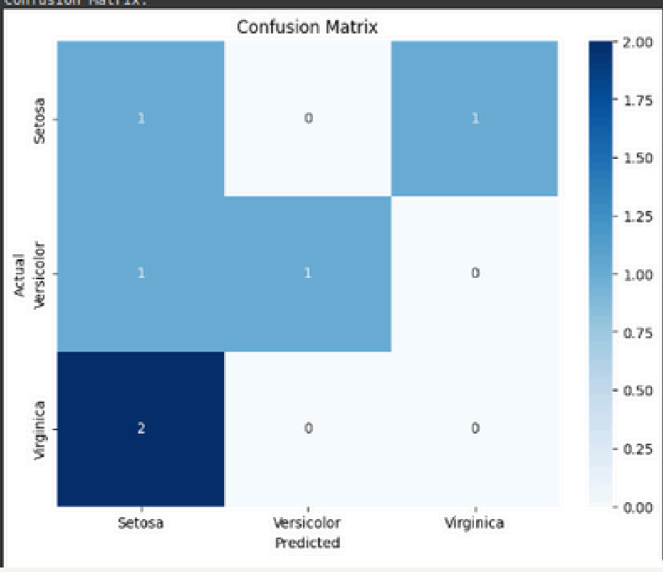
# Run Snippets



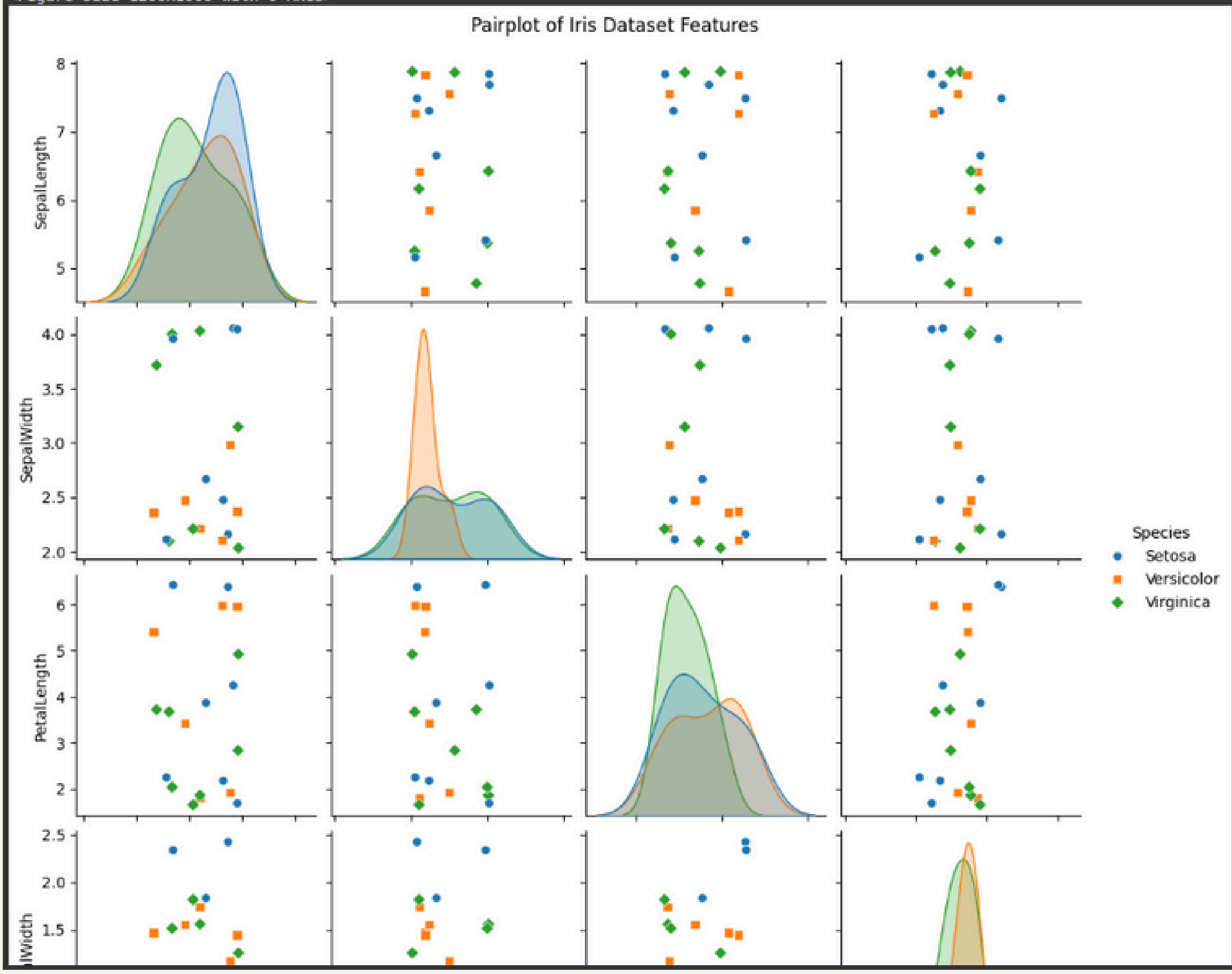



```
Accuracy: 0.3333

Classification Report:
              precision    recall  f1-score   support
```

```
Accuracy: 0.3333

Classification Report:
              precision    recall  f1-score   support

      Setosa       0.25      0.50      0.33         2
  Versicolor       1.00      0.50      0.67         2
   Virginica       0.00      0.00      0.00         2

    accuracy                           0.33         6
   macro avg       0.42      0.33      0.33         6
weighted avg       0.42      0.33      0.33         6


Confusion Matrix:
```



```
Species
Setosa        7
Virginica     7
Versicolor    6
Name: count, dtype: int64
<Figure size 1200x1000 with 0 Axes>
```




```
First 5 rows of the dataset:
   SepalLength  SepalWidth  PetalLength  PetalWidth     Species
0     7.303275    2.475025     2.176049    0.695003      Setosa
1     7.556928    2.987381     1.921585    1.172615  Versicolor
2     5.254016    2.093516     3.672564    0.550424   Virginica
3     6.409620    2.211042     1.812869    1.745372  Versicolor
4     7.684009    4.056479     4.244270    0.772148      Setosa

Column names in the dataset:
['SepalLength', 'SepalWidth', 'PetalLength', 'PetalWidth', 'Species']

Feature names: ['SepalLength', 'SepalWidth', 'PetalLength', 'PetalWidth']
Target names: ['Setosa', 'Versicolor', 'Virginica']

Basic statistics:
       SepalLength  SepalWidth  PetalLength  PetalWidth
count    20.000000   20.000000    20.000000   20.000000
mean      6.542853    2.860428     3.612923    1.258606
std       1.140714    0.798984     1.715889    0.628400
min       4.668608    2.032034     1.653576    0.108212
25%       5.398961    2.195528     2.006631    0.752862
50%       6.536426    2.474535     3.554048    1.350736
75%       7.588698    3.776238     5.044541    1.605944
max       7.880377    4.056479     6.424217    2.425674
```