# DWDM Tutorial 5

U18CO021: SAHIL BONDRE

**5. Experiment with any dataset of your choice using a decision tree model, logistic regression, naïve Bayes classifier, and a support vector model. Make a table of your results. Are the differences in model performance significant?**

## 1 Import Libraries

[1]:
```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

## 2 Load Data

[2]:
```
# IRIS Dataset
X, y = load_iris(return_X_y=True)
print(X.shape, y.shape)
```

```
(150, 4) (150,)
```

## 3 Pre Processing

[3]:
```
# No NAN Values
np.isnan(X).all() or np.isnan(y).all()
```

[3]: False

[4]:
```
# Splitting Test Train Dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

[5]:
```
# Normalisation
scalar_X = StandardScaler()

X_train = scalar_X.fit_transform(X_train)
X_test = scalar_X.transform(X_test)
```

## 4 Training

[6]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, f1_score
```

[7]:
```
dt = DecisionTreeClassifier()
lr = LogisticRegression()
nb = GaussianNB()
svm = SVC()
```

[8]:
```
# Training
dt.fit(X_train, y_train)
lr.fit(X_train, y_train)
nb.fit(X_train, y_train)
svm.fit(X_train, y_train)
```

[8]:
```
SVC()
```

## 5 Analysis

[9]:
```
y_pred_dt = dt.predict(X_test)
y_pred_lr = lr.predict(X_test)
y_pred_nb = nb.predict(X_test)
y_pred_svm = svm.predict(X_test)
```

[10]:
```
pred = [y_pred_dt, y_pred_lr, y_pred_nb, y_pred_svm]

f1 = [f1_score(y_test, y, average="macro") for y in pred]
acc = [accuracy_score(y_test, y) for y in pred]
```

[11]:
```
print("\033[1m F1 Score \033[0m")
classifier = ["Decesion Tree", "Logistic Regression", "Naive Bayes", "SVM"]

for i in range(len(f1)):
    print(f"{classifier[i]}: {f1[i]:0.2f}")

print()
print("\033[1m Accuracy \033[0m")

for i in range(len(f1)):
    print(f"{classifier[i]}: {acc[i]:0.2f}")
```

```
 F1 Score
Decesion Tree: 0.97
```

```
Logistic Regression: 0.97
Naive Bayes: 0.97
SVM: 0.94

 Accuracy
Decesion Tree: 0.97
Logistic Regression: 0.97
Naive Bayes: 0.97
SVM: 0.93
```