

## Ishan Gupta - Linear Support Vector Machine (SVM) - 19BCE7467

```
✓ [3] import numpy as np
1s import matplotlib.pyplot as plt
import pandas as pd
dataset = pd.read_csv('/content/diabetes.csv')
print(dataset)
```

|     | Pregnancies | Glucose | ... | Age | Outcome |
|-----|-------------|---------|-----|-----|---------|
| 0   | 6           | 148     | ... | 50  | 1       |
| 1   | 1           | 85      | ... | 31  | 0       |
| 2   | 8           | 183     | ... | 32  | 1       |
| 3   | 1           | 89      | ... | 21  | 0       |
| 4   | 0           | 137     | ... | 33  | 1       |
| ..  | ...         | ...     | ... | ... | ...     |
| 763 | 10          | 101     | ... | 63  | 0       |
| 764 | 2           | 122     | ... | 27  | 0       |
| 765 | 5           | 121     | ... | 30  | 0       |
| 766 | 1           | 126     | ... | 47  | 1       |
| 767 | 1           | 93      | ... | 23  | 0       |

[768 rows x 9 columns]

```
✓ [4] X = dataset.iloc[:, :-1].values
0s y = dataset.iloc[:, -1].values
```

```
✓ [5] # Splitting the dataset into the Training set and Test set
0s from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

```
✓ [6] # Feature Scaling
0s from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
✓ [7] # Training the SVM model on the Training set
0s from sklearn.svm import LinearSVC
classifier = LinearSVC(random_state = 0)
classifier.fit(X_train, y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/svm/_base.py:947: ConvergenceWarning: Liblin
"the number of iterations.", ConvergenceWarning)
LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
intercept_scaling=1, loss='squared_hinge', max_iter=1000,
multi_class='ovr', penalty='l2', random_state=0, tol=0.0001,
verbose=0)
```

```
✓ [8] # Making the Confusion Matrix
0s from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = classifier.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[117 13]
 [ 26 36]]
0.796875
```