

```
from sklearn import datasets
import pandas as pd
```

```
# load datasets
```

```
iris = datasets.load_iris()
```

```
titanic = pd.read_excel('train.xlsx')
titanic = pd.read_csv('train.csv')
```

KNN → K Nearest Neighbors – with splitting – Lab 1

```
import sklearn.model_selection as skms
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier

# Simple train/test split of the dataset
(iris_train_ftrs, iris_test_ftrs, iris_train_tgt, iris_test_tgt) =
skms.train_test_split(iris.data, iris.target, test_size=.25)

# default n_neighbors = 5
knn = KNeighborsClassifier(n_neighbors=3)
fit = knn.fit(iris_train_ftrs, iris_train_tgt)
preds = fit.predict(iris_test_ftrs)

# Evaluate the accuracy of the model
accuracy = accuracy_score(iris_test_tgt, preds)
print("Accuracy:", accuracy)
```

Tree → Tree – without splitting – Lab 2

```
from sklearn.model_selection import cross_val_score
from sklearn import tree

dtc = tree.DecisionTreeClassifier()

dtc.fit(iris.data, iris.target)

scores = cross_val_score(dtc, iris.data, iris.target, cv=3, scoring='accuracy')

print("Cross-validated accuracy scores:", scores)
```

GaussianNB → Gaussian Network – with splitting – Lab 3

```
from sklearn.naive_bayes import GaussianNB
import sklearn.model_selection as skms
from sklearn import metrics

iris_train_ftrs, iris_test_ftrs, iris_train_tgt, iris_test_tgt =
skms.train_test_split(iris.data, iris.target, test_size = .25)

nb = GaussianNB()

fit = nb.fit(iris_train_ftrs, iris_train_tgt)

pred = fit.predict(iris_test_ftrs)

accuracy = metrics.accuracy_score(iris_test_tgt, pred)
print(f'NB Accuracy: {accuracy}')
```

Confusion Matrix – Lab 3

```
from sklearn import metrics
import matplotlib.pyplot as plt

# Confusion matrix
con_mat = metrics.confusion_matrix(cancer_test_tgt, y_pred)

# Plotting the confusion matrix as Heatmap
# plt.figure(figsize=(8,6)) not needed it doesn't effect on the plot its for quality

sns.heatmap(con_mat, annot=True, fmt='d', cmap='Blues', cbar=False,
xticklabels=cancer.target_names, yticklabels=cancer.target_names)
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True label')
plt.show()
```

Tree – Lab2

```
from sklearn import metrics
import matplotlib.pyplot as plt

Model = tree.DecisionTreeClassifier(criterion='entropy', max_depth=best_depth,
min_samples_leaf=best_min_samples)

# plt.figure(figsize=(15,10)) not needed it doesn't effect on the plot its for quality

plot_tree(Model, filled=True, feature_names=['Pclass', 'Age', 'female'], class_names=['Not
Survived', 'Survived'])
plt.show()
```

