**Car evaluation dataset , used Randam forest , SVM and KNN**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay, accuracy\_score

import matplotlib.pyplot as plt

url = "/content/car\_evaluation.csv"

column\_names = ['buying', 'maint', 'doors', 'persons', 'lug\_boot', 'safety', 'class']

df = pd.read\_csv(url, names=column\_names)

label\_encoder = LabelEncoder()

for col in df.columns:

    df[col] = label\_encoder.fit\_transform(df[col])

X = df.drop('class', axis=1)

y = df['class']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

rf\_model = RandomForestClassifier(random\_state=42)

rf\_model.fit(X\_train, y\_train)

rf\_predictions = rf\_model.predict(X\_test)

rf\_accuracy = accuracy\_score(y\_test, rf\_predictions)

print(f"Random Forest Accuracy: {rf\_accuracy \* 100:.2f}%")

feature\_importances = rf\_model.feature\_importances\_

features = X.columns

plt.figure(figsize=(8, 5))

plt.barh(features, feature\_importances, color='blue')

plt.title('Random Forest Feature Importance')

plt.xlabel('Importance')

plt.ylabel('Features')

plt.show()

rf\_cm = confusion\_matrix(y\_test, rf\_predictions)

disp\_rf = ConfusionMatrixDisplay(confusion\_matrix=rf\_cm)

disp\_rf.plot(cmap='Blues')

plt.title('Random Forest Confusion Matrix')

plt.show()

svm\_model = SVC(kernel='linear')

svm\_model.fit(X\_train, y\_train)

svm\_predictions = svm\_model.predict(X\_test)

svm\_accuracy = accuracy\_score(y\_test, svm\_predictions)

print(f"SVM Accuracy: {svm\_accuracy \* 100:.2f}%")

svm\_cm = confusion\_matrix(y\_test, svm\_predictions)

disp\_svm = ConfusionMatrixDisplay(confusion\_matrix=svm\_cm)

disp\_svm.plot(cmap='Greens')

plt.title('SVM Confusion Matrix')

plt.show()

knn\_model = KNeighborsClassifier(n\_neighbors=5)

knn\_model.fit(X\_train, y\_train)

knn\_predictions = knn\_model.predict(X\_test)

knn\_accuracy = accuracy\_score(y\_test, knn\_predictions)

print(f"KNN Accuracy: {knn\_accuracy \* 100:.2f}%")

knn\_cm = confusion\_matrix(y\_test, knn\_predictions)

disp\_knn = ConfusionMatrixDisplay(confusion\_matrix=knn\_cm)

disp\_knn.plot(cmap='Oranges')

plt.title('KNN Confusion Matrix')

plt.show()

Output :-



 