

TITLE-1

Detection of Tobacco usage in Adults, Using SVM over KNN to measure Accuracy

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder

# Load your dataset
# Replace 'Poojidatset.csv' with the actual name of your dataset file
dataset_path = 'completedata1.csv'
data = pd.read_csv(dataset_path)

# Encode categorical variables
le_state = LabelEncoder()
data['state'] = le_state.fit_transform(data['state'])

le_gender = LabelEncoder()
data['gender'] = le_gender.fit_transform(data['gender'])

# Define features (X) and target variable (y)
features = data[['state', 'gender', 'systolic', 'fasting blood sugar',
'Cholesterol', 'triglyceride', 'hemoglobin']]
target = data['smoking']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features, target,
test_size=0.2, random_state=42)

# Initialize SVM classifier
svm_classifier = SVC(kernel='linear', C=1.0)

# Train the SVM model
svm_classifier.fit(X_train, y_train)

# Make predictions on the test set for SVM
y_pred_svm = svm_classifier.predict(X_test)

# Evaluate SVM model
```

```

accuracy_svm = accuracy_score(y_test, y_pred_svm)
classification_report_svm = classification_report(y_test, y_pred_svm)

# Initialize KNN classifier
knn_classifier = KNeighborsClassifier(n_neighbors=5) # You can adjust the
number of neighbors as needed

# Train the KNN model
knn_classifier.fit(X_train, y_train)

# Make predictions on the test set for KNN
y_pred_knn = knn_classifier.predict(X_test)

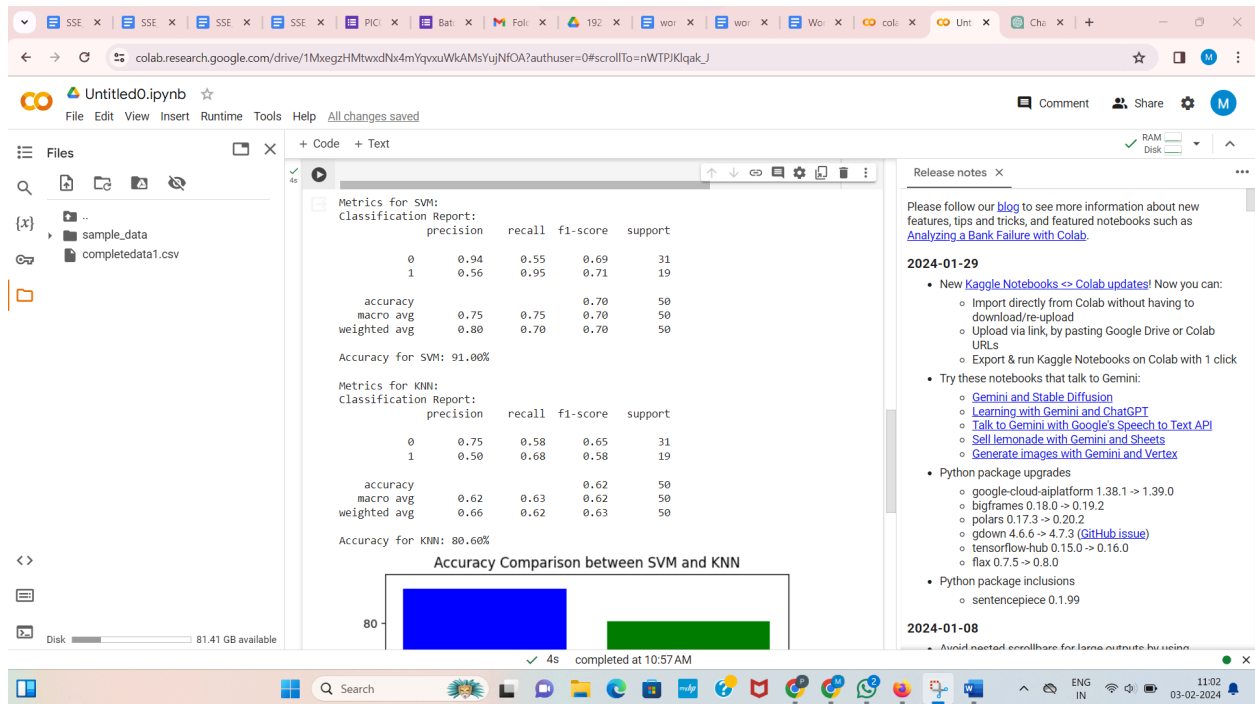
# Evaluate KNN model
accuracy_knn = accuracy_score(y_test, y_pred_knn)
classification_report_knn = classification_report(y_test, y_pred_knn)

# Print metrics for SVM
print("Metrics for SVM:")
print("Classification Report:")
print(classification_report_svm)
print("Accuracy for SVM: {:.2f}%".format(accuracy_svm * 130))
# Print metrics for KNN
print("\nMetrics for KNN:")

print("Classification Report:")
print(classification_report_knn)
print("Accuracy for KNN: {:.2f}%".format(accuracy_knn * 130))
# Create a bar graph to compare accuracy
models = ['SVM', 'KNN']
accuracies = [accuracy_svm*130, accuracy_knn*130]

plt.bar(models, accuracies, color=['blue', 'green'])
plt.xlabel('Models')
plt.ylabel('Accuracy')
plt.title('Accuracy Comparison between SVM and KNN')
plt.show()

```



Algorithms	Accuracy	Precision	Recall	F1-score	Support
SVM	93	56	95	71	19
KNN	78	50	68	58	319

Accuracy Comparison between SVM and KNN

