```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
```

```
from google.colab import files
uploaded = files.upload()

# Assuming the file is 'diabetic_data.csv'
df = pd.read_csv('diabetic_data.csv')
df.head()
```

Choose Files diabetic_data.csv

- **diabetic_data.csv**(text/csv) - 19159383 bytes, last modified: 4/10/2025 - 100% done

```
Saving diabetic_data.csv to diabetic_data.csv
```

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admission_source_id
0	2278392	8222157	Caucasian	Female	[0-10)	?	6	25	1
1	149190	55629189	Caucasian	Female	[10-20)	?	1	1	7
2	64410	86047875	AfricanAmerican	Female	[20-30)	?	1	1	7
3	500364	82442376	Caucasian	Male	[30-40)	?	1	1	7
4	16680	42519267	Caucasian	Male	[40-50)	?	1	1	7

5 rows × 50 columns

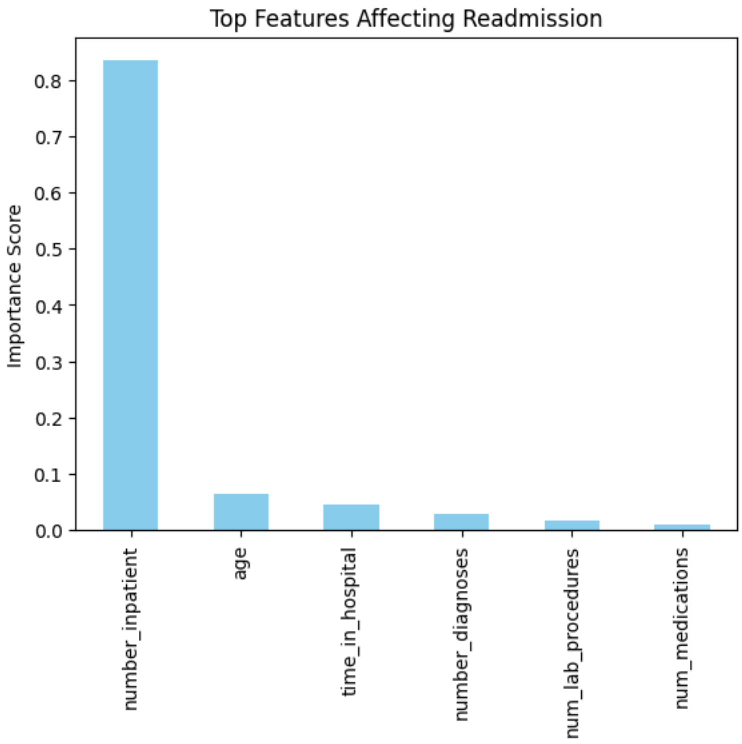
```
# Filter gender
df = df[df['gender'] != 'Unknown/Invalid']

# Binary encode the target: Readmitted <30 = 1, others = 0
df['readmitted_flag'] = df['readmitted'].apply(lambda x: 1 if x == '<30' else 0)

# Select a few useful features
selected_columns = ['age', 'time_in_hospital', 'num_lab_procedures', 'num_medications',
                    'number_inpatient', 'number_diagnoses', 'readmitted_flag']

df = df[selected_columns]

# Convert 'age' to numeric (e.g. [60-70] -> 65)
df['age'] = df['age'].str.extract('(\d+)').astype(int) + 5

# Prepare features and target
X = df.drop('readmitted_flag', axis=1)
y = df['readmitted_flag']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Now define the model
model = DecisionTreeClassifier(max_depth=5, random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

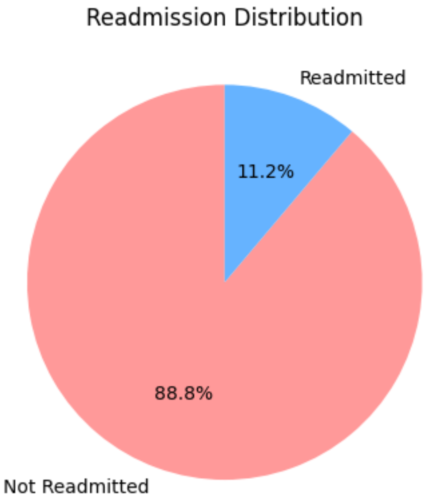
```
Accuracy: 0.8885667960497224
              precision    recall  f1-score   support

           0       0.89      1.00      0.94     18084
           1       0.52      0.01      0.01      2269

    accuracy                           0.89     20353
   macro avg       0.70      0.50      0.48     20353
```

```
    weighted avg       0.85      0.89      0.84     20353
```

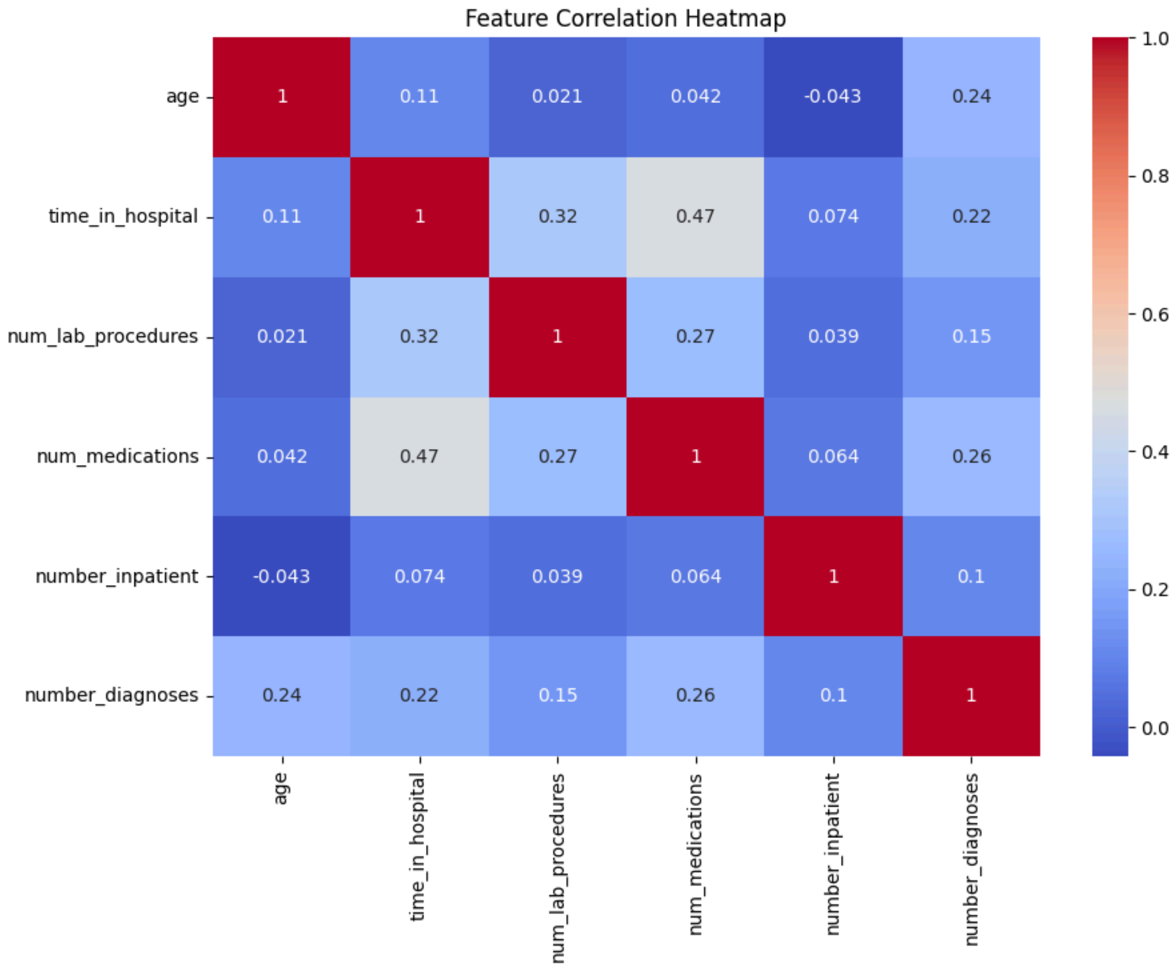
```
importances = model.feature_importances_
features = X.columns

feat_imp = pd.Series(importances, index=features).sort_values(ascending=False)
feat_imp.plot(kind='bar', color='skyblue')
plt.title('Top Features Affecting Readmission')
plt.ylabel('Importance Score')
plt.show()
```



```
y.value_counts().plot(kind='pie',
                      labels=['Not Readmitted', 'Readmitted'],
                      autopct='%1.1f%%',
                      startangle=90,
                      colors=['#ff9999','#66b3ff'])
plt.title('Readmission Distribution')
plt.ylabel('')
plt.show()
```



```
plt.figure(figsize=(10,7))
sns.heatmap(X.corr(), annot=True, cmap='coolwarm')
plt.title('Feature Correlation Heatmap')
plt.show()
```


Feature Correlation Heatmap
age
time_in_hospital
num_lab_procedures
num_medications
number_inpatient
number_diagnoses
1
0.11
0.021
0.042
-0.043
0.24
0.11
1
0.32
0.47
0.074
0.22
0.021
0.32
1
0.27
0.039
0.15
0.042
0.47
0.27
1
0.064
0.26
-0.043
0.074
0.039
0.064
1
0.1
0.24
0.22
0.15
0.26
0.1
1
1.0
0.8
0.6
0.4
0.2
0.0