


```
In [1]: pip install streamlit
```

Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: streamlit in c:\users\belea\appdata\roaming\python\python311\site-packages (1.34.0)
Requirement already satisfied: altair<6,>=4.0 in c:\users\belea\appdata\roaming\python\python311\site-packages (from streamlit) (5.3.0)
Requirement already satisfied: blinker<2,>=1.0.0 in c:\users\belea\appdata\roaming\python\python311\site-packages (from streamlit) (1.8.2)
Requirement already satisfied: cachetools<6,>=4.0 in c:\users\belea\appdata\roaming\python\python311\site-packages (from streamlit) (5.3.2)
Requirement already satisfied: click<9,>=7.0 in c:\programdata\anaconda3\lib\site-packages (from streamlit) (8.0.4)
Requirement already satisfied: numpy<2,>=1.19.3 in c:\programdata\anaconda3\lib\site-packages (from streamlit) (1.24.3)
Requirement already satisfied: packaging<25,>=16.8 in c:\programdata\anaconda3\lib\site-packages (from streamlit) (23.0)
Requirement already satisfied: pandas<3,>=1.3.0 in c:\programdata\anaconda3\lib\site-packages (from streamlit) (1.5.3)
Requirement already satisfied: pillow<11,>=7.1.0 in c:\programdata\anaconda3\lib\site-packages (from streamlit) (10.2.0)
Requirement already satisfied: protobuf<5,>=3.20 in c:\users\belea\appdata\roaming\python\python311\site-packages (from streamlit) (4.23.4)
Requirement already satisfied: pyarrow>=7.0 in c:\programdata\anaconda3\lib\site-packages (from streamlit) (11.0.0)
Requirement already satisfied: requests<3,>=2.27 in c:\programdata\anaconda3\lib\site-packages (from streamlit) (2.31.0)
Requirement already satisfied: rich<14,>=10.14.0 in c:\users\belea\appdata\roaming\python\python311\site-packages (from streamlit) (13.7.0)
Requirement already satisfied: tenacity<9,>=8.1.0 in c:\programdata\anaconda3\lib\site-packages (from streamlit) (8.2.2)
Requirement already satisfied: toml<2,>=0.10.1 in c:\programdata\anaconda3\lib\site-packages (from streamlit) (0.10.2)
Requirement already satisfied: typing-extensions<5,>=4.3.0 in c:\users\belea\appdata\roaming\python\python311\site-packages (from streamlit) (4.9.0)
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in c:\users\belea\appdata\roaming\python\python311\site-packages (from streamlit) (3.1.42)
Requirement already satisfied: pydeck<1,>=0.8.0b4 in c:\users\belea\appdata\roaming\python\python311\site-packages (from streamlit) (0.9.1)
Requirement already satisfied: tornado<7,>=6.0.3 in c:\programdata\anaconda3\lib\site-packages (from streamlit) (6.3.2)
Requirement already satisfied: watchdog>=2.1.5 in c:\programdata\anaconda3\lib\site-packages (from streamlit) (2.1.6)
Requirement already satisfied: jinja2 in c:\programdata\anaconda3\lib\site-packages (from altair<6,>=4.0->streamlit) (3.1.2)
Requirement already satisfied: jsonschema>=3.0 in c:\programdata\anaconda3\lib\site-packages (from altair<6,>=4.0->streamlit) (4.17.3)
Requirement already satisfied: toolz in c:\programdata\anaconda3\lib\site-packages (from altair<6,>=4.0->streamlit) (0.12.0)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-packages (from click<9,>=7.0->streamlit) (0.4.6)
Requirement already satisfied: gitdb<5,>=4.0.1 in c:\users\belea\appdata\roaming\python\python311\site-packages (from gitpython!=3.1.19,<4,>=3.0.7->streamlit) (4.0.11)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\programdata\anaconda3\lib\site-packages (from pandas<3,>=1.3.0->streamlit) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\site-packages (from pandas<3,>=1.3.0->streamlit) (2022.7)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\ana

```
conda3\lib\site-packages (from requests<3,>=2.27->streamlit) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.27->streamlit) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.27->streamlit) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.27->streamlit) (2024.2.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\programdata\anaconda3\lib\site-packages (from rich<14,>=10.14.0->streamlit) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\programdata\anaconda3\lib\site-packages (from rich<14,>=10.14.0->streamlit) (2.15.1)
Requirement already satisfied: smmap<6,>=3.0.1 in c:\users\belea\AppData\Roaming\Python\Python311\site-packages (from gitdb<5,>=4.0.1->gitpython!=3.1.19,<4,>=3.0.7->streamlit) (5.0.1)
Requirement already satisfied: MarkupSafe>=2.0 in c:\programdata\anaconda3\lib\site-packages (from jinja2->altair<6,>=4.0->streamlit) (2.1.1)
Requirement already satisfied: attrs>=17.4.0 in c:\programdata\anaconda3\lib\site-packages (from jsonschema>=3.0->altair<6,>=4.0->streamlit) (22.1.0)
Requirement already satisfied: pyparsing!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in c:\programdata\anaconda3\lib\site-packages (from jsonschema>=3.0->altair<6,>=4.0->streamlit) (0.18.0)
Requirement already satisfied: mdurl~=0.1 in c:\programdata\anaconda3\lib\site-packages (from markdown-it-py>=2.2.0->rich<14,>=10.14.0->streamlit) (0.1.0)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas<3,>=1.3.0->streamlit) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```



```
In [2]: import streamlit as st
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, StratifiedKFold
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.feature_selection import RFECV
import missingno as msno
import warnings

warnings.filterwarnings('ignore')

# Function to train and evaluate the model
def train_and_evaluate(model, X_train, X_test, Y_train, Y_test):
    model.fit(X_train, Y_train)
    Y_train_pred = model.predict(X_train)
    Y_test_pred = model.predict(X_test)
    return {
        "train_accuracy": accuracy_score(Y_train, Y_train_pred),
        "test_accuracy": accuracy_score(Y_test, Y_test_pred),
        "confusion_matrix": confusion_matrix(Y_test, Y_test_pred),
        "classification_report": classification_report(Y_test, Y_test_pred, out
    }

st.title('Diabetes Prediction Web App')

# File upload
uploaded_file = st.file_uploader("Choose a CSV file", type="csv")
if uploaded_file is not None:
    df = pd.read_csv(uploaded_file)
    st.write("Dataset:")
    st.write(df.head())

# Data preprocessing
df_copy = df.copy(deep=True)
df_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']] =
df_copy['Glucose'].fillna(df_copy['Glucose'].mean(), inplace=True)
df_copy['BloodPressure'].fillna(df_copy['BloodPressure'].mean(), inplace=True)
df_copy['SkinThickness'].fillna(df_copy['SkinThickness'].median(), inplace=True)
df_copy['Insulin'].fillna(df_copy['Insulin'].median(), inplace=True)
df_copy['BMI'].fillna(df_copy['BMI'].median(), inplace=True)

# EDA
st.subheader('Exploratory Data Analysis')
st.write("Missing values after imputation:")
st.write(df_copy.isnull().sum())

st.write("Data Distribution")
st.write(msno.bar(df_copy))
```

```

st.write("Histograms")
fig, ax = plt.subplots(3, 3, figsize=(20, 15))
df_copy.hist(bins=50, ax=ax)
st.pyplot(fig)

st.write("Correlation Heatmap")
corr_matrix = df_copy.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
st.pyplot()

# Feature engineering
df_copy['Nutritional_Status'] = pd.cut(df_copy['BMI'], bins=[-1, 18.5, 25,
def glucose_result(glucose):
    if glucose <= 140:
        return "Normal"
    elif glucose <= 198:
        return "Impaired Glucose Tolerance"
    else:
        return "Diabetic Level"
df_copy['Glucose Result'] = df_copy['Glucose'].apply(glucose_result)
def percentile_skin_thickness(age, skin_thickness):
    if age >= 20 and age <= 79:
        if skin_thickness < 11.9:
            return "<P5th"
        elif skin_thickness < 14.0:
            return "P5th-P10th"
        elif skin_thickness < 15.8:
            return "P10th-P15th"
        elif skin_thickness < 18.0:
            return "P15th-P25th"
        elif skin_thickness < 23.5:
            return "P25th-P50th"
        elif skin_thickness < 29.0:
            return "P50th-P75th"
        elif skin_thickness < 31.9:
            return "P75th-P85th"
        elif skin_thickness < 33.7:
            return "P85th-P90th"
        elif skin_thickness < 35.9:
            return "P90th-P95th"
        else:
            return ">P95th"
    elif age >= 80:
        return ">P95th"
    else:
        return "NA"
df_copy['Percentile skin thickness'] = df_copy.apply(lambda row: percentile

# Feature and target separation
X = df_copy.drop(columns=['Outcome', 'Glucose Result', 'Percentile skin thi
Y = df_copy['Outcome']

# Standardization
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

```

```

# Train-test split
X_train, X_test, Y_train, Y_test = train_test_split(X_scaled, Y, test_size=

# Model training and evaluation
models = {
    "K-Nearest Neighbors": KNeighborsClassifier(n_neighbors=5),
    "Support Vector Machine": SVC(kernel='linear'),
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier(n_estimators=200),
    "XGBoost": XGBClassifier(gamma=0)
}

results = {}
for model_name, model in models.items():
    results[model_name] = train_and_evaluate(model, X_train, X_test, Y_train)

st.subheader('Model Performance')
for model_name, result in results.items():
    st.write(f"### {model_name}")
    st.write(f"Accuracy on training data: {result['train_accuracy']}")
    st.write(f"Accuracy on test data: {result['test_accuracy']}")
    st.write("Confusion Matrix:")
    st.write(result['confusion_matrix'])
    st.write("Classification Report:")
    st.write(result['classification_report'])

# Feature selection using RFECV
rfecv = RFECV(estimator=SVC(kernel='linear'), step=1, cv=StratifiedKFold(10))
rfecv.fit(X, Y)
st.write("Optimal number of features:", rfecv.n_features_)
st.write("Selected features:", X.columns[rfecv.support_])

```

2024-05-23 02:02:33.914

Warning: to view this Streamlit app on a browser, run it with the following command:

```
streamlit run C:\ProgramData\anaconda3\Lib\site-packages\ipykernel_launcher
er.py [ARGUMENTS]
```

In []: