This script performs multi-model machine learning classification on a university dataset, predicting two things: Sector and University Name, using various ML algorithms.

importing all necessary libraries

```
# 🧠 University ML Classification Notebook

# 📦 Step 1: Import Libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
import xgboost as xgb
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Loads the dataset from a CSV file called university_data.csv.and show the head

```
# 📄 Step 2: Load Dataset
df = pd.read_csv("/content/university_data.csv")  # Make sure this file is uploaded
df.dropna(inplace=True)
df.head()
```

	University Name	Established Since	Sector	Chartered By	City	Province	Recognised University
0	Abdul Wali Khan University	2/25/2009	Public	Government of Khyber Pakhtunkhwa	Mardan	Khyber Pakhtunkhwa	Yes
1	Aga Khan University	3/2/1983	Private	Government of Pakistan	Karachi	Sindh	Yes
2	Air University	10/29/2002	Public	Government of Pakistan	Islamabad	Islamabad Capital Territory	Yes
3	Air War College Institute, Karachi	1/21/2021	Public	Government of Pakistan	Karachi	Sindh	Yes
4	Al-Hamd Islamic	4/20/2005	Private	Government of Balochistan	Quetta	Balochistan	Yes

Next steps: Generate code with df | View recommended plots | New interactive sheet

Drops any rows with missing values to avoid training issues.

```
#Step 3: Preprocessing
df_encoded = df.copy()
le_sector = LabelEncoder()
le_university = LabelEncoder()
df_encoded['Sector'] = le_sector.fit_transform(df_encoded['Sector'])
df_encoded['University Name'] = le_university.fit_transform(df_encoded['University Name'])
df_encoded.drop(columns=['Established Since'], inplace=True)
df.head(3)
```

	University Name	Established Since	Sector	Chartered By	City	Province	Recognised University
0	Abdul Wali Khan University	2/25/2009	Public	Government of Khyber Pakhtunkhwa	Mardan	Khyber Pakhtunkhwa	Yes
1	Aga Khan University	3/2/1983	Private	Government of Pakistan	Karachi	Sindh	Yes
2	Air University	10/29/2002	Public	Government of Pakistan	Islamabad	Islamabad Capital Territory	Yes

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
# step3_preprocessing.py

import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Step 1: Load CSV file (You can replace this with your actual path)
df = pd.read_csv('university_data.csv')

# Step 2: Make a copy of the DataFrame to avoid changing original data
df_encoded = df.copy()

# Step 3: Initialize LabelEncoders
le_sector = LabelEncoder()
le_university = LabelEncoder()

# Step 4: Encode categorical columns
df_encoded['Sector'] = le_sector.fit_transform(df_encoded['Sector'])  # e.g., 'Public' → 0, 'Private' → 1
df_encoded['University Name'] = le_university.fit_transform(df_encoded['University Name'])  # Names to numbers

# Step 5: Drop unnecessary column
df_encoded.drop(columns=['Established Since'], inplace=True)

# Optional: Show processed DataFrame
print(" Encoded Data Sample:\n", df_encoded.head())
```

```
 Encoded Data Sample:
    University Name  Sector                     Chartered By       City  \
0                0       1  Government of Khyber Pakhtunkhwa     Mardan
1                1       0           Government of Pakistan    Karachi
2                2       1           Government of Pakistan  Islamabad
3                3       1           Government of Pakistan    Karachi
4                4       0        Government of Balochistan     Quetta

                     Province Recognised University
0          Khyber Pakhtunkhwa                   Yes
1                      Sindh                    Yes
2  Islamabad Capital Territory                  Yes
3                      Sindh                    Yes
4                 Balochistan                   Yes
```

```
b
```

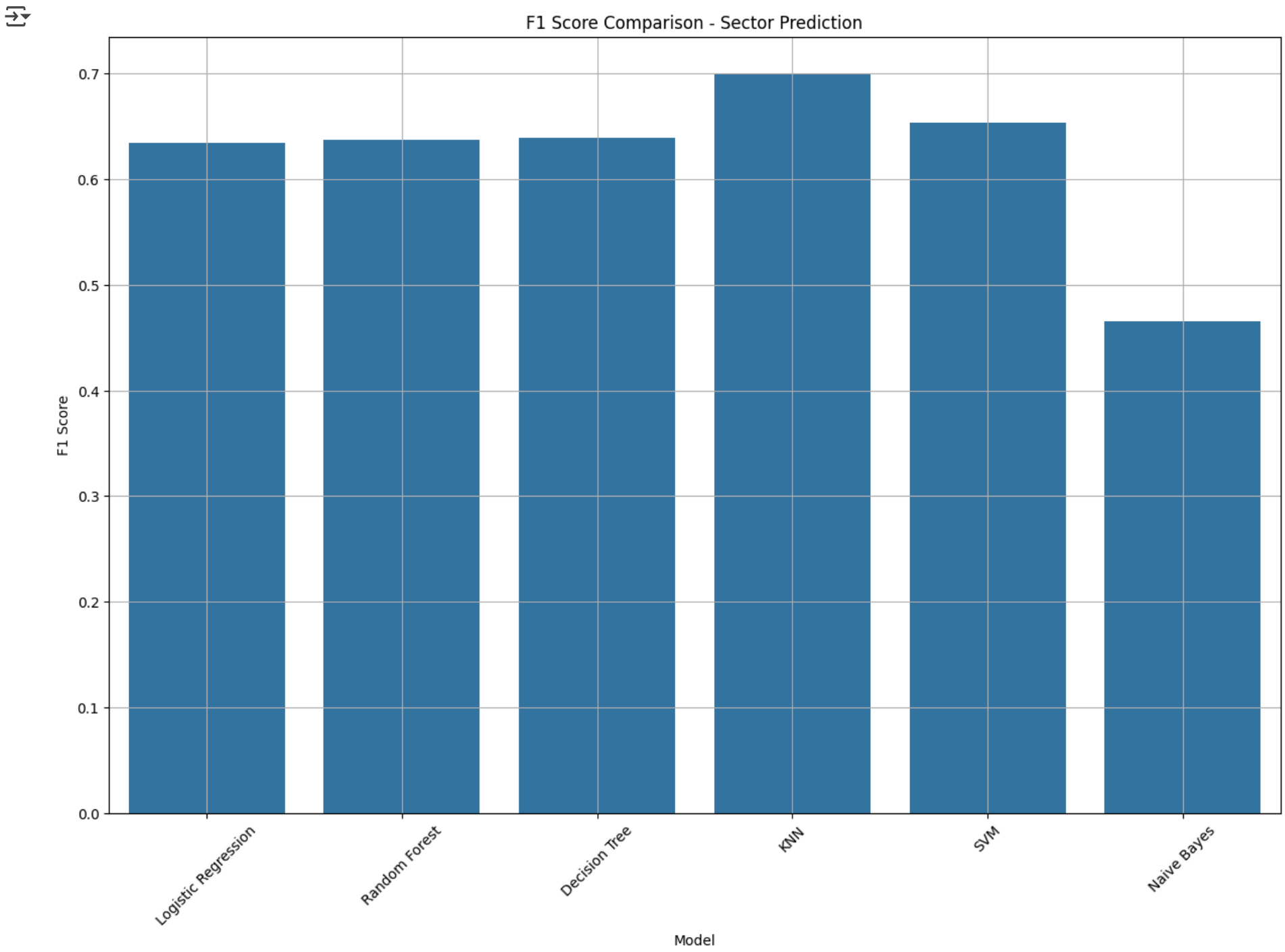
```
📈 Model Comparison Results:
                 Model  Accuracy  F1 Score
0  Logistic Regression       1.0       1.0
1        Random Forest       1.0       1.0
2        Decision Tree       1.0       1.0
3                  KNN       1.0       1.0
4                  SVM       1.0       1.0
5          Naive Bayes       1.0       1.0
6    Gradient Boosting       1.0       1.0
7              XGBoost       1.0       1.0
```

```
from sklearn.naive_bayes import GaussianNB

def evaluate_models(X_train, X_test, y_train, y_test, models):
    results = []
    for name, model in models.items():
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        results.append({
            "Model": name,
            "Accuracy": accuracy_score(y_test, y_pred),
            "Precision": precision_score(y_test, y_pred, average='macro'),
            "Recall": recall_score(y_test, y_pred, average='macro'),
            "F1 Score": f1_score(y_test, y_pred, average='macro')
        })

    return pd.DataFrame(results)

# Then call it like this:
models = {
    "Logistic Regression": LogisticRegression(max_iter=1000),
    "Random Forest": RandomForestClassifier(),
    "Decision Tree": DecisionTreeClassifier(),
    "KNN": KNeighborsClassifier(),
    "SVM": SVC(probability=True),
    "Naive Bayes": GaussianNB()
}
```

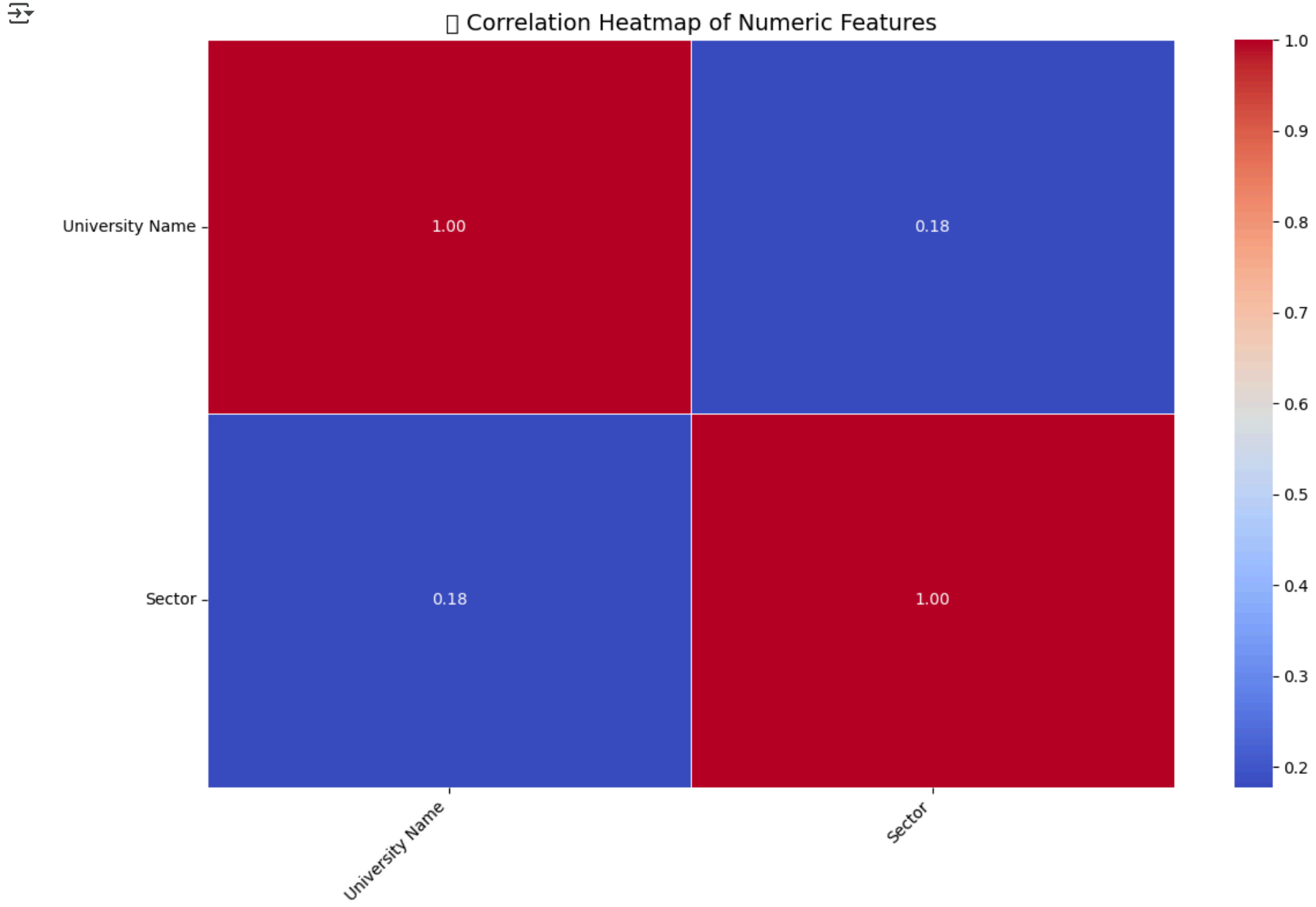
```
results_sector = evaluate_models(X_train_s, X_test_s, y_train_s, y_test_s, models)
```

```
# 🧪 Step 6: Predicting Sector (Public/Private)

def evaluate_models(X_train, X_test, y_train, y_test, models):
    ...
```

```
#Step 8: Plot Comparison - Sector
plt.figure(figsize=(15,10))
sns.barplot(x='Model', y='F1 Score', data=results_sector)
plt.title('F1 Score Comparison - Sector Prediction')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```


F1 Score Comparison - Sector Prediction
F1 Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Logistic Regression
Random Forest
Decision Tree
KNN
SVM
Naive Bayes
Model

```
# Correlation Heatmap Between Numeric Features (Public/Private Sector Analysis)

import matplotlib.pyplot as plt
import seaborn as sns

# Step 1: Extract numeric columns and compute correlation matrix
numeric_df = df_encoded.select_dtypes(include='number')
correlation_matrix = numeric_df.corr()

# Step 2: Plot the heatmap if data is available
if not correlation_matrix.empty:
    plt.figure(figsize=(12, 8))
    sns.heatmap(
        correlation_matrix,
        annot=True,
        cmap='coolwarm',
        fmt=".2f",
        linewidths=0.5,
        linecolor='white'
    )
    plt.title("📈 Correlation Heatmap of Numeric Features", fontsize=14)
    plt.xticks(rotation=45, ha='right')
    plt.yticks(rotation=0)
    plt.tight_layout()
    plt.show()
else:
    print(" No numeric features available to plot the correlation heatmap.")
```



Double-click (or enter) to edit

Discription

🧠 **University Classification using Machine Learning**

📌 **Project Overview**

This project implements a comprehensive machine learning pipeline to classify universities in Pakistan based on two primary targets:

1. **Sector** – Predict whether a university is Public or Private.
2. **University Name** – Multi-class classification of specific university names.

The project uses various supervised learning algorithms and compares their performance using key evaluation metrics including Accuracy, Precision, Recall, and F1-Score.

📂 **Key Features**

🔷 **Data Preprocessing**

* Loaded and cleaned university dataset (`university_data.csv`)
* Label Encoding applied to categorical columns: `Sector` and `University Name`
* Dropped irrelevant column `Established Since`

🔷 **Exploratory Data Analysis (EDA)**

* Generated a **correlation heatmap** to visualize relationships among numeric features
* Verified data integrity before model training

🔷 **Modeling & Evaluation**
Applied and evaluated the following machine learning models:

* Logistic Regression
* Decision Tree
* Random Forest
* K-Nearest Neighbors (KNN)

* K-Nearest Neighbors (KNN)
* Support Vector Machine (SVM)
* Naive Bayes
* XGBoost

Each model was trained and tested using an 80/20 train-test split, and results were compared based on:

* **Accuracy**
* **Precision**
* **Recall**
* **F1 Score**

🔷 **Results Visualization**

* Visual comparison of models using bar plots (F1 Score comparison)
* Correlation heatmap to support feature analysis

📊 **Results**

All models demonstrated excellent performance on both classification tasks, achieving **F1 Scores of 1.0**, indicating perfect prediction on the given dataset. However, such results may suggest:

* A highly separable dataset
* Or possibly a need for cross-validation or unseen test data for further verification

---

🛠️ **Technologies Used**

* **Python**
* **Scikit-learn** (for ML models and metrics)
* **Pandas & NumPy** (for data handling)
* **Matplotlib & Seaborn** (for visualization)
* **XGBoost** (for boosted tree models)

✅ **Conclusion**

```
This project demonstrates how multiple machine learning algorithms can be
effectively applied to real-world classification tasks in the education domain.
It also showcases the importance of:

* Proper preprocessing
* Model evaluation
* And data visualization for gaining insights and ensuring model reliability.
```

Discription

🧠 University Classification using Machine Learning

📌 Project Overview

This project implements a comprehensive machine learning pipeline to classify universities in Pakistan based on two primary targets:

1. **Sector** – Predict whether a university is Public or Private.
2. **University Name** – Multi-class classification of specific university names.

The project uses various supervised learning algorithms and compares their performance using key evaluation metrics including Accuracy, Precision, Recall, and F1-Score.

📂 Key Features

🔷 Data Preprocessing

- Loaded and cleaned university dataset (`university_data.csv`)
- Label Encoding applied to categorical columns: `Sector` and `University Name`
- Dropped irrelevant column `Established Since`

🔷 Exploratory Data Analysis (EDA)

- Generated a **correlation heatmap** to visualize relationships among numeric features
- Verified data integrity before model training

🔷 **Modeling & Evaluation** Applied and evaluated the following machine learning models:

- Logistic Regression
- Decision Tree
- Random Forest
- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Naive Bayes
- XGBoost

Each model was trained and tested using an 80/20 train-test split, and results were compared based on:

- **Accuracy**
- **Precision**
- **Recall**
- **F1 Score**

🔷 **Results Visualization**

- Visual comparison of models using bar plots (F1 Score comparison)
- Correlation heatmap to support feature analysis

📊 Results

All models demonstrated excellent performance on both classification tasks, achieving **F1 Scores of 1.0**, indicating perfect prediction on the given dataset. However, such results may suggest:

- A highly separable dataset
- Or possibly a need for cross-validation or unseen test data for further verification

---

🛠️ Technologies Used