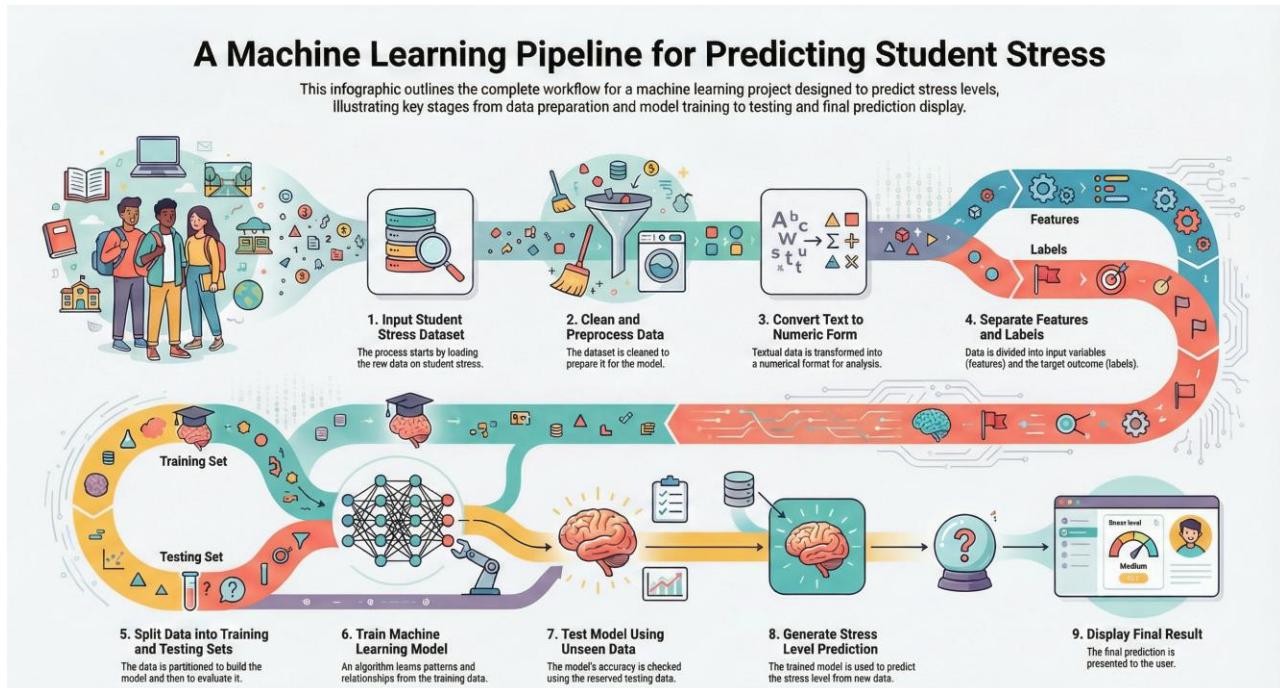




The
University
of Lahore

PROJECT TITLE:

STUDENT STRESS LEVEL PREDICTION USING MACHINE LEARNING



GROUP MEMBER:

Abdullah Bin Yousaf	70149366
Tehreem Ali Naqvi	70149255
Hammad Siddique	70145673
Areesha Javaid	70157002
Talha Ali Bhatti	70152029

SUBMITTED TO:

Dr. Syed Muhammad Hamedoon

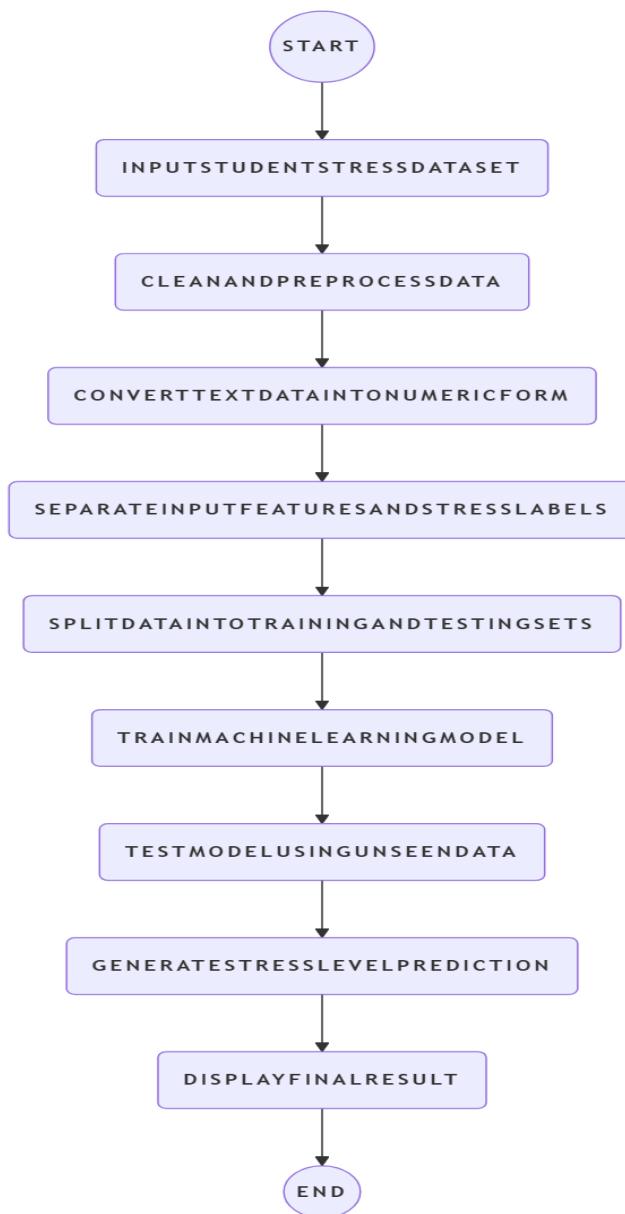
DESCRIPTION:

This project focuses on developing an AI-based student stress level prediction system using questionnaire-based data. The main goal is to identify different stress levels experienced by students, such as academic, social, or financial stress, using machine learning techniques. The dataset's consists of student responses related to sleep habits, academic pressure, lifestyle, and emotional well-being. After preprocessing the data, machine learning models are trained to classify stress levels accurately. The system helps in early identification of stress, which can assist educational institutions in taking timely actions to support students.

The final system predicts student stress levels and also provides:

- Model Accuracy
- Stress-Level Distribution
- Prediction for new students

FLOW DIAGRAM:



CODE:

```
df = pd.read_csv(
    "Stress_Dataset.csv",
    engine="python",
    encoding="latin1",
    on_bad_lines="skip"
)

df.head()
```

```
import re
new_columns = []
for col in df.columns:
    cleaned_col = col.strip().replace(' ', '_')
    # Remove any character that is not alphanumeric or underscore
    cleaned_col = re.sub(r'[^a-zA-Z0-9_]', '', cleaned_col)
    new_columns.append(cleaned_col)
df.columns = new_columns
print(df.columns)
```

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['Stress_Type_Encoded'] = le.fit_transform(df['Which_type_of_stress_do_you_primarily_experience'])

df[['Stress_Type_Encoded']].head()

Stress_Type_Encoded
0      1
1      1
2      1
3      1
4      1
```

```
target_encoder = LabelEncoder()
y = target_encoder.fit_transform(y)

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X = scaler.fit_transform(X)
```

```
from sklearn.model_selection import train_test_split

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

print(X_train.shape, X_test.shape)

... (674, 25) (169, 25)
```

```
from sklearn.metrics import accuracy_score

y_pred = rf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print("Model Accuracy:", accuracy)

... Model Accuracy: 0.9349112426035503
```

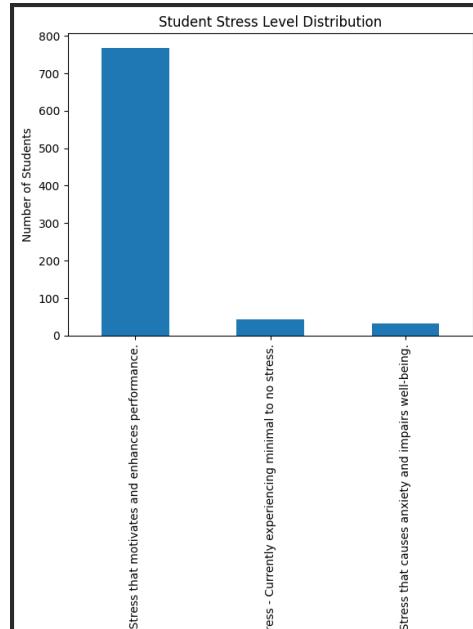
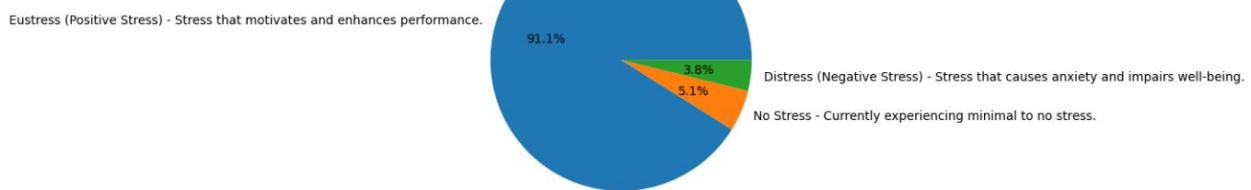
OUTPUT:

Displays the 10 rows data from the dataset

Gender	Age	Have you recently experienced stress in your life	Have you noticed a rapid heartbeat or palpitations	Have you been dealing with anxiety or tension recently	Do you attend classes regularly
			X		
0	0	20	3	4	2
1	0	20	2	3	2
2	0	20	5	4	2
3	1	20	3	4	3
4	0	20	3	3	3
5	0	20	3	4	3
6	0	20	2	2	1
7	0	20	5	5	5
8	0	20	3	2	1
9	0	21	3	4	2

Extracurricular_activities_conflicting_for_you	Do you attend classes regularly	Have you gainedlost weight	Which type of stress do you primarily experience	Stress_Type_Encoded
		X		
3	1	2	Eustress (Positive Stress) - Stress that motivates and enhances performance.	1
1	4	2	Eustress (Positive Stress) - Stress that motivates and enhances performance.	1
1	2	1	Eustress (Positive Stress) - Stress that motivates and enhances performance.	1
1	5	3	Eustress (Positive Stress) - Stress that motivates and enhances performance.	1
2	2	2	Eustress (Positive Stress) - Stress that motivates and enhances performance.	1
4	4	4	Eustress (Positive Stress) - Stress that motivates and enhances performance.	1
3	3	2	Eustress (Positive Stress) - Stress that motivates and enhances performance.	1
1	1	5	Eustress (Positive Stress) - Stress that motivates and enhances performance.	1
1	5	1	Eustress (Positive Stress) - Stress that motivates and enhances performance.	1
2	4	2	No Stress - Currently experiencing minimal to no stress.	2

Percentage of Stress Levels





Accuracy:

```
from sklearn.metrics import accuracy_score

y_pred = rf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print("Model Accuracy:", accuracy)
```

Model Accuracy: 0.9349112426035503

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)

lr_pred = lr.predict(X_test)
print("Logistic Regression Accuracy:", accuracy_score(y_test, lr_pred))

Logistic Regression Accuracy: 0.9763313609467456

from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier()
rf.fit(X_train, y_train)

rf_pred = rf.predict(X_test)
print("Random Forest Accuracy:", accuracy_score(y_test, rf_pred))

Random Forest Accuracy: 0.9349112426035503
```

```
from sklearn.svm import SVC

svm = SVC()
svm.fit(X_train, y_train)

svm_pred = svm.predict(X_test)
print("SVM Accuracy:", accuracy_score(y_test, svm_pred))

SVM Accuracy: 0.9526627218934911
```

New Student input:

```
# Example new student (same number of features)
new_student = X_test[0].reshape(1, -1)

prediction = rf.predict(new_student)
print("Predicted Stress Type:", target_encoder.inverse_transform(prediction))

Predicted Stress Type: [1]
```

SCOPE:

- Dataset cleaning and preprocessing
- Handling missing values
- Encoding categorical data
- Feature scaling and normalization
- Training multiple machine learning models
- Model evaluation using accuracy
- Visualization of stress distribution
- Prediction of stress level for new students