

# STUDENT MARKS PREDICTOR

ML PROJECT



TRISHANSH SAHANE  
VAIDIK LOTAN



# PROBLEM STATEMENT

## OBJECTIVE :

THE OBJECTIVE OF THIS PROJECT IS TO BUILD A MACHINE LEARNING MODEL USING ELASTIC-NET REGRESSION THAT PREDICTS A STUDENT'S MARKS (RESULT) BASED ON INPUT FEATURES ( SUCH AS STUDY HOURS, ATTENDANCE AND MORE).

THE IDEA IS TO UNDERSTAND HOW THESE HABITS INFLUENCE PERFORMANCE AND USE THAT RELATIONSHIP TO ESTIMATE A STUDENT'S EXPECTED RESULT.



# STEPS INVOLVED:



DATA LOADING

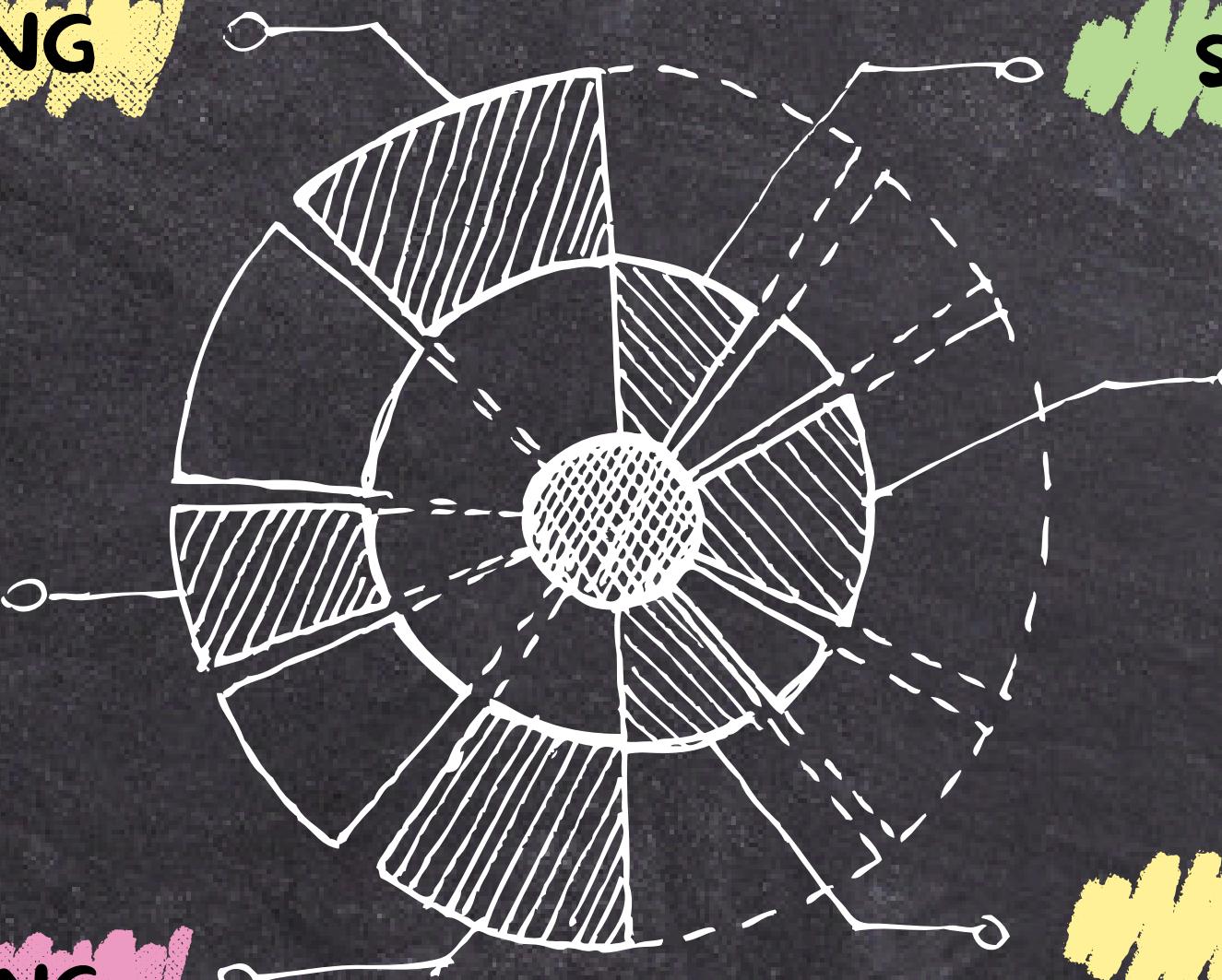
STANDARDIZATION

MODEL TRAINING

DATA CLEANING

FEATURE ENGINEERING

PREDICTION



# TECHSTACK USED

FRONTEND - HTML AND CSS

LIBRARIES AND FRAMEWORKS :

1. NUMPY
2. PANDAS
3. SEABORN
4. SCIKIT-LEARN (ML MODEL)
5. FLASK (WEB APP)

# DATA LOADING AND CLEANING



FIRSTLY, LOADED THE DATASET USING PANDAS

```
df = pd.read_csv("students_dataset.csv")
```

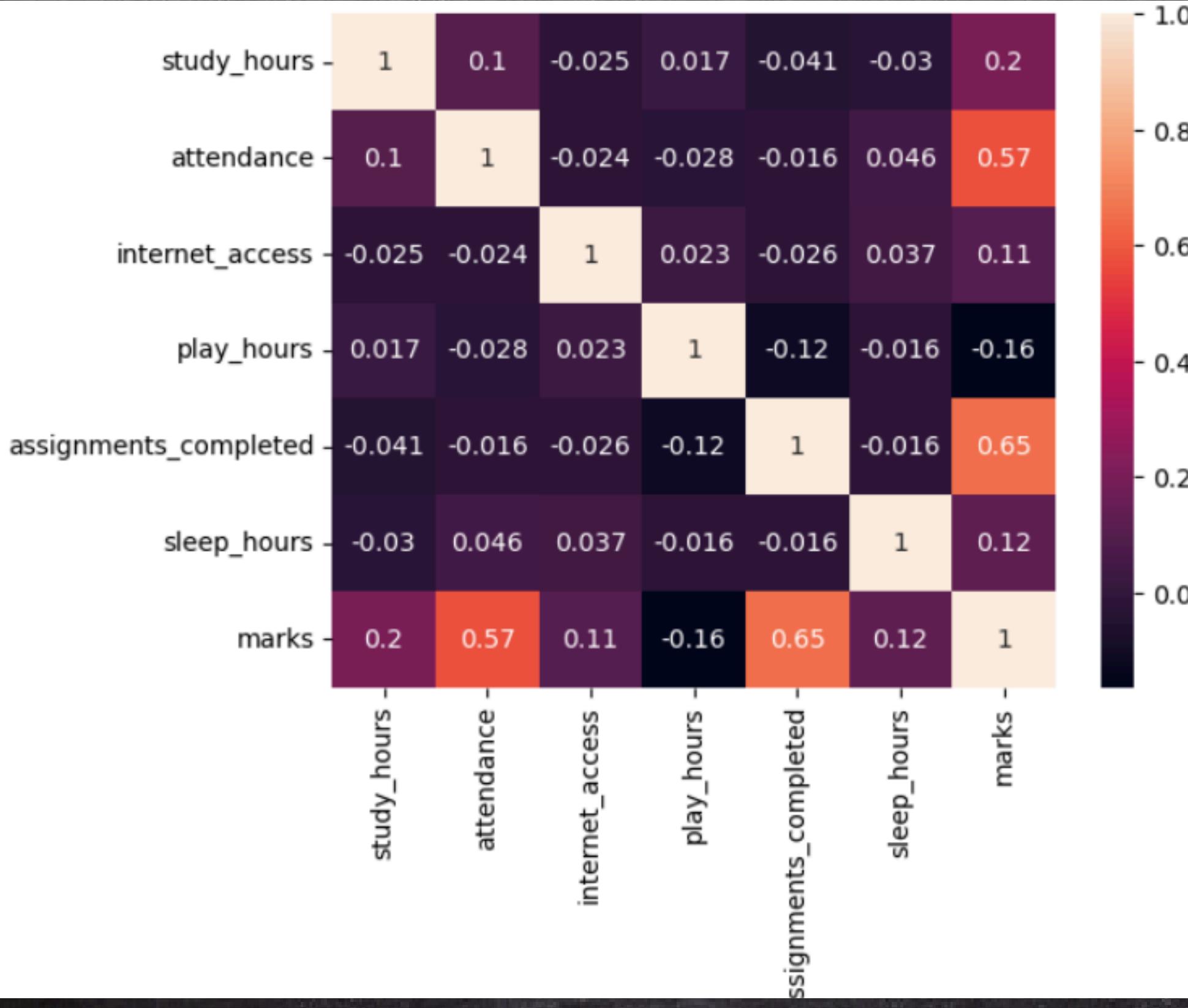
DATA IS ALREADY CLEANED, NO NULL VALUES

```
df.isnull().sum()
```

REMOVING UNNECESSARY COLUMNS

```
df = df.drop(columns=["rollno", "student_id", "name"])
```

# CORRELATION BETWEEN INPUT AND OUTPUT FEATURE



This heatmap shows how strongly each feature is connected to students marks. Higher positive values mean a strong positive link, while negative values mean the feature hurts the marks.

This heatmap shows how each factor relates to student marks. Attendance and assignments completed have the strongest positive impact, study hours help a little, and play hours show a small negative effect. The rest don't influence marks much.

# FEATURE ENGINEERING



## ENCODING CATEGORICAL FEATURES INTO NUMERICAL FEATURES

```
df["internet_access"] = df["internet_access"].map({"Yes": 1, "No": 0})
```

## DIVIDING INDEPENDENT/INPUT AND DEPENDENT/OUTPUT FEATURES FROM DATASET

```
x = df[["study_hours", "attendance", "internet_access", "play_hours", "assignments_completed", "sleep_hours"]]
y = df["marks"]
```

## SPLITTING TRAINING AND TESTING DATA

```
x_train, x_test, y_train, y_test = train_test_split(
    X, y, test_size=0.33, random_state=42
)
```

# MODEL BUILDING AND TRAINING



```
## Creating Pipeline
model_pipeline = Pipeline([
    ("scaler",StandardScaler()), ## Step 1 : Scale features
    ("model",ElasticNet(alpha=0.05,l1_ratio=0.5)) ## Step 2 : Train Model
])
## Training the model
model_pipeline.fit(X_train,y_train)
## Prediction
y_pred = model_pipeline.predict(X_test)
```

# MODEL EVALUATION



```
## Evaluation  
mse = mean_squared_error(y_test,y_pred)  
r2 = r2_score(y_test,y_pred)  
  
## Print Results  
print("MSE = ",np.round(mse,2)," r2 score = ",np.round(r2,2))
```

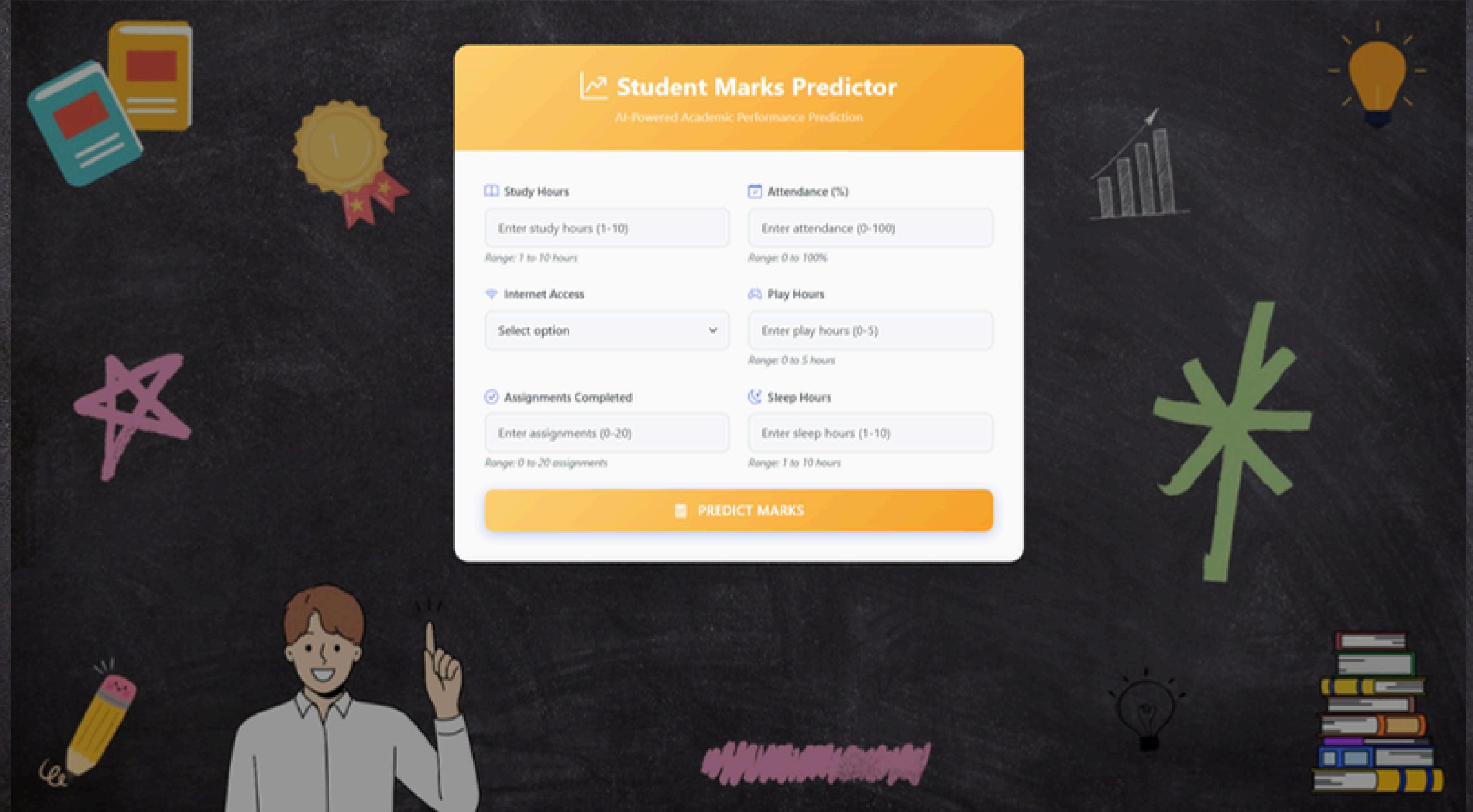
## OUTPUT

```
MSE = 11.07 r2 score = 0.83
```

AN MSE OF AROUND 11 MEANS MODEL'S PREDICTIONS ARE ON AVERAGE ONLY A LITTLE OFF FROM THE REAL MARKS.

AN R<sup>2</sup> SCORE OF 0.83 MEANS THE MODEL UNDERSTANDS ABOUT 83% OF THE PATTERN IN THE DATA, WHICH IS QUITE GOOD. (83% ACCURACY)  
OVERALL, THE MODEL PREDICTS MARKS FAIRLY ACCURATELY.

# SNAPSHOTS OF OUR PROJECT



# SNAPSHOTS OF OUR PROJECT



## 📈 Student Marks Predictor

AI-Powered Academic Performance Prediction

Study Hours  
Enter study hours (1-10)  
Range: 1 to 10 hours

Attendance (%)  
Enter attendance (0-100)  
Range: 0 to 100%

Internet Access  
Select option

Play Hours  
Enter play hours (0-5)  
Range: 0 to 5 hours

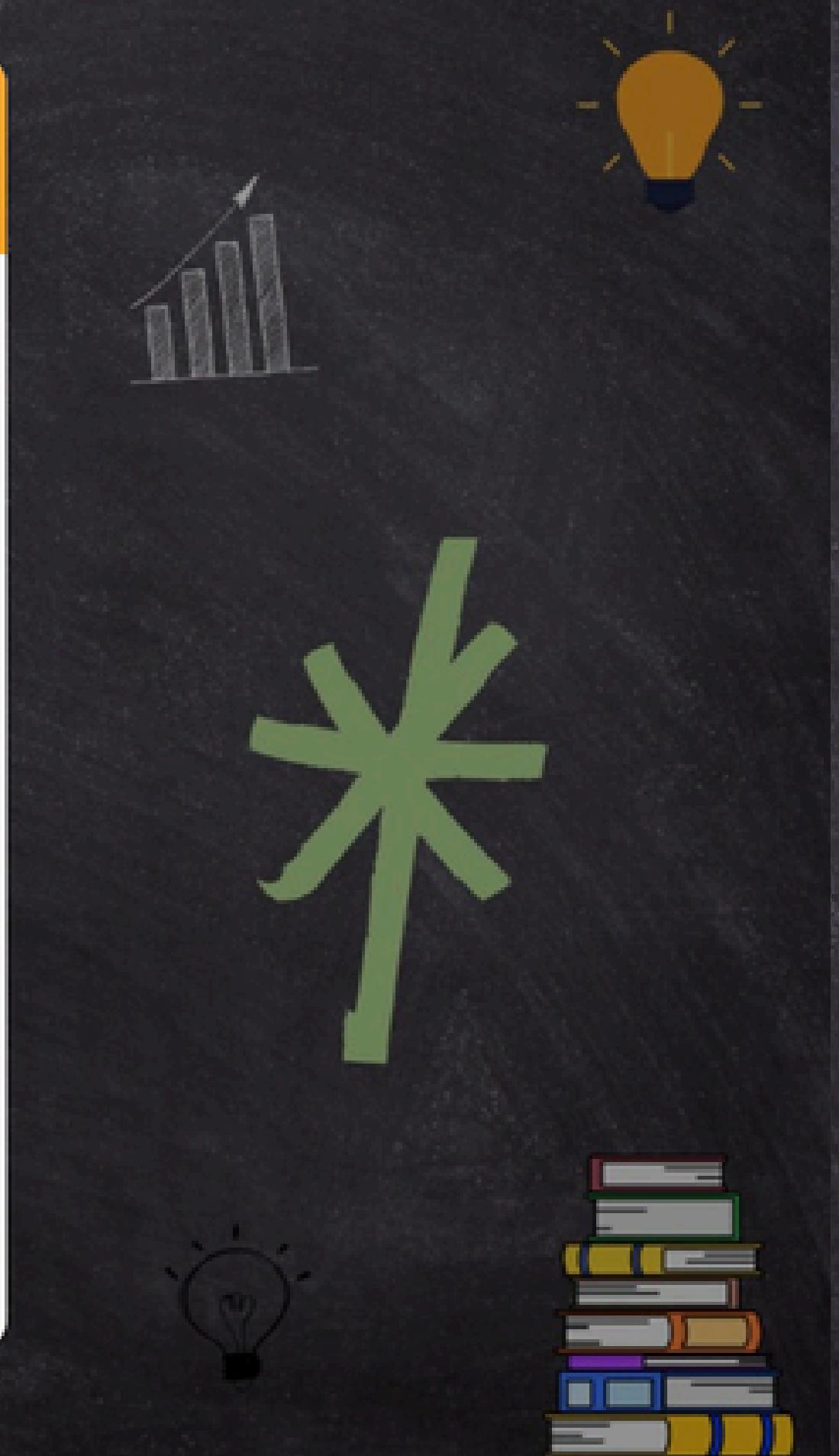
Assignments Completed  
Enter assignments (0-20)  
Range: 0 to 20 assignments

Sleep Hours  
Enter sleep hours (1-10)  
Range: 1 to 10 hours

**PREDICT MARKS**

**PREDICTED MARKS**  
**49.87 / 100**

Pass  
Congratulations! You're on track to pass.





**THANK YOU**

Github Link : <https://github.com/trish-2610/Student-Marks-Predictor>

CHECKOUT OUR ENTIRE PROJECT HERE