PREDICT STUDENT TEST SCORE

AIO Conquer
Warm Up Project Presentation

Presentated by Team CONQ013

MOTIVATION & OBJECTIVE

Motivations

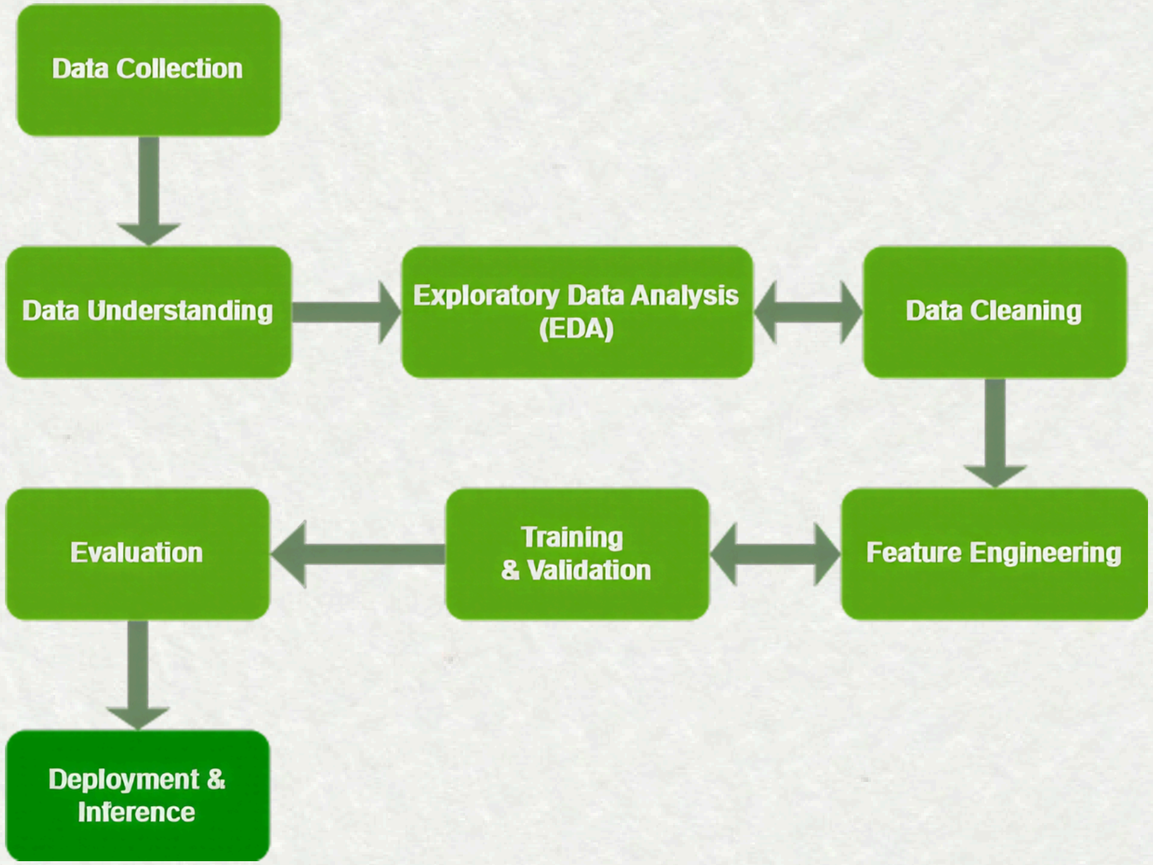
- To understand how a real ML project is built end-to-end
- To learn why each stage in the ML workflow is necessary
- Performance matters, but is not the primary goal

Objectives

- Predict student exam score based on personal & academic factors using ML
- **Input**: study habits, sleep behavior
- **Output:** Predicted exam score (0-100)
- **Real-world value:** Help educators identify at-risk students early

Machine Learning Framing

- Type: Supervised Learning
- Task: Regression



OVERALL ML WORKFLOW

The workflow is inherently iterative, with continuous feedback between data, features, and models.

DATA COLLECTION

Dataset: from Kaggle Tabular Playground Series Season 6 Episode 1: **"Predicting Student Test Scores"**

train.csv (45.98 MB)

Detail | Compact | Column

id	age	gender	course	study_hours	class_attendance	internet_access	sleep_hours	sleep_quality	study_method	facility_rating	exam_difficulty	exam_score
0 … 630k	17 … 24	other 34% male 33% Other (208310) 33%	b.tech 21% b.sc 18% Other (387210) 61%	0.08 … 7.91	40.6 … 99.4	true 579k 92% false 50.6k 8%	4.1 … 9.9	poor 34% good 34% Other (203236) 32%	coaching 21% self-study 21% Other (367172) 58%	medium 34% low 34% Other (203540) 32%	moderate 56% easy 28% Other (99478) 16%	19.6 … 100
0	21	female	b.sc	7.91	98.8	no	4.9	average	online videos	low	easy	78.3
1	18	other	diploma	4.95	94.8	yes	4.7	poor	self-study	medium	moderate	46.7
2	20	female	b.sc	4.68	92.6	yes	5.8	poor	coaching	high	moderate	99.0
3	19	male	b.sc	2.0	49.5	yes	8.3	average	group study	high	moderate	63.9
4	23	male	bca	7.65	86.9	yes	9.6	good	self-study	high	easy	100.0
5	24	male	b.com	5.04	85.1	yes	9.4	average	online videos	medium	moderate	70.1
6	20	male	b.sc	4.28	87.0	no	9.1	average	mixed	high	moderate	63.4
7	22	female	ba	4.19	44.9	yes	8.8	good	self-study	high	hard	76.8
8	22	other	b.com	1.06	98.3	yes	5.0	poor	mixed	low	moderate	46.7
9	18	male	bba	3.44	80.9	yes	6.2	good	group study	medium	easy	58.2
10	20	other	bba	7.24	51.6	yes	7.4	good	self-study	high	easy	85.9
11	19	female	b.tech	6.85	43.2	yes	8.5	poor	self-study	high	moderate	74.5
12	24	other	ba	7.4	92.4	yes	8.2	average	coaching	high	moderate	100.0
13	19	other	bba	6.99	90.5	yes	6.4	good	mixed	high	moderate	93.5
14	21	male	b.com	2.91	78.4	yes	6.7	average	online videos	low	easy	51.6
15	23	male	b.sc	7.47	64.8	yes	9.3	poor	self-study	low	moderate	76.8
16	23	male	b.tech	4.56	63.9	yes	8.4	average	mixed	high	moderate	58.6
17	21	male	bba	3.8	53.3	yes	4.7	good	coaching	high	moderate	67.4
18	17	female	b.tech	0.08	47.7	yes	9.0	average	coaching	low	moderate	32.2
19	17	male	b.tech	4.39	97.9	yes	9.1	good	group study	high	moderate	94.3
20	22	female	b.sc	6.23	92.3	yes	4.8	average	mixed	medium	moderate	77.4
21	22	female	bca	0.12	78.6	yes	4.1	poor	mixed	low	moderate	34.5
22	19	female	b.com	7.91	87.2	yes	8.2	good	online videos	high	easy	92.4

Synthetically-generated from deep learning model

DATA UNDERSTANDING

Using **df.info(), df.describe()**, and **df.head()**, we initially discovered a remarkably clean dataset:

Key observations:

- Dataset size: ~630,000 records
- Feature overview: 13 features, mix of numerical and categorical variables
- Target variable: exam_score (continuous)

- **Exam scores range widely from 19.6 to 100 points**
- **Average study time: ~4 hours daily**
- **Average sleep: ~7 hours nightly**

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 630000 entries, 0 to 629999
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   id                630000 non-null  int64
 1   age               630000 non-null  int64
 2   gender            630000 non-null  object
 3   course            630000 non-null  object
 4   study_hours       630000 non-null  float64
 5   class_attendance  630000 non-null  float64
 6   internet_access   630000 non-null  object
 7   sleep_hours       630000 non-null  float64
 8   sleep_quality     630000 non-null  object
 9   study_method      630000 non-null  object
 10  facility_rating   630000 non-null  object
 11  exam_difficulty   630000 non-null  object
 12  exam_score        630000 non-null  float64
dtypes: float64(4), int64(2), object(7)
memory usage: 62.5+ MB
```

[4]:
```
df.head()
```

[4]:

	id	age	gender	course	study_hours	class_attendance	internet_access	sleep_hours	sleep_quality	study_method	facility_rating	exam_difficulty	exam_score
0	0	21	female	b.sc	7.91	98.8	no	4.9	average	online videos	low	easy	78.3
1	1	18	other	diploma	4.95	94.8	yes	4.7	poor	self-study	medium	moderate	46.7
2	2	20	female	b.sc	4.68	92.6	yes	5.8	poor	coaching	high	moderate	99.0
3	3	19	male	b.sc	2.00	49.5	yes	8.3	average	group study	high	moderate	63.9
4	4	23	male	bca	7.65	86.9	yes	9.6	good	self-study	high	easy	100.0

:
```
df.describe()
```

:

	id	age	study_hours	class_attendance	sleep_hours	exam_score
count	630000.000000	630000.000000	630000.000000	630000.000000	630000.000000	630000.000000
mean	314999.500000	20.545821	4.002337	71.987261	7.072758	62.506672
std	181865.479132	2.260238	2.359880	17.430098	1.744811	18.916884
min	0.000000	17.000000	0.080000	40.600000	4.100000	19.599000
25%	157499.750000	19.000000	1.970000	57.000000	5.600000	48.800000
50%	314999.500000	21.000000	4.000000	72.600000	7.100000	62.600000
75%	472499.250000	23.000000	6.050000	87.200000	8.600000	76.300000
max	629999.000000	24.000000	7.910000	99.400000	9.900000	100.000000

EDA - DATA QUALITY

Missing Value:
Use df.isnull().sum() to count missing entries.

- Less than 5%? Drop rows or impute.
- Between 5-30%? Use advanced imputation (mean, median, KNN).
- Over 30%? Consider dropping the column.

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

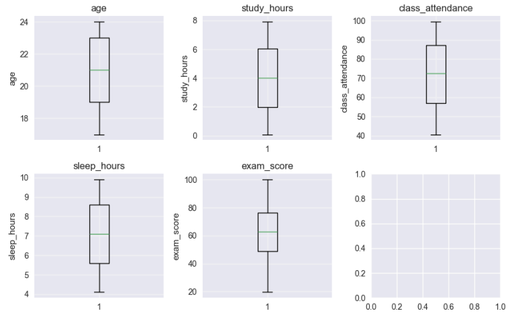
```
age                 0
gender              0
course              0
study_hours         0
class_attendance    0
internet_access     0
sleep_hours         0
sleep_quality       0
study_method        0
facility_rating     0
exam_difficulty     0
exam_score          0
dtype: int64
```

Duplicate Rows
Check with df.duplicated().sum().
Duplicates can cause overfitting and data leakage. Remove them carefully, keeping first occurrence with **keep='first'**.

```
df.duplicated().sum()
```

```
0
```

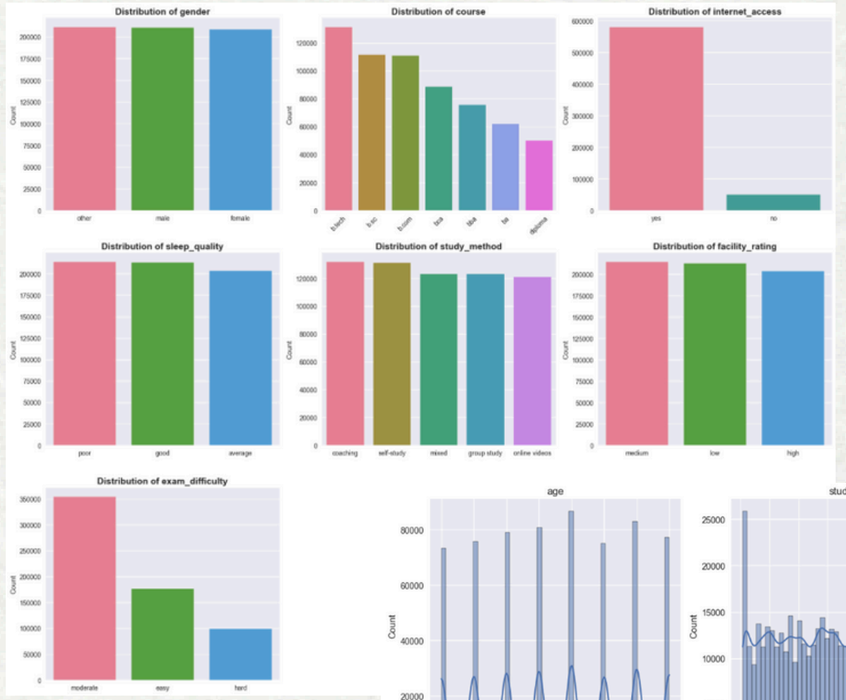
Outliers
Apply IQR method: Calculate Q1, Q3, and IQR. Outliers fall below Q1 - 1.5×IQR or above Q3 + 1.5×IQR.
Don't automatically delete—some outliers are valuable insights!

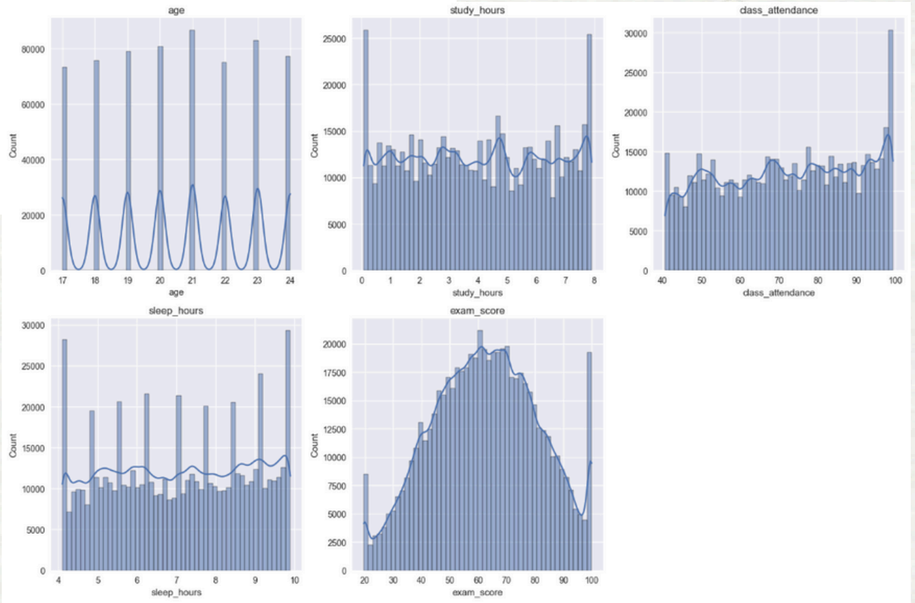


Data Logic
Verify data types match expected formats (ex: datetime, numeric vs). Check score range (0-100), sleep quality (0-24). Fix encoding issues and standardize formats.

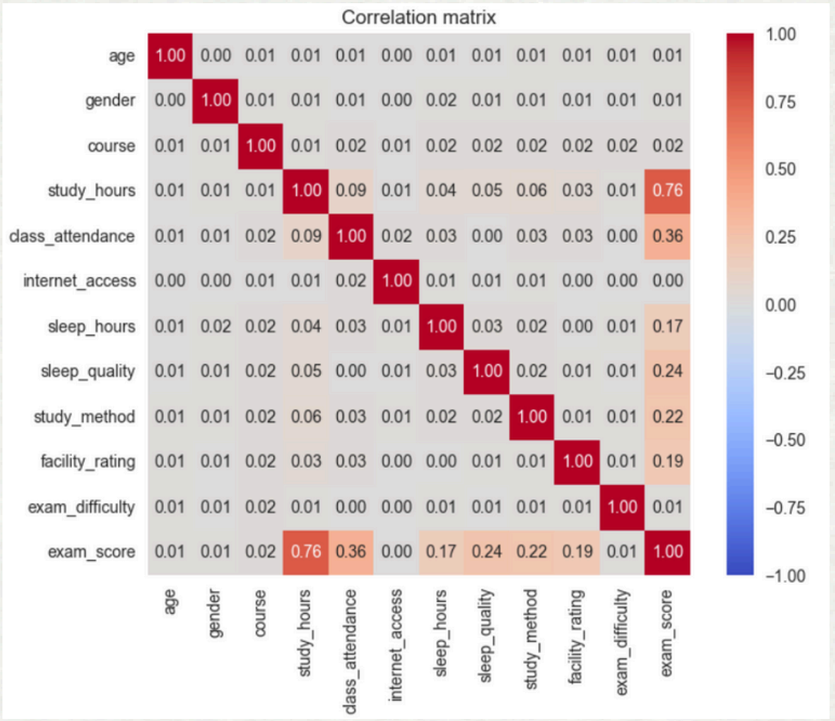
EDA - DATA DISTRIBUTION

Distribution Analysis





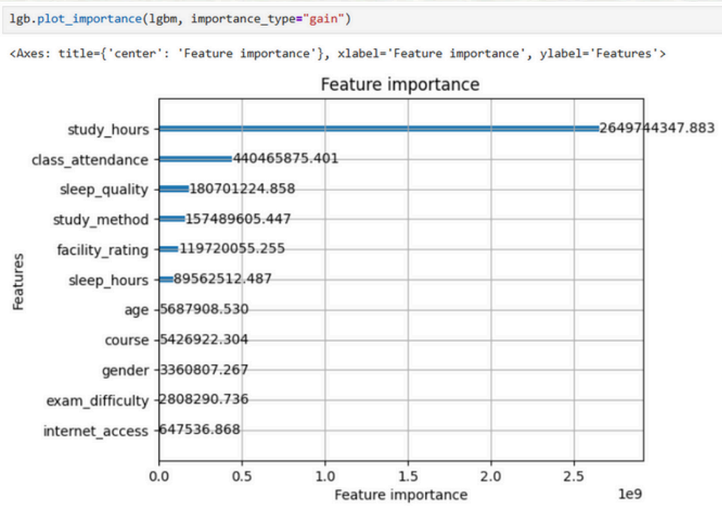
Relationship Analysis



FEATURE ENGINEERING

Remove Irrelevant Features

Drop columns like IDs with no predictive value.
Remove features with near-zero correlation (<0.05), also confirmed by tree-based model importance analysis.



Feature Encoding

Numerical features:
Used in their original scale without transformation.

Categorical features:

- Ordinal encoding for ordered variables (sleep_quality, facility_rating, exam_difficulty).
- One-hot encoding for nominal variables (study_method, course).

Create New Features

Interaction features:

- study_efficiency = study_hours × class_attendance,
- sleep_attend_product = sleep_hours × class_attendance.

Deviation feature:

- sleep_deviation = |sleep_hours − 7| to capture deviation from optimal sleep duration.

MODEL TRAINING

Model Selection Strategy

- Baseline: Linear Regression
- Regularized Models: Ridge (L2), Lasso (L1)
- Final Model: LightGBM (tree-based)

Why LightGBM?

- Captures non-linear relationships and feature interactions
- Fast training on large dataset (630K records)
- Best performance: RMSE ≈ 8.75 on validation set

Training Setup

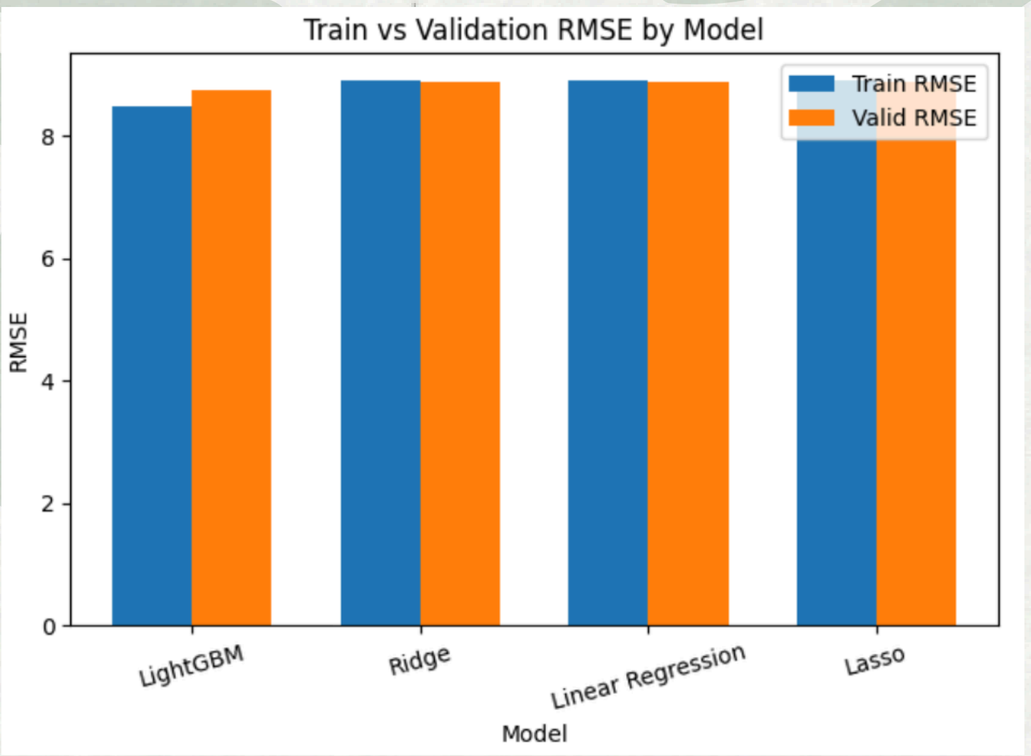
- Train/Validation split: 80/20
- Early stopping

EVALUATION

Performance Metric:

Validation RMSE *(lower is better)*

- LightGBM achieves the lowest validation RMSE, indicating better performance.
- Our model achieved RMSE ≈ 8.75, quite close to Kaggle leaderboard's 8.53—validating our EDA and feature engineering pipeline.



Predicting Student Test Scores — Late Submission

Overview | Data | Code | Models | Discussion | Leaderboard | Rules | Team | Submissions

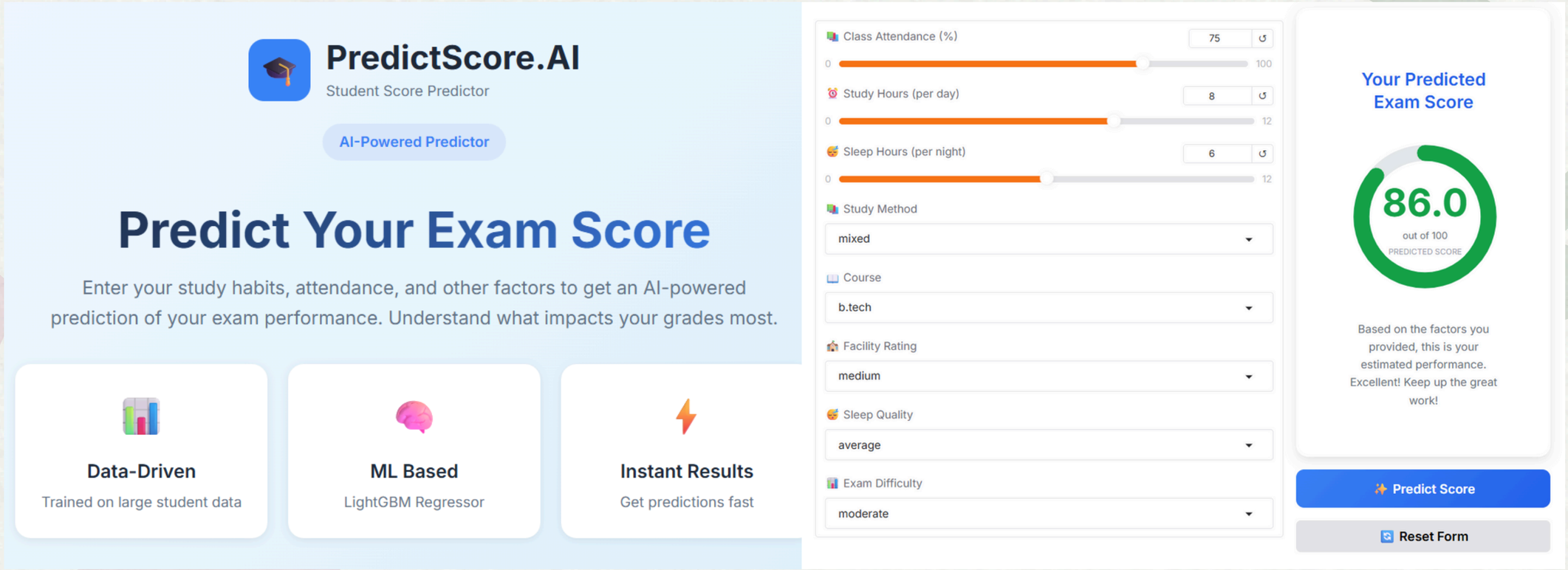
This leaderboard is calculated with approximately 20% of the test data. The final results will be based on the other 80%, so the final standings may be different.

#	Team	Members	Score	Entries	Last	Solution
1	Mahog		8.53096	51	3d	
2	Yew Jin Lim		8.53303	61	3d	
3	Traiko Dinev		8.53475	155	2d	

```
============================================================
MODEL COMPARISON SUMMARY
============================================================
            Model  Train RMSE  Valid RMSE  Valid R²  Valid MAE  Overfit Gap
         LightGBM    8.476879    8.749616  0.784742   6.972499    -0.272737
            Ridge    8.896482    8.886395  0.777960   7.093235     0.010088
Linear Regression    8.896457    8.886428  0.777958   7.093134     0.010029
            Lasso    8.898044    8.887389  0.777910   7.094680     0.010655

BEST MODEL: LightGBM
   Valid RMSE: 8.7496
   Valid R²:   0.7847
   Valid MAE:  6.9725
```

DEPLOYMENT gradio



FURRTHER IMPROVEMENT

01 Iterative EDA & Feature Engineering

- Revisit EDA after feature creation to validate new feature distributions

02 Model & Training Enhancements

- Try advanced models (e.g., Gradient Boosting, Ensemble methods)
- Perform systematic hyperparameter tuning (Grid / Random / Bayesian Search)

03 Deployment & Monitoring (Future Work)

- Monitor model performance and data drift in production
- Automate the pipeline using MLOps practices

THANK YOU FOR YOUR ATTENTION!

AIO Conquer
Warm Up Project Presentation

Presentated by Team CONQ013