OM SAI RAM

→ STUDENTS – Performance

→ Dataset used — Student Performance

→ It contains 10,000 records & 6 columns

→ linear regression model, XGBoost models used.

***STEPS OF CODE**

① importing the libaries. like numpy, Pandas, matplot, seaborn.

→ numpy & Pandas are used for numerical & data operations.

→ seaborn & matplot are used for visualization.

② Ignore warnings.

```
import warnings
warnings.filterwarning('ignore')
```

③ Loading the dataset.

```
df = pd.read_csv("drive path")
df.head(5)
∴ head = first five rows.
```

④ summarizing the dataset

```
df .describe()
df.info()
```

→ provides summary including mean & standard deviation.

→ shows dataset structure, datatypes & missing values

5 identifying & remove duplicate record

```
df.duplicated.sum()
dF = df.drop_duplicates()
```

-> method checks for dupiicate rows in dataset

-> counts total number of duplicate rows in dataset [.sum()].

-> df.drop_duplicates_removes duplicate rows from the dataset.

5 we use Boolean series because to change binary into numerical format.{true/false}

```
df['extracurricular Activities'] = (df('extracuriccular Activities']
== 'yes'). astype(int)
```

*-> astype(int) = which converts into integer types (0, 1)

7 Linear regression model

~~impor~~

```
from sklearn.model_selection import train_test_split
from sklearn.linearmodel import linear regression.
```

-> importing the models selection to train & test.

8) will drop the required output & will train a test the data using linear regression model

X_train, y_train → used to train the model.

x_test, y_test → used to test the model

9) Now, will print the linear regression score of the tested model

→ XGBOOST model

10) will import required libaries.

11) Now will be encoding the data because it converts text into numbers.

```
label_encoder = Labelencoder()
```

12) will split the data which we need as a output.

13) If encoded is needed then will do, if not will train the data & test the data.

14) will install xgboost.

```
!pipinstall --upgrade scikit-learn xgboost
```

(15) [illegible] model

(16) [illegible] hyper parameters, [illegible] the data into 5 parts, & [illegible]

17) we use GridSearchCV to find the best parameters.

18) Trains the best model using xgboost

19) Now will test the model (calculate mean absolute error (MAE) to get accuracy

(20) Now print the outputs of trained, tested & MAE test on datas.

*> Here question is to get the best model & best students performance

→ code

↓

```
bestperformance = df['performance index'].max()
best_student = df[df['performance index'] == best
                                      performance]

#print results

print("best students performance index:", best
                                      performance)

print (best_student).
```

* Keypoints to rember.

1. It contains student performance & 10,000 records.
2. we use linear regression & XGBoost Model
3. we conclude the best model is XGBoost & the best student is the student whose performance index is 100(max). ✓