Model Development Phase Template

Date	10 July 2024
Team ID	739923
Project Title	Predicting the Compressive Strength of Concrete
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

Paste the screenshot of the model training code

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

```
from sklearn.linear_model import LinearRegression
```

```
lr = LinearRegression()
lr.fit(x_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
lr.score(x_train,y_train)
lr.score(x_test,y_test)
```

```
from sklearn.ensemble import GradientBoostingRegressor
```

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingRegressor
cols = df.columns.drop('Concrete compressive strength')
x = df[cols]
y = df['Concrete compressive strength']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

# Initialize the GradientBoostingRegressor
gbr_model = GradientBoostingRegressor()  # Use a different variable name
gbr_model.fit(x_train, y_train)
```

```
▾ GradientBoostingRegressor
GradientBoostingRegressor()
```

Model Validation and Evaluation Report:

Model	Training Report	Accuracy	Metrix

Linear Regression	`x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)`
	`from sklearn.linear_model import LinearRegression`
	`lr = LinearRegression()`
	`lr.fit(x_train, y_train)`
	`▾ LinearRegression`
	`LinearRegression()`
	`lr.score(x_train,y_train)`
	`lr.score(x_test,y_test)`
		0.74	`from sklearn.linear_model import LinearRegression`
			`lr=LinearRegression()`
			`lr.fit(x_train,y_train)`
			`mse=mean_squared_error(ypred,y_test)`
			`print("mean squared error:",mse)`
			`r2_lr=r2_score(ypred,y_test)`
			`print("r2 score",r2_lr)`
			`mean squared error: 95.97548435337708`
			`r2 score 0.42303938808034913`

Gradient Boosting Regressor	`from sklearn.ensemble import GradientBoostingRegressor`
	`from sklearn.model_selection import train_test_split`
	`from sklearn.ensemble import GradientBoostingRegressor`
	`cols = df.columns.drop('Concrete compressive strength')`
	`x = df[cols]`
	`y = df['Concrete compressive strength']`
	`x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)`
	`# Initialize the GradientBoostingRegressor`
	`gbr_model = GradientBoostingRegressor()  # Use a different variable name`
	`gbr_model.fit(x_train, y_train)`
	`▾ GradientBoostingRegressor`
	`GradientBoostingRegressor()`
		0.88	`print("MAE",mean_absolute_error(y_test,y_pred))`
			`print("MSE",mean_squared_error(y_test,y_pred))`
			`print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))`
			`MAE 3.7624256501261932`
			`MSE 28.721713317320482`
			`RMSE 5.359264251492035`