



Model Development Phase Template

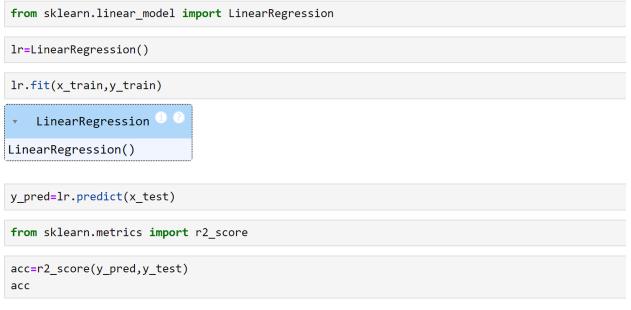
Date	12 July 2024	
Team ID	SWTID1720160264	
Project Title	Predicting Compressive Strength Of Concrete Using Machine Learning	
Maximum Marks	4 Marks	

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 Mean squared error,RMSE and R2 score for multiple models, presented through respective screenshots.

Initial Model Training Code:

1.Importing and Building Linear Regression model



0.7145646613461942





2. Importing and Building Ridge and Lasso Regression model



3.Importing and Building GradientBoostingRegressor

```
from sklearn.ensemble import GradientBoostingRegressor

grb=GradientBoostingRegressor()
grb.fit(x_train,y_train)

v GradientBoostingRegressor v CoradientBoostingRegressor()

grb-GradientBoostingRegressor v CoradientBoostingRegressor()

y_pred=grb.predict(x_test)

score=r2_score(y_pred,y_test)
```





4. Importing and Building XGBRegressor

```
import xgboost
xgb=xgboost.XGBRegressor()
xgb.fit(x_train,y_train)
                                  XGBRegressor
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable categorical=False, eval metric=None, feature types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)
y_pred=xgb.predict(x_test)
score=r2_score(y_pred,y_test)
score
```

5. Importing and Building RandomForestRegressor

```
from sklearn.ensemble import RandomForestRegressor

rf=RandomForestRegressor(criterion='squared_error',random_state=0)

rf.fit(x_train,y_train)
pred=rf.predict(x_test)

score=r2_score(y_test,pred)
print('R2 score of Random Forest Regression:',score*100)
```





Model	Mean Squared Error	R2 Score	RMSE
Linear Regression	print("MSE: ",metrics.mean_squared_error(y_pred,y_test)) MSE: 62.467942036660126	71.4	<pre>print("RMSE: ",np.sqrt(metrics.mean_squared_error(y_pred,y_test))) RMSE: 7.90366366735132</pre>
Ridge Regression	print("MSE: ",metrics.mean_squared_error(pred1,y_test)) MSE: 62.46780180902786	71.45	<pre>print("RMSE: ",np.sqrt(metrics.mean_squared_error(pred1,y_test))) RMSE: 7.903657495680583</pre>
Lasso Regression	print("MSE: ",metrics.mean_squared_error(pred2,y_test)) MSE: 62.085175545855364	71.36	print("RMSE: ",np.sqrt(metrics.mean_squared_error(pred2,y_test))) RMSE: 7.879414670256628
Gradient boosting Regressor	<pre>print("MSE: ",metrics.mean_squared_error(y_pred,y_test)) MSE: 27.05593121168309</pre>	88.80	print("RMSE: ",np.sqrt(metrics.mean_squared_error(y_pred,y_test))) RMSE: 5.201531621713271
XGBoost Regressor	print("MSE: ",metrics.mean_squared_error(y_pred,y_test)) MSE: 20.446855694652488	93.11	<pre>print("RMSE: ",np.sqrt(metrics.mean_squared_error(y_pred,y_test))) RMSE: 4.521819953807591</pre>
Random Forest Regressor	print("MSE: ",metrics.mean_squared_error(pred,y_test)) MSE: 26.496806978116595	91.08	<pre>print("RMSE: ",np.sqrt(metrics.mean_squared_error(pred,y_test))) RMSE: 5.147504927449472</pre>