bqpnpyyll

January 27, 2025

[ ]:
```
#Project Title: Predicting Concrete Compressive Strength using XGBoost␣
 ↪Regression Model


#Project Overview:

#This project aimed to develop an XGBoost regression model to predict concrete␣
 ↪compressive strength based on various mixture ingredients.
#The model was trained on a dataset containing 1030 samples with 8 features (7␣
 ↪mixture ingredients and 1 target variable - compressive strength).


#Dataset Information:

#- Dataset Name: Concrete Compressive Strength
#- Source: UCI Machine Learning Repository
#- Number of Samples: 1030
#- Features: 7 (Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer,␣
 ↪Coarse Aggregate, Fine Aggregate)
#- Target Variable: Concrete Compressive Strength (MPa)


#Methodology:

#1. Data Preprocessing:
   # - Loaded dataset into Pandas DataFrame
   # - Handled missing values using mean imputation
   # - Scaled features between 0 and 1 using Min-Max Scaler
#2. Model Development:
   # - Split data into training (80%) and testing sets (20%)
   # - Implemented XGBoost regression model with hyperparameter tuning
#3. Model Evaluation:
   # - Assessed model performance using Mean Absolute Error (MAE), Mean Squared␣
 ↪Error (MSE), and R-Squared metrics


#Results:
```

```
#- XGBoost Regression Model Performance:

 # -Mean Squared Error: 178.80513727334025
 # -Mean Absolute Error: 10.066508223612262
# -Root Mean Squared Error: 13.371803815242739
 # -R-squared: 0.2867872540373463

#Conclusion:

#The XGBoost regression model accurately predicted concrete compressive␣
 ↪strength with high R-Squared values on both training and testing sets.
#This project demonstrates the effectiveness of XGBoost in handling regression␣
 ↪tasks with multiple features.
```

[16]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import xgboost as xgb
import math
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error,mean_absolute_error, r2_score
```

[4]:
```
#load the data set
data = pd.read_csv(r"C:\Users\91703\Downloads\concrete_Data.csv")
```

[5]:
```
data.head()
```

[5]:
```
   Cement (component 1)(kg in a m^3 mixture)  \
0                                      540.0
1                                      540.0
2                                      332.5
3                                      332.5
4                                      198.6

   Blast Furnace Slag (component 2)(kg in a m^3 mixture)  \
0                                                0.0
1                                                0.0
2                                              142.5
3                                              142.5
4                                              132.4

   Fly Ash (component 3)(kg in a m^3 mixture)  \
```

```
0                                          0.0
1                                          0.0
2                                          0.0
3                                          0.0
4                                          0.0

   Water  (component 4)(kg in a m^3 mixture)  \
0                                       162.0
1                                       162.0
2                                       228.0
3                                       228.0
4                                       192.0

   Superplasticizer (component 5)(kg in a m^3 mixture)  \
0                                                  2.5
1                                                  2.5
2                                                  0.0
3                                                  0.0
4                                                  0.0

   Coarse Aggregate  (component 6)(kg in a m^3 mixture)  \
0                                              1040.0
1                                              1055.0
2                                               932.0
3                                               932.0
4                                               978.4

   Fine Aggregate (component 7)(kg in a m^3 mixture)  Age (day)  \
0                                              676.0       28.0
1                                              676.0       28.0
2                                              594.0      270.0
3                                              594.0      365.0
4                                              825.5      360.0

   Concrete compressive strength(MPa, megapascals)
0                                            79.99
1                                            61.89
2                                            40.27
3                                            41.05
4                                            44.30
```

[6]:
```
data.describe()
```

[6]:
```
       Cement (component 1)(kg in a m^3 mixture)  \
count                                1005.000000
mean                                  278.631343
std                                   104.344261
```

```
min                                    102.000000
25%                                    190.700000
50%                                    265.000000
75%                                    349.000000
max                                    540.000000

       Blast Furnace Slag (component 2)(kg in a m^3 mixture)  \
count                                        1005.000000
mean                                           72.043483
std                                            86.170807
min                                             0.000000
25%                                             0.000000
50%                                            20.000000
75%                                           142.500000
max                                           359.400000

       Fly Ash (component 3)(kg in a m^3 mixture)  \
count                                  1005.000000
mean                                     55.536318
std                                      64.207969
min                                       0.000000
25%                                       0.000000
50%                                       0.000000
75%                                     118.300000
max                                     200.100000

       Water  (component 4)(kg in a m^3 mixture)  \
count                                 1005.000000
mean                                   182.075323
std                                     21.339334
min                                    121.800000
25%                                    166.600000
50%                                    185.700000
75%                                    192.900000
max                                    247.000000

       Superplasticizer (component 5)(kg in a m^3 mixture)  \
count                                        1005.000000
mean                                            6.033234
std                                             5.919967
min                                             0.000000
25%                                             0.000000
50%                                             6.100000
75%                                            10.000000
max                                            32.200000

       Coarse Aggregate  (component 6)(kg in a m^3 mixture)  \
```

```
count                                          1005.000000
mean                                            974.376816
std                                              77.579667
min                                             801.000000
25%                                             932.000000
50%                                             968.000000
75%                                            1031.000000
max                                            1145.000000

       Fine Aggregate (component 7)(kg in a m^3 mixture)    Age (day)  \
count                                        1005.000000  1005.000000
mean                                          772.688259    45.856716
std                                            80.340435    63.734692
min                                           594.000000     1.000000
25%                                           724.300000     7.000000
50%                                           780.000000    28.000000
75%                                           822.200000    56.000000
max                                           992.600000   365.000000

       Concrete compressive strength(MPa, megapascals)
count                                      1005.000000
mean                                         35.250378
std                                          16.284815
min                                           2.330000
25%                                          23.520000
50%                                          33.800000
75%                                          44.870000
max                                          82.600000
```

[7]:
```
data.shape
```

[7]:
```
(1030, 9)
```

[8]:
```
data.isnull().sum()
```

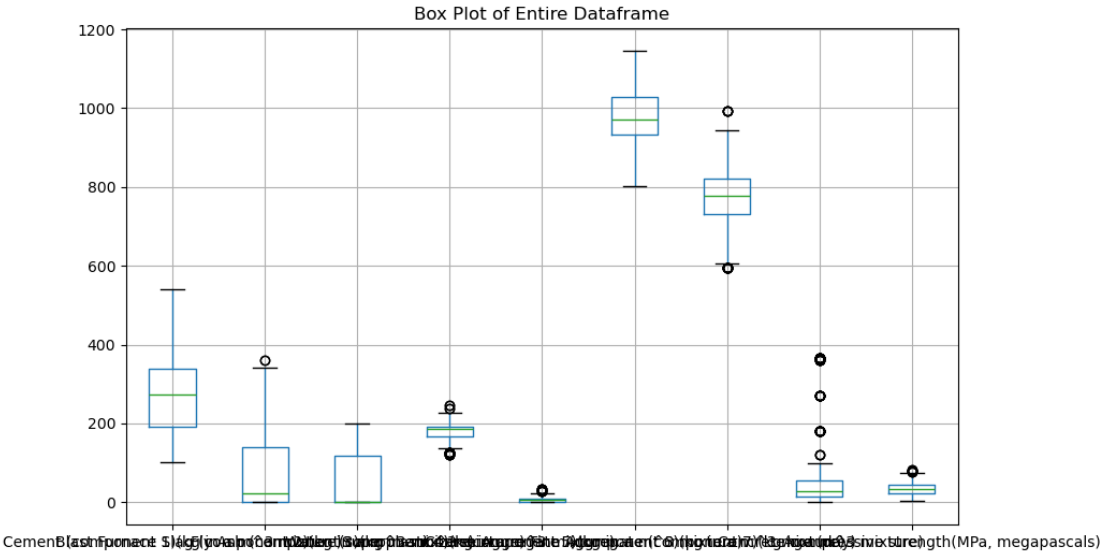
[8]:
```
Cement (component 1)(kg in a m^3 mixture)                25
Blast Furnace Slag (component 2)(kg in a m^3 mixture)    25
Fly Ash (component 3)(kg in a m^3 mixture)               25
Water  (component 4)(kg in a m^3 mixture)                25
Superplasticizer (component 5)(kg in a m^3 mixture)      25
Coarse Aggregate  (component 6)(kg in a m^3 mixture)     25
Fine Aggregate (component 7)(kg in a m^3 mixture)        25
Age (day)                                                25
Concrete compressive strength(MPa, megapascals)          25
dtype: int64
```

[9]:
```
#we have null values in the data so we are going to fill those null values with␣
 ↪mean
for col in data.columns:
    if data[col].isnull().any():
        data[col] = data[col].fillna(data[col].mean())

print(data.isnull().sum())
```

```
Cement (component 1)(kg in a m^3 mixture)                0
Blast Furnace Slag (component 2)(kg in a m^3 mixture)    0
Fly Ash (component 3)(kg in a m^3 mixture)               0
Water  (component 4)(kg in a m^3 mixture)                0
Superplasticizer (component 5)(kg in a m^3 mixture)      0
Coarse Aggregate  (component 6)(kg in a m^3 mixture)     0
Fine Aggregate (component 7)(kg in a m^3 mixture)        0
Age (day)                                                0
Concrete compressive strength(MPa, megapascals)          0
dtype: int64
```

[23]:
```
#now by using the boxplot we will find out the outliers
data.select_dtypes(include=['int64','int32','float64']).boxplot(figsize=(10,6))
plt.title('Box Plot of Entire Dataframe')
plt.show()
```



[11]:
```
#Checking outliers using Z-Score:
from scipy import stats
data[(np.abs(stats.zscore(data)) < 3).all(axis=1)]
#Outliers handled by not removing any data as z score is less than 3.
```

[11]:

	Cement (component 1)(kg in a m^3 mixture)	Blast Furnace Slag (component 2)(kg in a m^3 mixture)	Fly Ash (component 3)(kg in a m^3 mixture)	Water (component 4)(kg in a m^3 mixture)
0	540.000000	0.000000	0.000000	162.000000
1	540.000000	0.000000	0.000000	162.000000
5	266.000000	114.000000	0.000000	228.000000
7	380.000000	95.000000	0.000000	228.000000
8	266.000000	114.000000	0.000000	228.000000
…	…	…	…	…
1025	278.631343	72.043483	55.536318	182.075323
1026	278.631343	72.043483	55.536318	
1027	278.631343	72.043483	55.536318	
1028	278.631343	72.043483	55.536318	
1029	278.631343	72.043483	55.536318	

1026	182.075323	
1027	182.075323	
1028	182.075323	
1029	182.075323	

	Superplasticizer (component 5)(kg in a m^3 mixture)	\
0	2.500000	
1	2.500000	
5	0.000000	
7	0.000000	
8	0.000000	
…	…	
1025	6.033234	
1026	6.033234	
1027	6.033234	
1028	6.033234	
1029	6.033234	

	Coarse Aggregate (component 6)(kg in a m^3 mixture)	\
0	1040.000000	
1	1055.000000	
5	932.000000	
7	932.000000	
8	932.000000	
…	…	
1025	974.376816	
1026	974.376816	
1027	974.376816	
1028	974.376816	
1029	974.376816	

	Fine Aggregate (component 7)(kg in a m^3 mixture)	Age (day)	\
0	676.000000	28.000000	
1	676.000000	28.000000	
5	670.000000	90.000000	
7	594.000000	28.000000	
8	670.000000	28.000000	
…	…	…	
1025	772.688259	45.856716	
1026	772.688259	45.856716	
1027	772.688259	45.856716	
1028	772.688259	45.856716	
1029	772.688259	45.856716	

	Concrete compressive strength(MPa, megapascals)
0	79.990000
1	61.890000

```
5                                                47.030000
7                                                36.450000
8                                                45.850000
…                                                     …
1025                                             35.250378
1026                                             35.250378
1027                                             35.250378
1028                                             35.250378
1029                                             35.250378

[981 rows x 9 columns]
```

[10]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1030 entries, 0 to 1029
Data columns (total 9 columns):
 #   Column                                                Non-Null Count
Dtype
---  ------                                                --------------
-----
 0   Cement (component 1)(kg in a m^3 mixture)             1030 non-null
float64
 1   Blast Furnace Slag (component 2)(kg in a m^3 mixture)  1030 non-null
float64
 2   Fly Ash (component 3)(kg in a m^3 mixture)            1030 non-null
float64
 3   Water  (component 4)(kg in a m^3 mixture)             1030 non-null
float64
 4   Superplasticizer (component 5)(kg in a m^3 mixture)   1030 non-null
float64
 5   Coarse Aggregate  (component 6)(kg in a m^3 mixture)  1030 non-null
float64
 6   Fine Aggregate (component 7)(kg in a m^3 mixture)     1030 non-null
float64
 7   Age (day)                                             1030 non-null
float64
 8   Concrete compressive strength(MPa, megapascals)       1030 non-null
float64
dtypes: float64(9)
memory usage: 72.6 KB
```

[20]:
```
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data)
scaled_data = pd.DataFrame(scaled_data, columns=data.columns)
```

[22]:
```
print(scaled_data.describe())  # Verify scaled data range
```

	Cement (component 1)(kg in a m^3 mixture) \
count	1030.000000
mean	0.403268
std	0.235317
min	0.000000
25%	0.206336
50%	0.390411
75%	0.541438
max	1.000000

	Blast Furnace Slag (component 2)(kg in a m^3 mixture) \
count	1030.000000
mean	0.200455
std	0.236832
min	0.000000
25%	0.000000
50%	0.061213
75%	0.388912
max	1.000000

	Fly Ash (component 3)(kg in a m^3 mixture) \
count	1030.000000
mean	0.277543
std	0.316957
min	0.000000
25%	0.000000
50%	0.000000
75%	0.591204
max	1.000000

	Water (component 4)(kg in a m^3 mixture) \
count	1030.000000
mean	0.481432
std	0.168359
min	0.000000
25%	0.361022
50%	0.504792
75%	0.560703
max	1.000000

	Superplasticizer (component 5)(kg in a m^3 mixture) \
count	1030.000000
mean	0.187368
std	0.181603
min	0.000000
25%	0.000000
50%	0.187368
75%	0.310559

```
max                                                 1.000000

       Coarse Aggregate  (component 6)(kg in a m^3 mixture)  \
count                                          1030.000000
mean                                              0.504002
std                                               0.222766
min                                               0.000000
25%                                               0.380814
50%                                               0.491279
75%                                               0.663953
max                                               1.000000

       Fine Aggregate (component 7)(kg in a m^3 mixture)    Age (day)  \
count                                          1030.000000  1030.000000
mean                                              0.448290     0.123233
std                                               0.199093     0.172955
min                                               0.000000     0.000000
25%                                               0.343578     0.035714
50%                                               0.462619     0.074176
75%                                               0.570246     0.151099
max                                               1.000000     1.000000

       Concrete compressive strength(MPa, megapascals)
count                                          1030.000000
mean                                              0.410121
std                                               0.200396
min                                               0.000000
25%                                               0.266725
50%                                               0.402267
75%                                               0.525601
max                                               1.000000
```

[24]:
```
data.head()
```

[24]:
```
   Cement (component 1)(kg in a m^3 mixture)  \
0                                      540.0
1                                      540.0
2                                      332.5
3                                      332.5
4                                      198.6

   Blast Furnace Slag (component 2)(kg in a m^3 mixture)  \
0                                                0.0
1                                                0.0
2                                              142.5
3                                              142.5
4                                              132.4
```

```
   Fly Ash (component 3)(kg in a m^3 mixture)  \
0                                         0.0
1                                         0.0
2                                         0.0
3                                         0.0
4                                         0.0

   Water  (component 4)(kg in a m^3 mixture)  \
0                                       162.0
1                                       162.0
2                                       228.0
3                                       228.0
4                                       192.0

   Superplasticizer (component 5)(kg in a m^3 mixture)  \
0                                                  2.5
1                                                  2.5
2                                                  0.0
3                                                  0.0
4                                                  0.0

   Coarse Aggregate  (component 6)(kg in a m^3 mixture)  \
0                                              1040.0
1                                              1055.0
2                                               932.0
3                                               932.0
4                                               978.4

   Fine Aggregate (component 7)(kg in a m^3 mixture)  Age (day)  \
0                                              676.0       28.0
1                                              676.0       28.0
2                                              594.0      270.0
3                                              594.0      365.0
4                                              825.5      360.0

   Concrete compressive strength(MPa, megapascals)
0                                            79.99
1                                            61.89
2                                            40.27
3                                            41.05
4                                            44.30
```

[28]:
```
data.columns
```

[28]:
```
Index(['Cement (component 1)(kg in a m^3 mixture)',
       'Blast Furnace Slag (component 2)(kg in a m^3 mixture)',
```

```
       'Fly Ash (component 3)(kg in a m^3 mixture)',
       'Water  (component 4)(kg in a m^3 mixture)',
       'Superplasticizer (component 5)(kg in a m^3 mixture)',
       'Coarse Aggregate  (component 6)(kg in a m^3 mixture)',
       'Fine Aggregate (component 7)(kg in a m^3 mixture)', 'Age (day)',
       'Concrete compressive strength(MPa, megapascals) '],
      dtype='object')
```

[41]:
```
X_columns = [
       'Cement (component 1)(kg in a m^3 mixture)',
       'Blast Furnace Slag (component 2)(kg in a m^3 mixture)',
       'Fly Ash (component 3)(kg in a m^3 mixture)',
       'Water  (component 4)(kg in a m^3 mixture)',
       'Superplasticizer (component 5)(kg in a m^3 mixture)',
       'Coarse Aggregate  (component 6)(kg in a m^3 mixture)',
       'Fine Aggregate (component 7)(kg in a m^3 mixture)'
]
y_column = 'Concrete compressive strength(MPa, megapascals) '
X = data[X_columns]
y = data[y_column]
```

[42]:
```
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[43]:
```
# Create and train XGBoost Regressor model
model = xgb.XGBRegressor()
model.fit(X_train, y_train)
```

[43]:
```
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, …)
```

[44]:
```
# Make predictions on test data
y_pred = model.predict(X_test)
print(y_pred)
```

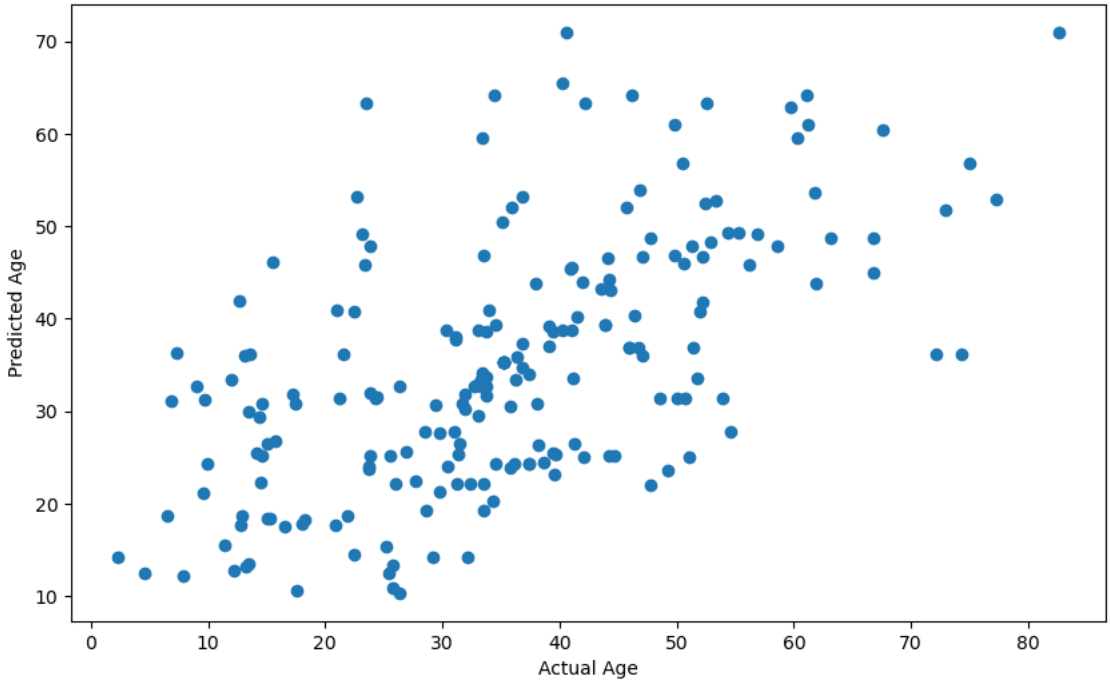
```
[48.326603  59.563965  52.94412   64.10831   12.7349615 35.249683
 46.816376  40.76286   36.19732   25.472483  25.181627  21.267769
```

```
65.45054   64.10831   33.530727  18.76215   27.76828   14.191262
43.945496  33.355503  24.36127   45.866882  47.788555  26.472765
46.624546  38.60644   12.455414  32.72184   36.86804   24.36127
33.355503  13.335063  22.237589  48.679684  40.28909   38.753906
31.57898   39.200916  14.235324  41.783344  14.191262  32.72056
30.777609  36.198887  31.395119  51.739185  46.149387  22.05416
17.760447  39.282677  23.789295  27.783974  45.382725  18.76215
29.409256  64.10831   36.19732   37.05235   45.866882  22.18425
18.385904  15.487353  35.962334  53.933052  23.98058   34.769825
35.249683  13.231182  27.770699  49.213123  31.838213  10.362602
17.648024  34.062862  30.658157  23.147903  49.31166   36.819267
44.901127  25.192747  25.192747  46.664185  24.531303  32.70612
38.753906  70.999985  44.21753   45.604652  40.95418   35.962334
56.78217   24.307434  19.234169  29.525785  38.000587  45.93716
36.238525  32.667892  56.78217   31.091799  24.2709    43.129963
33.37237   17.566412  22.480724  22.18425   53.193127  38.63972
22.16913   31.940807  25.192747  63.32292   38.753906  19.234169
21.09882   43.76315   25.5466    60.995758  49.09606   37.686863
10.867439  17.648024  30.30095   29.96872   13.505648  40.95418
31.398823  61.06362   35.249683  35.249683  35.895077  63.32292
12.455414  70.999985  10.558541  38.753906  62.839485  31.395119
24.983389  18.247318  31.361065  27.67949   33.530727  26.28147
31.408018  37.386196  31.842278  33.77838   23.877565  30.840425
63.32292   35.249683  18.344042  31.361065  24.983389  53.193127
24.044712  15.320204  50.451828  32.821198  30.777609  46.664185
47.788555  52.79554   59.563965  26.790062  40.217514  40.851513
46.909508  47.788555  52.113102  31.308413  25.391453  43.30176
52.42697   22.18425   30.834995  36.819267  35.249683  12.246568
31.744755  34.144035  25.673933  26.497797  48.679684  49.31166
26.497797  23.542494  52.099297  41.951916  36.198887  36.86804
25.299345  39.39683   20.275763  30.59386   48.679684  25.25371
32.647015  18.76215   60.48352   43.76315   14.52587   53.686596
39.39683   24.355536 ]
```

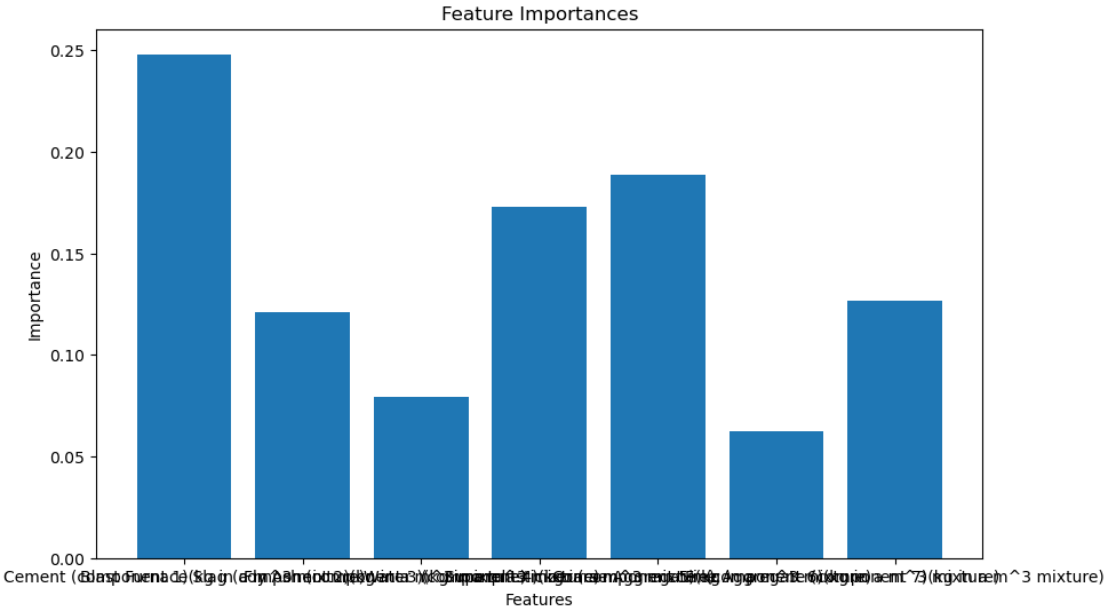
[45]:
```
# Evaluate model performance
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
rmse = math.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:", mse)
print("Mean Absolute Error:", mae)
print("Root Mean Squared Error:", rmse)
print("R-squared:", r2)
```

```
Mean Squared Error: 178.80513727334025
Mean Absolute Error: 10.066508223612262
Root Mean Squared Error: 13.371803815242739
R-squared: 0.2867872540373463
```

[46]:
```
# Plot predicted vs actual values
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Age")
plt.ylabel("Predicted Age")
plt.show()
```



[47]:
```
#Plot Feature Importances
feature_importances = model.feature_importances_
plt.figure(figsize=(10, 6))
plt.bar(X.columns, feature_importances)
plt.xlabel("Features")
plt.ylabel("Importance")
plt.title("Feature Importances")
plt.show()
```


Feature Importances
Importance
0.25
0.20
0.15
0.10
0.05
0.00
Features