In [1]:
```
import pandas as pd
import numpy as np
import scipy as sp
```

Import Data

In [2]:
```
train_df = pd.read_csv('train.csv')
test_df = pd.read_csv('test.csv')
```

In [ ]:
```
train_df
```

In [ ]:
```
test_df
```

In [ ]:
```
train_df.describe
```

In [ ]:
```
train_df.info(10)
```

Label Encoding

In [7]:
```
from sklearn.preprocessing import LabelEncoder
```

In [8]:
```
le = LabelEncoder()

test_ids = test_df['ID']

train_df = train_df.drop(columns=['ID'])
test_df = test_df.drop(columns=['ID'])

for col in train_df.columns:
    if train_df[col].dtype == 'object':
        train_df[col] = le.fit_transform(train_df[col])

for col in test_df.columns:
    if test_df[col].dtype == 'object':
        test_df[col] = le.fit_transform(test_df[col])
```

In [ ]:
```
train_df.describe
```

In [ ]:
```
test_df.describe
```

Remove Zero Variance Features

In [11]:
```
from sklearn.feature_selection import VarianceThreshold
```

In [12]:
```
selector = VarianceThreshold(threshold=0)  # Remove features with zero variance

X_train = train_df.drop(columns=['y'])
y_train = train_df['y']
```

```
X_train_reduced = selector.fit_transform(X_train)
test_df_reduced = selector.transform(test_df)
```

Dimensionality Reduction

In [13]:
```
from sklearn.decomposition import PCA
```

In [14]:
```
pca = PCA(n_components=0.95)

X_train_pca = pca.fit_transform(X_train_reduced)
test_df_pca = pca.fit_transform(test_df_reduced)
```

In [ ]:
```
X_train_pca.shape, test_df_pca.shape
```

Train Model

In [16]:
```
from xgboost import XGBRegressor
```

In [17]:
```
xg = XGBRegressor()
```

In [ ]:
```
xg.fit(X_train_pca, y_train)
```

Predictions

In [19]:
```
test_pred = xg.predict(test_df_pca)
```

In [20]:
```
submission = pd.DataFrame({'Id': test_ids, 'Predicted_Test_Bench_Time': test_pred})
submission.to_csv('submission.csv', index=False)
```