1. Final Pipeline with best saved models

1.1 Library Imports

```
In [1]: import pandas as pd
        import joblib
        import random
        import xgboost as xgb
        import pickle
        from sklearn.metrics import r2_score
        import warnings
        warnings.filterwarnings("ignore")
```

1.2 Get one random data point prom the test dataset

```
In [2]: data=pd.read_csv("train.csv")
```

```
In [3]: query = data.loc[:10]
        # query.loc[0:10] = test.loc[datapoint]
        X = query.drop('y', axis=1)
        y = query['y']
```

1.3 Preprocessing function

In [4]:

```
def preprocess_categorical(data, IDs):
    """
    data : pandas dataframe
    IDs: ID feature
    return: dataframe

    This function takes the dataframe as input,
    encodes the categorical features.
    """
    # create empty lists for collecting feature names
    cat_features = ['X0','X1','X2','X3','X4','X5','X6','X8']
    Binary_features = ['X10', 'X11', 'X12', 'X13', 'X14', 'X15', 'X16', 'X17', 'X18', 'X19', 'X20', 'X21', 'X22', 'X2
3',
                       'X24', 'X26', 'X27', 'X28', 'X29', 'X30', 'X31', 'X32', 'X33', 'X34', 'X35', 'X36', 'X37', 'X3
8', 'X39', 'X40', 'X41', 'X42', 'X43', 'X44', 'X45', 'X46', 'X47', 'X48', 'X49', 'X50', 'X51', 'X52', 'X53', 'X54', 'X
55', 'X56', 'X57', 'X58', 'X59', 'X60', 'X61', 'X62', 'X63', 'X64', 'X65', 'X66', 'X67', 'X68', 'X69', 'X70', 'X71',
'X73', 'X74', 'X75', 'X76', 'X77', 'X78', 'X79', 'X80', 'X81', 'X82', 'X83', 'X84', 'X85', 'X86', 'X87', 'X88', 'X89',
'X90', 'X91', 'X92', 'X93', 'X94', 'X95', 'X96', 'X97', 'X98', 'X99', 'X100', 'X101', 'X102', 'X103', 'X104', 'X105',
'X106', 'X107', 'X108', 'X109', 'X110', 'X111', 'X112', 'X113', 'X114', 'X115', 'X116', 'X117', 'X118', 'X119', 'X120'
, 'X122', 'X123', 'X124', 'X125', 'X126', 'X127', 'X128', 'X129', 'X130', 'X131', 'X132', 'X133', 'X134', 'X135', 'X13
6', 'X137', 'X138', 'X139', 'X140', 'X141', 'X142', 'X143', 'X144', 'X145', 'X146', 'X147', 'X148', 'X150', 'X151', 'X
152', 'X153', 'X154', 'X155', 'X156', 'X157', 'X158', 'X159', 'X160', 'X161', 'X162', 'X163', 'X164', 'X165', 'X166',
'X167', 'X168', 'X169', 'X170', 'X171', 'X172', 'X173', 'X174', 'X175', 'X176', 'X177', 'X178', 'X179', 'X180', 'X181'
, 'X182', 'X183', 'X184', 'X185', 'X186', 'X187', 'X189', 'X190', 'X191', 'X192', 'X194', 'X195', 'X196', 'X197', 'X19
8', 'X199', 'X200', 'X201', 'X202', 'X203', 'X204', 'X205', 'X206', 'X207', 'X208', 'X209', 'X210', 'X211', 'X212', 'X
213', 'X214', 'X215', 'X216', 'X217', 'X218', 'X219', 'X220', 'X221', 'X222', 'X223', 'X224', 'X225', 'X226', 'X227',
'X228', 'X229', 'X230', 'X231', 'X232', 'X233', 'X234', 'X235', 'X236', 'X237', 'X238', 'X239', 'X240', 'X241', 'X242'
, 'X243', 'X244', 'X245', 'X246', 'X247', 'X248', 'X249', 'X250', 'X251', 'X252', 'X253', 'X254', 'X255', 'X256', 'X25
7', 'X258', 'X259', 'X260', 'X261', 'X262', 'X263', 'X264', 'X265', 'X266', 'X267', 'X268', 'X269', 'X270', 'X271', 'X
272', 'X273', 'X274', 'X275', 'X276', 'X277', 'X278', 'X279', 'X280', 'X281', 'X282', 'X283', 'X284', 'X285', 'X286',
'X287', 'X288', 'X289', 'X290', 'X291', 'X292', 'X293', 'X294', 'X295', 'X296', 'X297', 'X298', 'X299', 'X300', 'X301'
, 'X302', 'X304', 'X305', 'X306', 'X307', 'X308', 'X309', 'X310', 'X311', 'X312', 'X313', 'X314', 'X315', 'X316', 'X31
7', 'X318', 'X319', 'X320', 'X321', 'X322', 'X323', 'X324', 'X325', 'X326', 'X327', 'X328', 'X329', 'X330', 'X331', 'X
332', 'X333', 'X334', 'X335', 'X336', 'X337', 'X338', 'X339', 'X340', 'X341', 'X342', 'X343', 'X344', 'X345', 'X346',
'X347', 'X348', 'X349', 'X350', 'X351', 'X352', 'X353', 'X354', 'X355', 'X356', 'X357', 'X358', 'X359', 'X360', 'X361'
, 'X362', 'X363', 'X364', 'X365', 'X366', 'X367', 'X368', 'X369', 'X370', 'X371', 'X372', 'X373', 'X374', 'X375', 'X37
6', 'X377', 'X378', 'X379', 'X380', 'X382', 'X383', 'X384', 'X385']

    # create categorical feature dataframe
    cat_df = data[cat_features]
    # create binary feature dataframe
    bin_df = pd.DataFrame(data[Binary_features], dtype='int64', columns = Binary_features)
    bin_df.insert(0, 'ID', IDs)
    bin_df = pd.DataFrame(bin_df, dtype='int64', columns = bin_df.columns)

    # Now encode each categorical feature
    for feature in cat_features:
        encoder = joblib.load(f'{feature}encoder.sav')
        cat_df[feature] = encoder.transform(cat_df[feature])

    # Create new categorical feature dataframe
    cat_df = pd.DataFrame(cat_df, columns = cat_features)
    cat_df.insert(0, 'ID', IDs)
    cat_df = pd.DataFrame(cat_df, dtype='int64',columns = cat_df.columns)
    # Merge binary and categorical dataframes together
    new_data = pd.merge(cat_df, bin_df, on='ID', how='left')
    # return dataframe
    return new_data
```

2. Final Predict Function

final_fun_1(X)

In [5]:

```
def final_fun_1(X):
    X_processed = preprocess_categorical(X, X.ID)
    model1 = joblib.load('final_best_model1.pkl')
    y_pred1 = model1.predict(X_processed)
    dtest = xgb.DMatrix(X_processed)
    model2 = joblib.load('final_best_model2.pkl')
    y_pred2 = model2.predict(dtest)
    # Average the test data preditions of both models
    pred_test = (y_pred1 + y_pred2)/2
    return pred_test
```

final_fun_2(X,y)

In [6]:

```
def final_fun_2(X,y):
    X_processed = preprocess_categorical(X, X.ID)
    model1 = joblib.load('final_best_model1.pkl')
    y_pred1 = model1.predict(X_processed)
    dtest = xgb.DMatrix(X_processed)
    model2 = joblib.load('final_best_model2.pkl')
    y_pred2 = model2.predict(dtest)
    # Average the test data preditions of both models
    pred_test = (y_pred1 + y_pred2)/2

    return r2_score(y,pred_test)
```

In [7]:

```
X
```

Out[7]:

	ID	X0	X1	X2	X3	X4	X5	X6	X8	X10	...	X375	X376	X377	X378	X379	X380	X382	X383	X384	X385
0	0	k	v	at	a	d	u	j	o	0	...	0	0	1	0	0	0	0	0	0	0
1	6	k	t	av	e	d	y	l	o	0	...	1	0	0	0	0	0	0	0	0	0
2	7	az	w	n	c	d	x	j	x	0	...	0	0	0	0	0	0	1	0	0	0
3	9	az	t	n	f	d	x	l	e	0	...	0	0	0	0	0	0	0	0	0	0
4	13	az	v	n	f	d	h	d	n	0	...	0	0	0	0	0	0	0	0	0	0
5	18	t	b	e	c	d	g	h	s	0	...	0	0	1	0	0	0	0	0	0	0
6	24	al	r	e	f	d	f	h	s	0	...	0	0	0	0	0	0	0	0	0	0
7	25	o	l	as	f	d	f	j	a	0	...	0	0	0	0	0	0	0	0	0	0
8	27	w	s	as	e	d	f	i	h	0	...	1	0	0	0	0	0	0	0	0	0
9	30	j	b	aq	c	d	f	a	e	0	...	0	0	1	0	0	0	0	0	0	0
10	31	h	r	r	f	d	f	h	p	0	...	0	0	0	0	0	0	0	0	0	0

11 rows × 377 columns

In [8]:

```
y
```

Out[8]:

```
0     130.81
1      88.53
2      76.26
3      80.62
4      78.02
5      92.93
6     128.76
7      91.91
8     108.67
9     126.99
10    102.09
Name: y, dtype: float64
```

In [9]:

```
print(f"predicted outputs are: \n {final_fun_1(X)}")
```

```
predicted outputs are: 
 [109.4382    96.81688   78.91841   79.19795   79.68168   96.462906
 102.292175  95.04414  112.91803  113.5829   105.116936]
```

In [10]:

```
print(f"R2 Score for predicted output is {final_fun_2(X,y)}")
```

```
R2 Score for predicted output is 0.6548714255887931
```