LAB EVAL 102117154

PREDICTIVE ANALYSIS

```
import os

for dirname, _, filenames in os.walk('/kaggle/input'):

for filename in filenames:

print(os.path.join(dirname, filename))
```

```
/kaggle/input/2024ucs654labeval1004/Lab Eval/test_data.parquet
/kaggle/input/2024ucs654labeval1004/Lab Eval/train_data.parquet
```

```
import numpy as np

import pandas as pd

tantrain = pd.read_parquet('/kaggle/input/2024ucs654labeval1004/Lab Eval/train_data.parquet')

tantest = pd.read_parquet('/kaggle/input/2024ucs654labeval1004/Lab Eval/test_data.parquet')

tantrain.head()
```

id	era	data_type	feature_honoured_observational_balaamite	feature_polaroid_vadose_quinze	feature_untidy_withdraw
06e83fe1c412	0439	train	2	3	
:bf822b8badb	0329	train	4	3	
c4b2f97ee047	0287	train	2	4	
03b605aff41f	0357	train	2	4	
:47de23c5039	0061	train	1	1	

```
tantest.head()
```

: id	era	data_type	feature_honoured_observational_balaamite	feature_polaroid_vadose_quinze	feature_untidy_v
n44a9f78060492f8	0061	train	0	2	
n5aea54a0db90cd1	0420	train	4	3	
nd369b32a080e8be	0246	train	3	0	
n154c044f9ae1b5a	0075	train	4	0	
n3ac31439e3e09e6	0132	train	4	2	

```
tantrain.isnull().sum()
```

```
era                                     0
data_type                               0
feature_honoured_observational_balaamite    0
feature_polaroid_vadose_quinze          0
feature_untidy_withdrawn_bargeman       0
                                       ..
T5                                      0
T6                                      0
T7                                      0
T8                                      0
T9                                      0
Length: 2388, dtype: int64
```

```
tantrain.shape
```

```
(1936416, 2388)
```

random_seed = 42

sample_size=1930000

np.random.seed(random_seed)

tantrain= tantrain.sample(n=sample_size)

y_tantrain = tantrain['T4']

tantrain=tantrain.drop(columns=['era', 'data_type', 'T0', 'T1', 'T2', 'T3','T5', 'T6', 'T7', 'T8', 'T9'])

X_tantrain = tantrain

del tantrain

corr_values = X_tantrain.apply(lambda x: x.corr(y_tantrain))

print(corr_values)

```
feature_honoured_observational_balaamite        -0.000496
feature_polaroid_vadose_quinze                  -0.000276
feature_untidy_withdrawn_bargeman               -0.000746
feature_genuine_kyphotic_trehala                -0.000478
feature_unenthralled_sportful_schoolhouse       -0.001857
                                                   ...
feature_crankier_stupefied_bailsman              0.000693
feature_vizirial_bespangled_pteridophyte         0.000432
feature_interventionist_gambling_osteomalacia    0.000164
feature_unnameable_unphonetic_conniver           0.000092
T4                                               1.000000
Length: 2377, dtype: float64
```

corr_values = corr_values.sort_values()

print(corr_values)

```
feature_third_discreet_solute                  -0.016201
feature_obbligato_crackbrained_wolverhampton   -0.014353
feature_terroristic_tripersonal_pashm          -0.013887
feature_unconjugal_chiropodial_amorosity       -0.013800
feature_undisguised_unenviable_stamen          -0.012849
                                                  ...
feature_preachy_unsatisfying_chaeta             0.013241
feature_bumpier_maidenlike_chordata             0.013629
feature_ulterior_flabbier_antimasque            0.013975
feature_suspensory_unrecounted_transcendent     0.016124
T4                                              1.000000
Length: 2377, dtype: float64
```

cortan = corr_values[corr_values<=-0.01]

cor2tan = corr_values[corr_values>=0.01]

print(cor2tan.index)

```
Index(['feature_thymic_formidable_misericord',
       'feature_splashier_conservant_ultramarine',
       'feature_iridescent_abiogenetic_sena',
       'feature_detectable_fogbound_dicastery',
       'feature_instructional_confutative_shaktism',
       'feature_premillennial_furuncular_founding',
       'feature_applausive_forgettable_mishanter',
       'feature_electronegative_lactogenic_merc',
       'feature_community_premandibular_fervor',
       'feature_satisfied_aymaran_enterotomy',
       'feature_left_retroflexed_underclassman',
       'feature_liberticidal_subaqua_embassador',
       'feature_chunky_fallen_erasure', 'feature_sodding_choosy_eruption',
       'feature_fourieristic_allied_mugwumpery',
       'feature_preachy_unsatisfying_chaeta',
       'feature_bumpier_maidenlike_chordata',
       'feature_ulterior_flabbier_antimasque',
       'feature_suspensory_unrecounted_transcendent', 'T4'],
      dtype='object')
```

finalcolumns=['feature_third_discreet_solute',

'feature_obbligato_crackbrained_wolverhampton',

'feature_undisguised_unenviable_stamen',

'feature_terroristic_tripersonal_pashm',

'feature_unconjugal_chiropodial_amorosity',

'feature_undrilled_wheezier_countermand',

'feature_encysted_conventionalized_dematerialization',

'feature_unbarking_apolitical_hibernian',

'feature_surrogate_unmalleable_tasset', 'feature_wetter_unbaffled_loma',

'feature_unscriptural_coconut_trisulphide',

'feature_optical_kempt_aisle', 'feature_fanfold_tartarian_diamondback',

'feature_elmier_unidentifiable_broccoli',

'feature_eruciform_novice_thanker', 'feature_zincky_unseemly_butt',

'feature_multipolar_syncopated_ambrotype',

'feature_addressable_intransitive_reconnoitrer',

'feature_lemuroid_unwishful_mannequin',

'feature_unreproving_capsian_decolourization',

'feature_bursarial_southmost_kaduna',

'feature_goyish_riparian_recipient', 'feature_unpreached_pickiest_lint',

'feature_amitotic_gonadial_submediant',

'feature_domanial_shellproof_rationing',

'feature_subfusc_furriest_nervule',

'feature_herniated_exasperate_victorian',

'feature_setose_processed_crevice',

'feature_gandhian_discretional_cricoid',

'feature_associate_unproper_gridder',

'feature_laziest_saronic_hornbeam', 'feature_milkier_gassy_pincushion',

'feature_shrinelike_introverted_eagre',

'feature_smuggest_galvanic_memorial',

'feature_toed_accusatory_zoologist', 'feature_kirtled_cockiest_etaerio',

'feature_fearsome_merry_bluewing', 'feature_scissile_dejected_kainite',

'feature_incertain_catchable_zibet', 'feature_synodal_feisty_weave',

'feature_anencephalic_unattempted_pschent',

'feature_shrinelike_unreplaceable_nitrogenization',

'feature_bamboo_nosier_phil', 'feature_litigant_unsizable_rhebok',

'feature_sensitive_incendiary_heraclid',

'feature_fungible_allotted_deterioration',

'feature_idled_unwieldy_improvement',

'feature_deposed_toughish_bribery', 'feature_cupular_porky_catafalque','feature_preterite_antediluvian_parasailing',

'feature_unpitied_jingoist_pyretology',

'feature_hippiatric_tinctorial_slowpoke',

'feature_swelled_jugate_haystack', 'feature_bifocal_disposable_clacton',

'feature_thymic_formidable_misericord',

'feature_electronegative_lactogenic_merc',

'feature_cloaked_taillike_usurpation',

'feature_nonnegotiable_errant_soya', 'feature_sodding_choosy_eruption',

'feature_fumed_pivotal_oscine', 'feature_unconfinable_snuffly_cupid',

'feature_detectable_fogbound_dicastery',

'feature_phrenetic_visitorial_entrenchment',

'feature_subatomic_raffish_hexagram',

'feature_fishable_ascendible_micky',

'feature_manufactured_nodal_seeking',

'feature_splashier_conservant_ultramarine',

'feature_premillennial_furuncular_founding',

'feature_instructional_confutative_shaktism',

'feature_community_premandibular_fervor',

'feature_left_retroflexed_underclassman',

'feature_satisfied_aymaran_enterotomy',

'feature_liberticidal_subaqua_embassador',

'feature_fourieristic_allied_mugwumpery',

'feature_chunky_fallen_erasure', 'feature_preachy_unsatisfying_chaeta',

'feature_bumpier_maidenlike_chordata',

'feature_ulterior_flabbier_antimasque',

'feature_suspensory_unrecounted_transcendent']

X_tantrain=X_tantrain[finalcolumns]

from sklearn.model_selection import train_test_split

X_tantrain_splitting, X_tantest_splitting, y_tantrain_splitting, y_tantest_splitting = train_test_split(X_tantrain, y_tantrain, test_size=0.2, random_state=42)

```
print(cortan.index.shape)
```

```
(23,)
```

```
import xgboost as xgb

dubtrain = xgb.DMatrix(X_tantrain_splitting, label=y_tantrain_splitting)

dubtest = xgb.DMatrix(X_tantest_splitting, label=y_tantest_splitting)

params = {

'max_depth': 20,

'learning_rate': 0.07,

'objective': 'reg:squarederror',

'eval_metric': 'rmse',

'n_estimators' : 250

}

tan_reg= xgb.XGBRegressor(**params)

tan_reg.fit(X_tantrain_splitting, y_tantrain_splitting)

y_tanprediction = tan_reg.predict(X_tantest_splitting)

from sklearn.metrics import r2_score

r2finalscoretan = r2_score(y_tantest_splitting, y_tanprediction)

print("Final R-square Score:", r2finalscoretan)
```

```
Final R-square Score: 0.3851515073195304
```

```
final_tan_test_data = tantest.drop(columns=['era', 'data_type'])

del tantest

final_tan_test_data=final_tan_test_data[finalcolumns]

mypred = tan_reg.predict(final_tan_test_data)

mysub = pd.DataFrame({'ID': final_tan_test_data.index, 'Target': np.round(mypred, 2)})

mysub.to_csv('submission_final.csv', index=False)

mysub.head()
```

	ID	Target
0	n44a9f78060492f8	0.50
1	n5aea54a0db90cd1	0.54
2	nd369b32a080e8be	0.40
3	n154c044f9ae1b5a	0.54
4	n3ac31439e3e09e6	0.51