NOTEBOOK CREATED BY SURYA SUBRAMANI @ UMich Dearborn

UMID - 69499602

CIS 579 001 Intro to AI - Midterm Project

GoDaddy-Microbusiness Density Forecasting

to forecast microbusiness activity across the United States, as measured by the density of microbusinesses in US counties.

data source : https://www.kaggle.com/competitions/godaddy-microbusiness-density-forecasting/data

What do we have to build a model ?

1)Train Data

2)Test Data

3)Census data for 2017 to 2021

In [38]:
```
import pandas as pd
import numpy as np
```

```python
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
import xgboost as xgb
# Read the data from a CSV file and set the date column as the index
data = pd.read_csv('train-1.csv', parse_dates=['first_day_of_month'])
census = pd.read_csv('census_starter.csv')
import warnings
warnings.filterwarnings("ignore")
```

In [19]:
```python
data.head(3)
```

Out[19]:

	row_id	cfips	county	state	first_day_of_month	microbusiness_density	active
0	1001_2019-08-01	1001	Autauga County	Alabama	2019-08-01	3.007682	1249
1	1001_2019-09-01	1001	Autauga County	Alabama	2019-09-01	2.884870	1198
2	1001_2019-10-01	1001	Autauga County	Alabama	2019-10-01	3.055843	1269

Row_id - Level of the data

Unique combination of unique identifier for each county in different states and first day of the month

cfips

unique identifier for each county in different states

County

county in each state

state

States in United states

date column

first day of the month

Microbussiness density - Target Variable

Microbusinesses per 100 people over the age of 18 in the given county.

Active

Number of Active micro bussiness in that cfips for that month

In [20]:
```
data[["first_day_of_month"]].describe()
```

Out[20]:

	first_day_of_month
count	122265
unique	39
top	2019-08-01 00:00:00
freq	3135
first	2019-08-01 00:00:00
last	2022-10-01 00:00:00

In [21]:
```
meta_data = data[["cfips","county","state"]].drop_duplicates()
meta_data.head()
```

Out[21]:

	cfips	county	state
0	1001	Autauga County	Alabama
39	1003	Baldwin County	Alabama
78	1005	Barbour County	Alabama
117	1007	Bibb County	Alabama
156	1009	Blount County	Alabama

In [22]:

```
rts = data[["row_id","active"]]
rts.head()
```

Out[22]:

	row_id	active
0	1001_2019-08-01	1249
1	1001_2019-09-01	1198
2	1001_2019-10-01	1269
3	1001_2019-11-01	1243
4	1001_2019-12-01	1243

In [23]:

```
census.head()[[ 'cfips','pct_bb_2017', 'pct_bb_2018', 'pct_bb_2019', 'pct_bb_2020',
       'pct_bb_2021', 'pct_college_2017', 'pct_college_2018',
       'pct_college_2019', 'pct_college_2020', 'pct_college_2021',
       'pct_foreign_born_2017', 'pct_foreign_born_2018',
       'pct_foreign_born_2019', 'pct_foreign_born_2020',
       'pct_foreign_born_2021', 'pct_it_workers_2017', 'pct_it_workers_2018',
       'pct_it_workers_2019', 'pct_it_workers_2020', 'pct_it_workers_2021',
       'median_hh_inc_2017', 'median_hh_inc_2018', 'median_hh_inc_2019',
       'median_hh_inc_2020', 'median_hh_inc_2021']]
```

Out[23]:

	cfips	pct_bb_2017	pct_bb_2018	pct_bb_2019	pct_bb_2020	pct_bb_2021	pct_college_2017	pct_college_2018	pct_college_2019	pct_college_2020	...
0	1001	76.6	78.9	80.6	82.7	85.5	14.5	15.9	16.1	16.7	...
1	1003	74.5	78.1	81.8	85.1	87.9	20.4	20.7	21.0	20.2	...
2	1005	57.2	60.4	60.5	64.6	64.6	7.6	7.8	7.6	7.3	...
3	1007	62.0	66.1	69.2	76.1	74.6	8.1	7.6	6.5	7.4	...
4	1009	65.8	68.5	73.0	79.6	81.0	8.7	8.1	8.6	8.9	...

5 rows × 26 columns

pct*bb*[year] - The percentage of households in the county with access to broadband of any type. Derived from ACS table B28002: PRESENCE AND TYPES OF INTERNET SUBSCRIPTIONS IN HOUSEHOLD.

cfips - The CFIPS code.

pct*college*[year] - The percent of the population in the county over age 25 with a 4-year college degree. Derived from ACS table S1501: EDUCATIONAL ATTAINMENT.

pct_foreign*born*[year] - The percent of the population in the county born outside of the United States. Derived from ACS table DP02: SELECTED SOCIAL CHARACTERISTICS IN THE UNITED STATES.

pct_it*workers*[year] - The percent of the workforce in the county employed in information related industries. Derived from ACS table S2405: INDUSTRY BY OCCUPATION FOR THE CIVILIAN EMPLOYED POPULATION 16 YEARS AND OVER.

median_hh*inc*[year] - The median household income in the county. Derived from ACS table S1901: INCOME IN THE PAST 12 MONTHS (IN 2021 INFLATION-ADJUSTED DOLLARS).

aggregating the time series based on state

In [24]:
```
df_agg_state = data.groupby(["state"])["microbusiness_density"].sum().reset_index().sort_values(["microbusiness_density"])
df_agg_state
```

Out[24]:

	state	microbusiness_density
8	District of Columbia	526.850567
39	Rhode Island	1362.145747
11	Hawaii	1778.801403
7	Delaware	2127.598977
6	Connecticut	2217.518366
29	New Hampshire	2509.145302
2	Arizona	2885.646367
19	Maine	3180.834020
48	West Virginia	3979.954619
45	Vermont	4042.490106
31	New Mexico	4428.778698
1	Alaska	4661.258310
21	Massachusetts	4855.031573
34	North Dakota	4999.870177
24	Mississippi	5424.754950
18	Louisiana	5605.635756
40	South Carolina	5635.771277
0	Alabama	5809.415407
3	Arkansas	5846.856972
30	New Jersey	6335.505169
20	Maryland	6616.973599
36	Oklahoma	7829.962569
47	Washington	8091.225348
50	Wyoming	8105.924815

	state	microbusiness_density
28	Nevada	8327.364798
26	Montana	8503.581363
41	South Dakota	8841.746603
27	Nebraska	9224.313831
37	Oregon	9288.659600
12	Idaho	9383.975962
38	Pennsylvania	9401.306744
44	Utah	9420.903996
17	Kentucky	9721.107985
49	Wisconsin	10007.776133
16	Kansas	10086.356252
35	Ohio	10415.929944
14	Indiana	10989.579688
22	Michigan	11648.813596
23	Minnesota	11764.073330
15	Iowa	12049.931524
25	Missouri	12200.609843
42	Tennessee	12356.889239
32	New York	12619.303182
13	Illinois	13008.081021
33	North Carolina	15779.419395
4	California	17090.936967
9	Florida	18142.933470
10	Georgia	20410.434905

	state	microbusiness_density
5	Colorado	21770.780950
46	Virginia	22650.570301
43	Texas	32804.161419

District of Columbia Analysis

In [25]:
```
data[data.state == "District of Columbia"].groupby(["county"])["microbusiness_density"].sum().reset_index().sort_values(["microbu
```

Out[25]:

	county	microbusiness_density
0	District of Columbia	526.850567

Texas state analysis

In [26]:
```
data[data.state == "Texas"].groupby(["county"])["microbusiness_density"].sum().reset_index().sort_values(["microbusiness_density'
```

Out[26]:

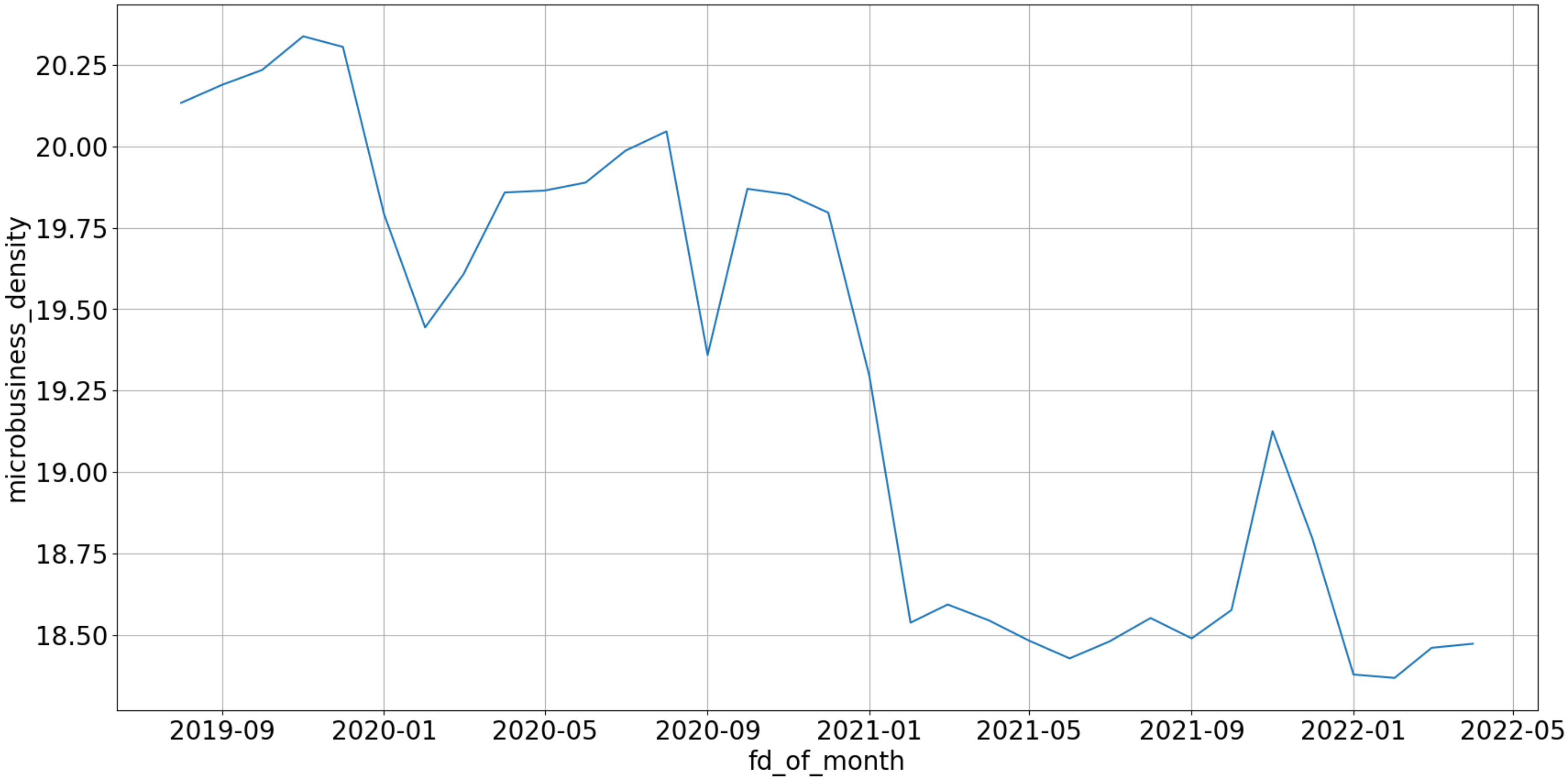
	county	microbusiness_density
129	Kendall County	519.879657
245	Williamson County	542.115238
42	Collin County	614.161772
85	Gillespie County	671.869068
226	Travis County	746.984370

In [27]:
```
df_texas_travis = data[data.county == "Travis County"][["first_day_of_month","microbusiness_density"]]
```

In [28]:
```
df_texas_travis_h = df_texas_travis.set_index("first_day_of_month").head(33)
df_texas_travis_t = df_texas_travis.set_index("first_day_of_month").tail(6)
```

In [39]:

```
plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size': 20})
#plt.style.use('fivethirtyeight')

plt.plot(df_texas_travis_h)
plt.xlabel('fd_of_month')
plt.ylabel('microbusiness_density')

plt.grid()
plt.show()
```



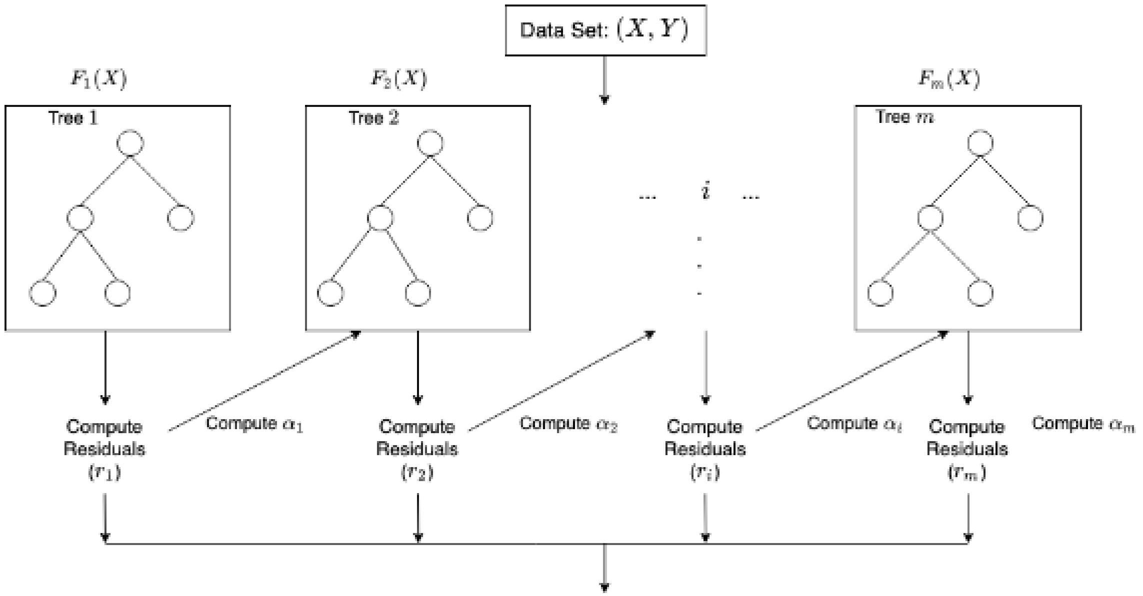
In [30]:

```
# Fit an ARIMA model to the data
model = ARIMA(df_texas_travis_h, order=(1,1,1))
model_fit = model.fit()
```

```python
# Print the model summary
print(model_fit.summary())

# Use the ARIMA model to forecast the next 12 values in the time series
forecast = model_fit.forecast(steps=6)

# Print the forecasted values
print(forecast)
```

```
                                  SARIMAX Results
=================================================================================
Dep. Variable:     microbusiness_density   No. Observations:                   33
Model:                    ARIMA(1, 1, 1)   Log Likelihood                  -5.283
Date:                   Tue, 14 Mar 2023   AIC                             16.566
Time:                           16:02:50   BIC                             20.963
Sample:                       08-01-2019   HQIC                            18.024
                            - 04-01-2022
Covariance Type:                     opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.8806      0.207     -4.245      0.000      -1.287      -0.474
ma.L1          0.9981      3.668      0.272      0.786      -6.190       8.187
sigma2         0.0788      0.280      0.281      0.778      -0.470       0.628
===================================================================================
Ljung-Box (L1) (Q):                   0.03   Jarque-Bera (JB):                 1.43
Prob(Q):                              0.87   Prob(JB):                         0.49
Heteroskedasticity (H):               1.21   Skew:                            -0.47
Prob(H) (two-sided):                  0.76   Kurtosis:                         3.44
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
2022-05-01    18.478442
2022-06-01    18.472848
2022-07-01    18.477774
2022-08-01    18.473436
2022-09-01    18.477256
2022-10-01    18.473892
Freq: MS, Name: predicted_mean, dtype: float64
```

XGBOOST



$$F_m(X) = F_{m-1}(X) + \alpha_m h_m(X, r_{m-1}),$$

where α_i, and r_i are the regularization parameters and residuals computed with the i^{th} tree respectfully, and h_i is a function that is trained to predict residuals, r_i using X for the i^{th} tree. To compute α_i we use the residuals computed, r_i and compute the following: $arg \min_{\alpha} = \sum_{i=1}^{m} L(Y_i, F_{i-1}(X_i) + \alpha h_i(X_i, r_{i-1}))$ where $L(Y, F(X))$ is a differentiable loss function.

In [31]:
```
df_travis_h_xgb = data[data.county == "Travis County"].reset_index().head(33).reset_index()[["index","microbusiness_density"]]
df_travis_t_xgb = data[data.county == "Travis County"].reset_index()[["microbusiness_density"]].reset_index().tail(6)
```

In [32]:
```
df_travis_t_xgb
```

Out[32]:

	index	microbusiness_density
33	33	18.339979
34	34	18.399454
35	35	18.607262
36	36	18.529423
37	37	18.494839
38	38	18.475967

In [33]:

```
reg = xgb.XGBRegressor(n_estimators=1000)
reg.fit(df_travis_h_xgb["index"], df_travis_h_xgb.microbusiness_density,
        eval_set=[(df_travis_h_xgb["index"], df_travis_h_xgb.microbusiness_density), (df_travis_t_xgb["index"], df_travis_t_xgb.m
        early_stopping_rounds=50,
       verbose=False)
```

Out[33]:

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, gpu_id=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,
             n_estimators=1000, n_jobs=None, num_parallel_tree=None,
             predictor=None, random_state=None, ...)
```

In [34]:

```
a = reg.predict(df_travis_t_xgb["index"])
```

In [35]:

```
import pandas as pd
np.array(a).reshape(-1,1)
f = pd.DataFrame(forecast)
```

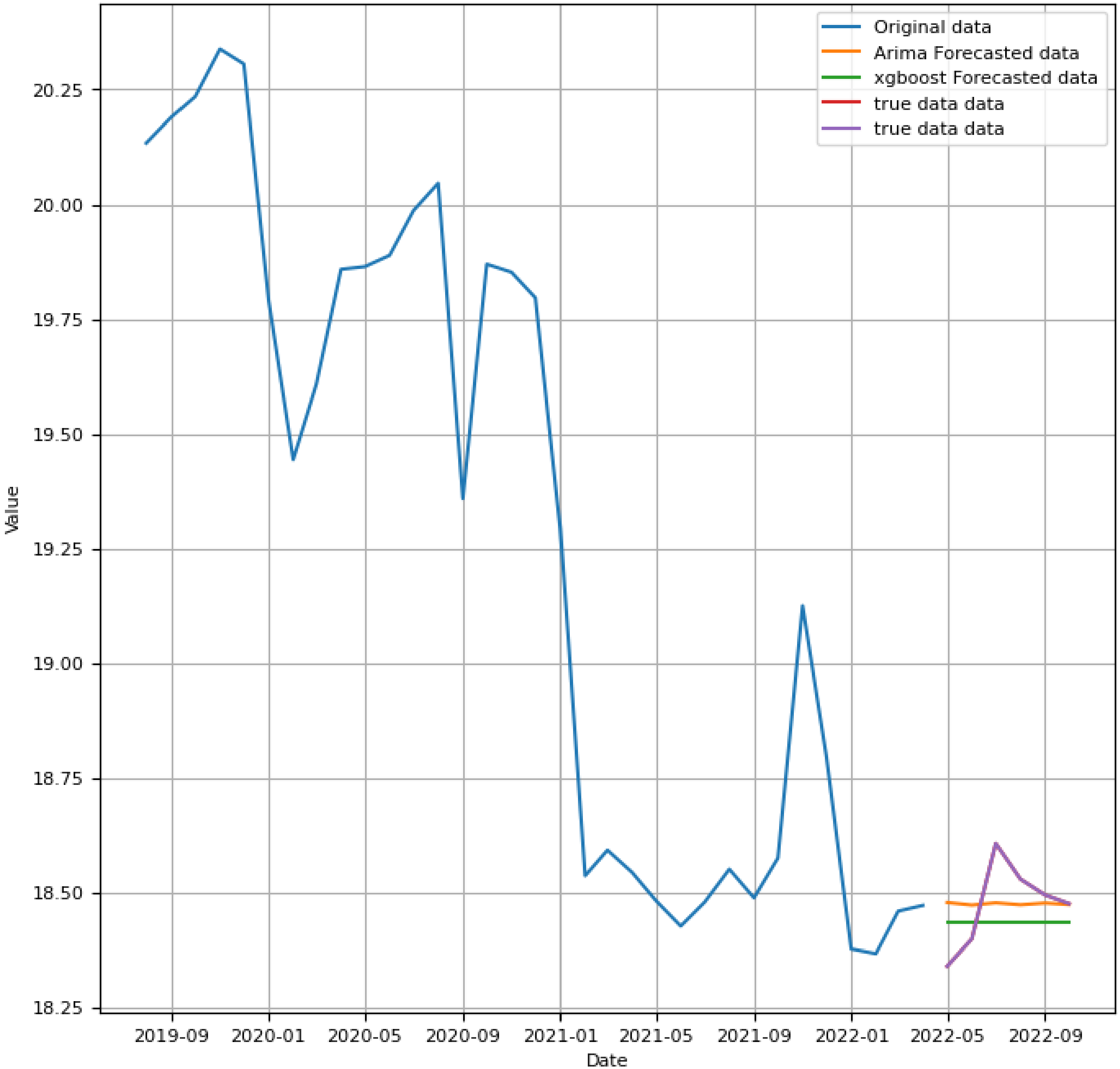
In [36]:

```
f["xgboost"] = a
```

In [43]:

```
# Plot the forecasted values along with the original data
plt.figure(figsize=(8,8))
plt.rcParams.update({'font.size': 8})
```

```
plt.plot(df_texas_travis_h, label='Original data')
plt.plot(f.predicted_mean, label='Arima Forecasted data')
plt.plot(f.xgboost, label='xgboost Forecasted data')
plt.plot(df_texas_travis_t, label='true data data')
plt.plot(df_texas_travis_t, label='true data data')
plt.xlabel('Date')
plt.ylabel('Value')
plt.legend()
plt.grid()
plt.show()
```



Original data
Arima Forecasted data
xgboost Forecasted data
true data data
true data data
20.25
20.00
19.75
19.50
19.25
19.00
18.75
18.50
18.25
Value
2019-09
2020-01
2020-05
2020-09
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
Date

What you can see for the final project

a robust time series forecasting model with suitable regressors using XGB and LSTM (ANN) (tensor flow)

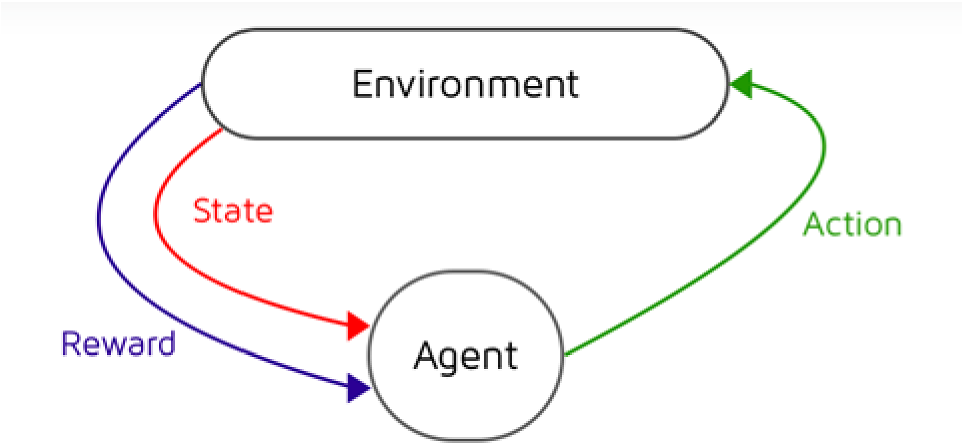
what u can see in the model ?

Analysing correlation, skewness and trend mapping across regressors with target variable

Experimenting Reinforcement learning with sparse time series data

What is Reinforcement learning?

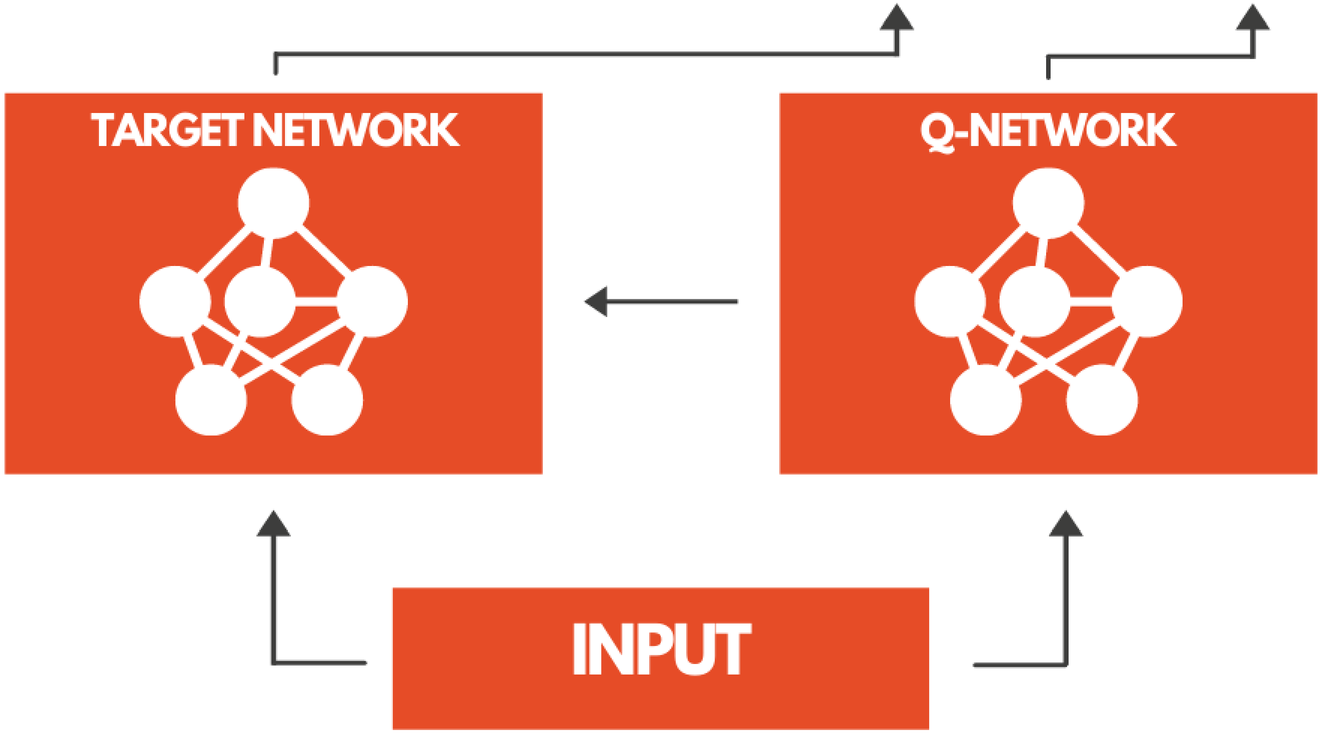
Reinforcement learning is a machine learning training method based on rewarding desired behaviors and/or punishing undesired ones. it works on markovs decisions process



Q-learning and Double-Q-learning

$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{current value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\bigg(\underbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}_{\text{new value (temporal difference target)}} - \underbrace{Q(s_t, a_t)}_{\text{current value}} \bigg)}^{\text{temporal difference}}$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \boxed{\gamma \max_a Q(S_{t+1}, a)} - \boxed{Q(S_t, A_t)} \right]$$



In [ ]: