DAR F21 Project Status Notebook Assignment 3

DeFi

Jason Podgorski (GitHub: podgoj)

09/29/2021

Contents

Introduction

A strong interest of mine in data science is machine learning. Once I analyzed the transaction dataset, I thought the application of time series forecasting models could help uncover patterns in DeFi. I'm using this notebook to dive into learning about how forecasting models work and hopefully become advanced in the subject to help our team discover meaningful trends throughout the semester.

Time Series Forecasting

There are two difference types of time series forecasting models.

1. Traditional Time Series Models

- Recursive (can make predictions easily for any time in the future)
- Tougher to get right
- Can't add regressors for future values
- Ex.
  1. Univariate Models
  - ARIMA
  - SARIMAX
  - Prophet
  - Neural Prophet
  2. Multivariate Models
  - Vector Autoregression

2. Machine Learning Models

- Model trained for set time in the future (two days in advance)
- Easier to get right
- Can add regressors for future values
- Ex.
  - Neural Network Regressor
  - Catboost Regressor
  - Other Regression Models

CodeEmporium. “Time Series Forecasting with Machine Learning.” YouTube, YouTube, 19 Jan. 2021, www.youtube.com/watch?v=_ZQ-lQrK9Rg.

The Question

Can we forecast the number of unique users each day in AAVE?

The feature I chose to forecast was unique users because it is interesting to quantify just how big AAVE will become in terms of popularity. We can also see if there are certain times during the year that spike in activity. My plan is to forecast this feature using both a Time Series Model and a Machine Learning Model and quantify how well each model does with the data.

The Data

```r
# load Rds (binary version of csv file) into dataframe
df <- read_rds('../../Data/transactions.Rds')
head(df)
```

```
##       amount borrowRate borrowRateMode   onBehalfOf         pool reserve
## 1   41501.63   6.274937       Variable 8.502518e+47 1.034668e+48     DAI
## 2 7000000.00   2.589628       Variable 4.635974e+47 1.034668e+48    USDT
## 3   15000.00   8.802541       Variable 3.735263e+47 1.034668e+48    USDC
## 4    8193.19  48.747052         Stable 6.896232e+47 1.034668e+48    USDC
## 5   11000.00   3.225055       Variable 1.089455e+48 1.034668e+48    USDT
## 6   40000.00   5.739208       Variable 2.178337e+47 1.034668e+48    USDT
##    timestamp         user   type reservePriceETH reservePriceUSD  amountUSD
## 1 1621340435 8.502518e+47 borrow    2.852900e+14       0.9948044   41286.00
## 2 1622477822 4.635974e+47 borrow    3.812835e+14       1.0000000 7000000.00
## 3 1619775984 3.735263e+47 borrow    3.611000e+14       1.0043389   15065.08
## 4 1615481632 6.896232e+47 borrow    5.562201e+14       0.9993909    8188.20
## 5 1626914745 1.089455e+48 borrow    4.971100e+14       1.0000000   11000.00
## 6 1620936688 2.178337e+47 borrow    2.725248e+14       1.0000000   40000.00
##   collateralAmount collateralReserve principalAmount principalReserve
## 1               NA                                NA
## 2               NA                                NA
## 3               NA                                NA
## 4               NA                                NA
## 5               NA                                NA
## 6               NA                                NA
##   reservePriceETHPrincipal reservePriceUSDPrincipal reservePriceETHCollateral
## 1                       NA                       NA                        NA
## 2                       NA                       NA                        NA
## 3                       NA                       NA                        NA
## 4                       NA                       NA                        NA
## 5                       NA                       NA                        NA
## 6                       NA                       NA                        NA
##   reservePriceUSDCollateral amountUSDPincipal amountUSDCollateral
## 1                        NA                NA                  NA
## 2                        NA                NA                  NA
## 3                        NA                NA                  NA
## 4                        NA                NA                  NA
## 5                        NA                NA                  NA
## 6                        NA                NA                  NA
##   borrowRateModeFrom borrowRateModeTo stableBorrowRate variableBorrowRate
## 1                                                   NA                 NA
## 2                                                   NA                 NA
## 3                                                   NA                 NA
## 4                                                   NA                 NA
## 5                                                   NA                 NA
## 6                                                   NA                 NA
```

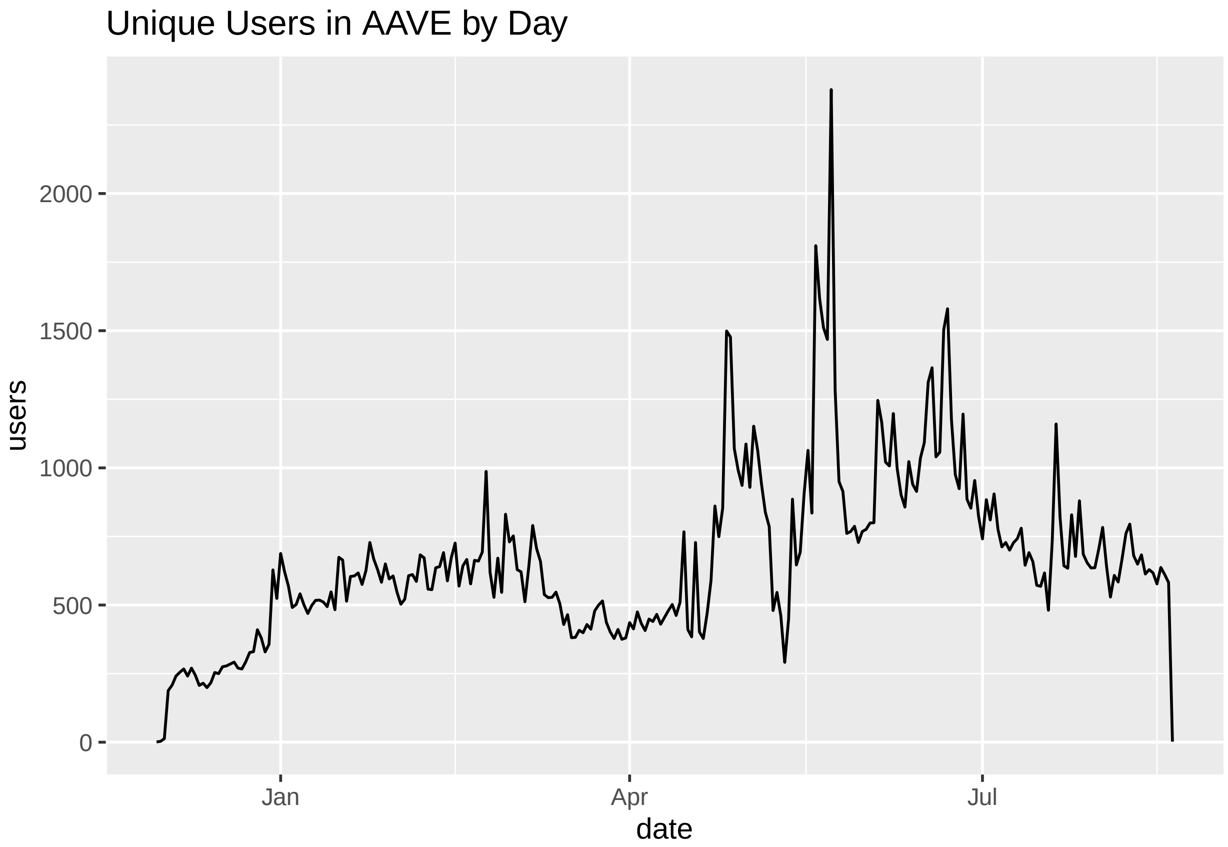
```r
# create a new column in date format using timestamp variable
df <- df[order(df$timestamp),]
posixt <- as.POSIXct(df$timestamp, origin = "1970-01-01")
df$date <- as.Date(posixt)
head(df$date)
```

```
## [1] "2020-11-30" "2020-11-30" "2020-12-01" "2020-12-01" "2020-12-01"
## [6] "2020-12-01"
```

```r
# create new dataframe with the number of unique users each day
# create features with the sum of users from the last 2 days, 7 days, 30 days
users <- df[c("date", "user")]
users <- users %>%
  group_by(date) %>%
  summarize(users = length(unique(user)))
users$users_2 <-  rollsumr(users$users, k = 2, fill = NA)
users$users_7 <- rollsumr(users$users, k = 7, fill = NA)
users$users_30 <- rollsumr(users$users, k = 30, fill = NA)
head(users, 7)
```

```
## # A tibble: 7 x 5
##   date       users users_2 users_7 users_30
##   <date>     <int>   <int>   <int>    <int>
## 1 2020-11-30     1      NA      NA       NA
## 2 2020-12-01     3       4      NA       NA
## 3 2020-12-02    13      16      NA       NA
## 4 2020-12-03   188     201      NA       NA
## 5 2020-12-04   208     396      NA       NA
## 6 2020-12-05   241     449      NA       NA
## 7 2020-12-06   255     496     909       NA
```

```r
ggplot(users, aes(date, users)) +
  geom_line() + ggtitle("Unique Users in AAVE by Day")
```



While we don’t notice any obvious seasonality, this feature is good to forecast because the data spans the full range of dates with many fluctuations throughout the set. In general, time series models are only as good as the data we have.

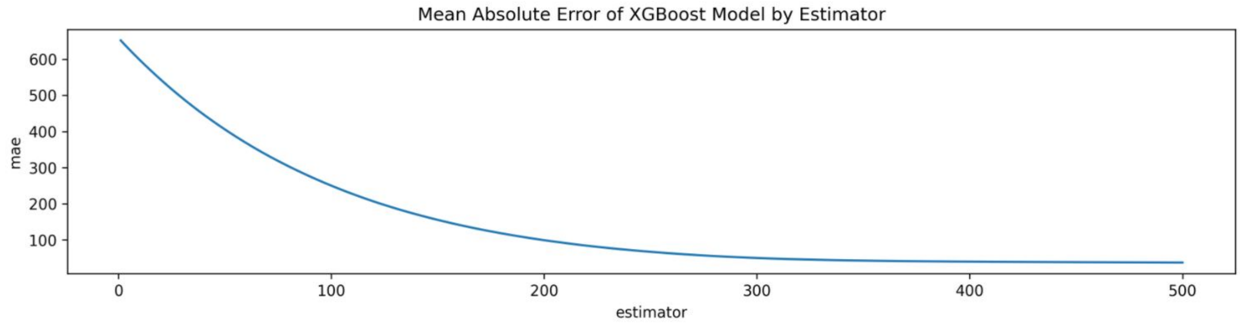
```
# output users dataframe as csv to use in python for time series models
write_csv(users, "users.csv")
```

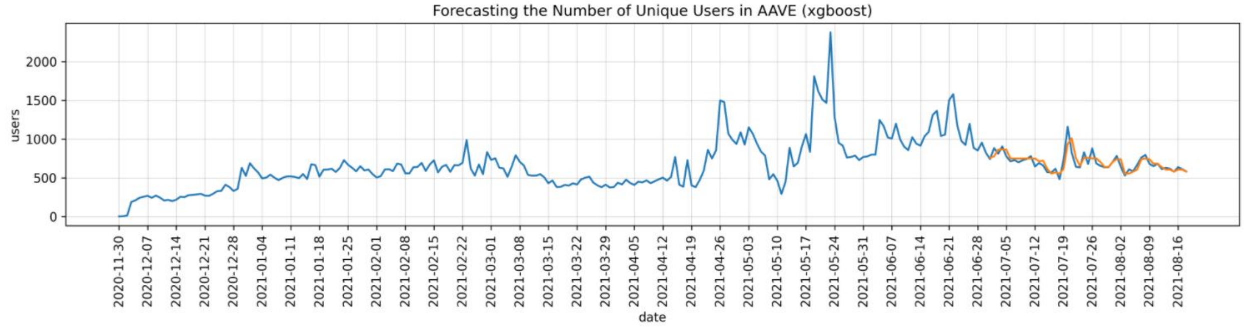
Time Series Forecast Modeling in Python

Link to Notebook: https://colab.research.google.com/drive/1DE1TR7UAlvwzti08CQhegT-uewyBcNnK?usp=sharing

Notebook is also attached in LMS and pushed to GitHub.

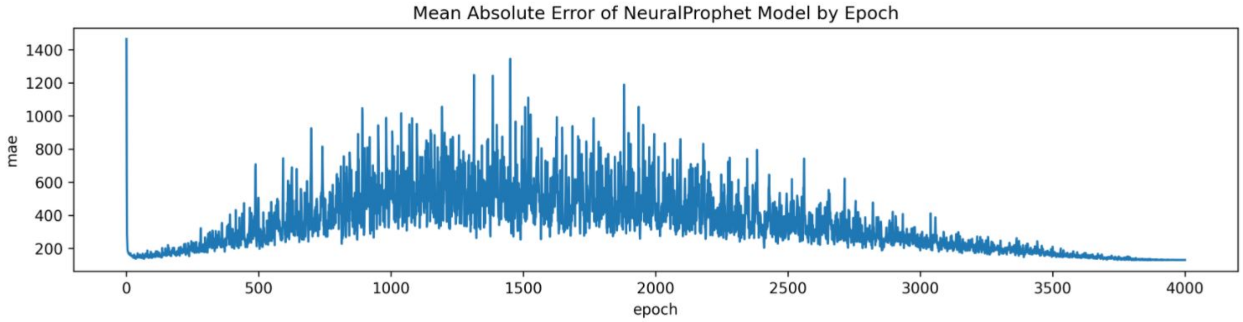
XGBoost

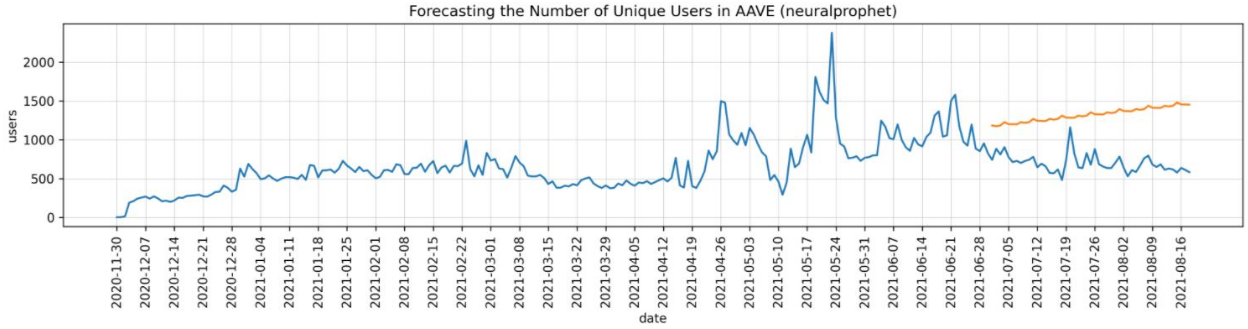




The XGBoost model performed with great success. The model's mean absolute error converged at 50 users. This means on average, the model is 50 users off from the actual value on a given day. We know the model wasn't overfit become it performed strongly when comparing the test data. I believe this model was a good choice for the dataset because we lacked a large volume of dates. However, I was able to supplement the data with 3 additional features to help predict the fluctuations throughout the year.

NeuralProphet





The NeuralProphet model was a poor choice for this dataset. The model's mean absolute error converged at 131 users. This means on average, the model is 131 users off from the actual value on a given day. My hypothesis for this model being unsuccessful is that the dataset was too small. The neural prophet model only takes 1 feature which will make it harder to predict smaller datasets. I've seen the NeuralProphet model be successful with at least a few years worth of dates.

Conclusion

Overall, the xgboost model performed with great success. I would like to try other machine learning models like Random Forest or Neural Network Regression models to see if I can improve our results even more. This was my first experience with time series forecasting. I hope as a group we can narrow in on certain features to explore or more specific questions involving time series. I would really like to dive-in and fully optimize a model to help our group predict essential trends in DeFi.