ProjectReport

Sanjana Annamaneni ,Anjali S Varier

May 3, 2019

Introduction

What is Ethereum?

The Ethereum provides a platform that is decentralised that uses smart contracts which runs on a customized blockchain, which is powerful and a shared global infrastructure and these allow applications to run as programmed applicationswithout any frauding from the third party. [Source : https://www.ethereum.org/]

What is ERC20?

ERC20 is a significant standard for tokens on Etherium. This defines a common list of rules that enables developers to accurately predict how new tokens will function within the larger Ethereum system. [Source : https://www.investopedia.com/news/what-erc20-and-what-does-it-mean-ethereum/]

FUNFAIR TOKEN

FunFair is a decentralized gaming technology platform which uses the Ethereum blockchain, smart contracts, and proprietary state channels (Fate Channels) to deliver casino solutions with state of the art games that are fast, fun, and fair.

KYBER NETWORK TOKEN

KyberNetwork is an on-chain protocol which allows instant exchange and conversion of digital assets (e.g. crypto tokens) and cryptocurrencies (e.g. Ether, Bitcoin, ZCash) with high liquidity.

LOOPRING TOKEN

Loopring is intended to serve as common building block with open standards, driving interoperability among decentralized applications (dApps) that incorporate exchange functionality.

[Source:https://eidoo.io/erc20-tokens-list]and develop

Our goal

The goals for the project are >To find the distribution of how many times a pair users (i.e., address1 and address2) 1 - buys, 2 - sells a token with each other and the best distribution type fits these distributions for funfair token. >To find the most active buyers and sellers in each of your three token network, and track them in other tokens and develop a regression model where "buys" of the top K buyers (by number of buys or amount of buys) are regressors, and token price is the outcome for tokens funfair,kyber network and loopring.

Preprocessing

```r
setwd("C:/Users/Anjali/Documents/R workspace") #setting the working directory
ffunprocessed<-read.table('Ethereum token graphs/networkfunfairTX.txt',sep = " ")
names(ffunprocessed)<-c('sellersID','buyersID','timestamp','tokenamt')
#preprocessing...
```

```r
decimal<- 10^8 # decimal to make it equal
supply<- 10999873621
#remove all the outliers that are greater than the token amt
ffprocessed<-ffunprocessed %>% filter(tokenamt<decimal*supply)
#Finding out the number of outliers
ffunprocessed %>% filter(tokenamt >= decimal * supply)%>% nrow()
```

```
## [1] 91
```

```r
#since the outliers are greater than 30(>30) we have to analyse the outliers and find out the
#number of users involved in these dubious transcations
ffoutliers<-ffunprocessed %>% filter(tokenamt >= decimal * supply)
```

77 of these outliers had the same buyer id, that is many people sold tokens to the same buyer this buyer could be fake buyers cause he was buying tokens more than the supply and also from various users in near time stamp hence removed these outliers by taking unique buyers & sellers.

```r
summary(ffoutliers)
```

```
##    sellersID          buyersID         timestamp
##  Min.   :     81   Min.   :     81   Min.   :1.500e+09
##  1st Qu.:3845928   1st Qu.: 204018   1st Qu.:1.525e+09
##  Median :3845963   Median : 204018   Median :1.526e+09
##  Mean   :3606602   Mean   : 540394   Mean   :1.523e+09
##  3rd Qu.:3846000   3rd Qu.: 204018   3rd Qu.:1.526e+09
##  Max.   :3894967   Max.   :3897664   Max.   :1.526e+09
##     tokenamt
##  Min.   :5.600e+18
##  1st Qu.:5.094e+76
##  Median :5.094e+76
##  Mean   :4.486e+76
##  3rd Qu.:5.094e+76
##  Max.   :5.790e+76
```

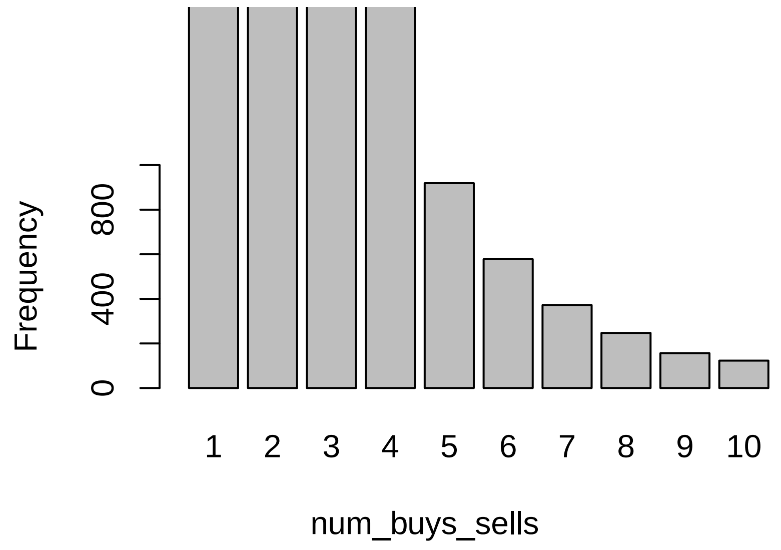
```r
#we can observe that the mean and max values have huge difference thus we took a sample data
```

```r
#we have to know find the distribution followed by the pair of buyers and sellers
```

```r
df<-ff_data %>% group_by(buyersID,sellersID) %>% summarize(n=n())
buy_sell_dis<-df %>% arrange(-n)
buy_sell_dis_10<-df %>% arrange(-n)%>%head(10)
```

```r
#now we want distribution for frequency of buyers and sellers
distribution<-as.data.frame(table(buy_sell_dis$n))
distribution<-distribution%>%arrange(-Freq)
buys_frequency <-c(distribution[,'Freq'])
```

```r
barplot(distribution$Freq,names.arg =distribution $Var1,ylab ="Frequency",xlab = "num_buys_sells",
xlim = c(0,10),ylim = c(0,1000))
```



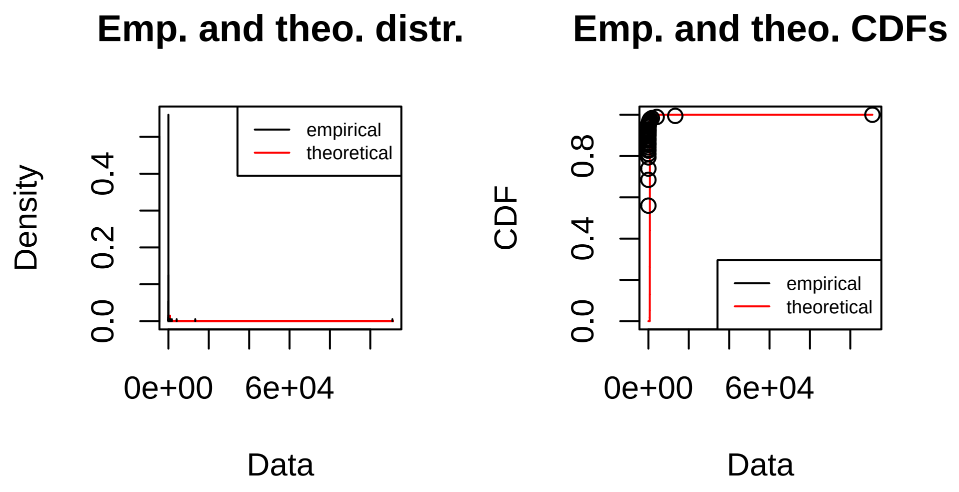
Fitting best distribution

```
fit.normal.dis <- fitdist(buys_frequency, 'norm')
```

- By assuming that our data fits normal distribution, we see that the mean is not 0 and the standard deviation is very much greater than one and the plot also doesnt match our data pattern.we can a draw a cullen frey graph to check the distribution.
- It seems like our data lies in the poisson distribution range "' Conclusion of Question 1

thus lets assume our data follows poissons distribution hence we used fitdist to check our assumtion that it is poisson distribution is true or not.

```
fit.pois.dis <- fitdist(buys_frequency, 'pois')
plot(fit.pois.dis)
```



```
mean(buys_frequency)
```

```
## [1] 724.6576
```

Conclusion of Question 1

```
fit.pois.dis
```

```
## Fitting of the distribution ' pois ' by maximum likelihood
## Parameters:
##        estimate Std. Error
## lambda 724.6576   1.984459
```

we see that lambda=mean and the plot correctly fits the data points thus we can conclude that our distribution is possion distribution.

Question 2

Finding top k buyers & sellers of funfair

After removing the outliers, we will find the buyer and seller by maximum token amount from the token graph table and then form a token graph table as shown below.

```
trans_max <- aggregate(tokenamt ~ date, data1, max)
temp_data<-merge(trans_max,active_buyers_table)
tempdata<-unique(temp_data)
```

Then we will merge the token graph table & token price table by date.

```
tp<-merge(top,active_buyers_table,by="date")
```

- We grouped the merged table by date & then sorted the resulting table by volume (i.e, amount of buys/sells in a day) for finding the top k buyers.We will check which has the highest correlation between volume & open or volume & close.

```
order_by_volume<-tp[order(as.numeric(as.character(tp$volume)), decreasing=TRUE) , ]
unique_order_by_vol<-subset(order_by_volume,!duplicated(order_by_volume$volume))
top_unique_vol<-unique_order_by_vol[1:80,]
```

Finding top k buyers & sellers of kybernetwork & loopring

- We will follow the same procedure as funfair for finding the top k buyers & sellers.
- First we will check for outliers,find the buyer and seller by maximum token amount from the token graph table,then merge the token graph table & token price tabl by date, sort the resulting table by volume (i.e, amount of buys/sells in a day) for finding the top k buyers.
- We will check which has the highest between volume & open or volume & close.
- Take top 200 buyers of kybernetwork & top 10 buyers of loopring which is the converging value as shown below.

Tracking of tokens of most active buyers & sellers

- There are 22 users from the top 50 users of funfair token are present in the kybernetwork token and 15 users in loopring token.
- There are only 10 users from the top 200 users of kyber token are present in the funfair token and 50 users in loopring token.
- There are only 10 users from the top 50 users of loopring token are present in the funfair token and 27 users in kybernetwork token.

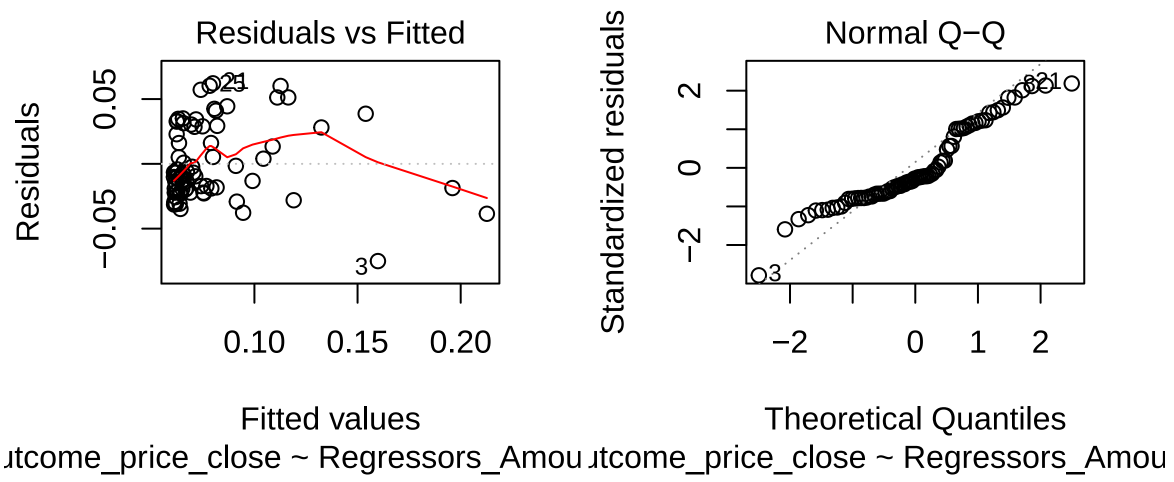
Regression Model for all the 3 tokens

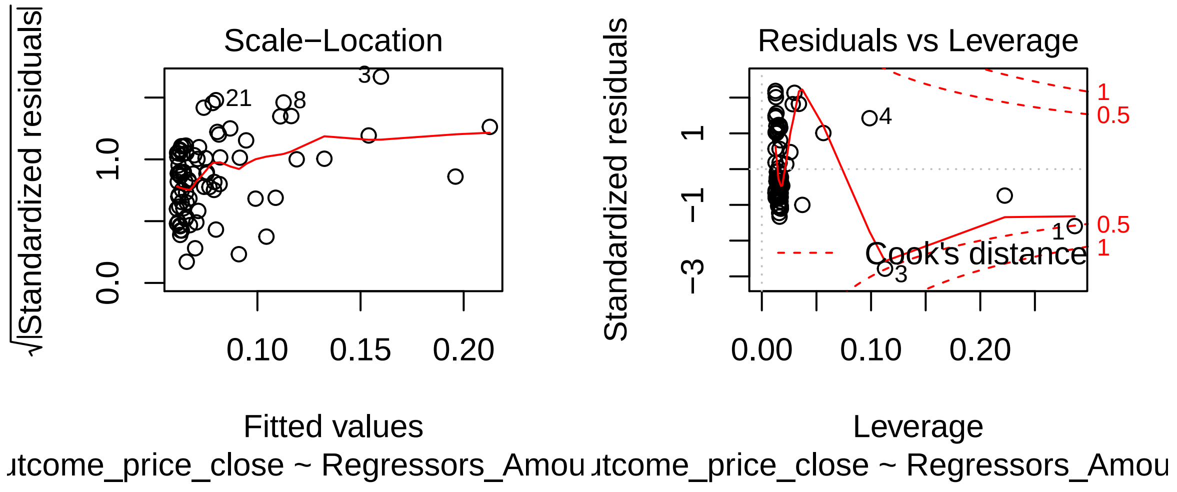
Regression model for funfair

```
summary(model_close)
```

```
##
## Call:
## lm(formula = Outcome_price_close ~ Regressors_Amount_of_buy,
##     data = top_unique_vol)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.075053 -0.019249 -0.007804  0.028201  0.062167
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)               5.481e-02  4.192e-03  13.073  < 2e-16 ***
## Regressors_Amount_of_buy 1.203e-09  1.348e-10   8.928 1.49e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02861 on 78 degrees of freedom
## Multiple R-squared:  0.5054, Adjusted R-squared:  0.4991
## F-statistic: 79.71 on 1 and 78 DF,  p-value: 1.489e-13
```

```
plot(model_close)
```



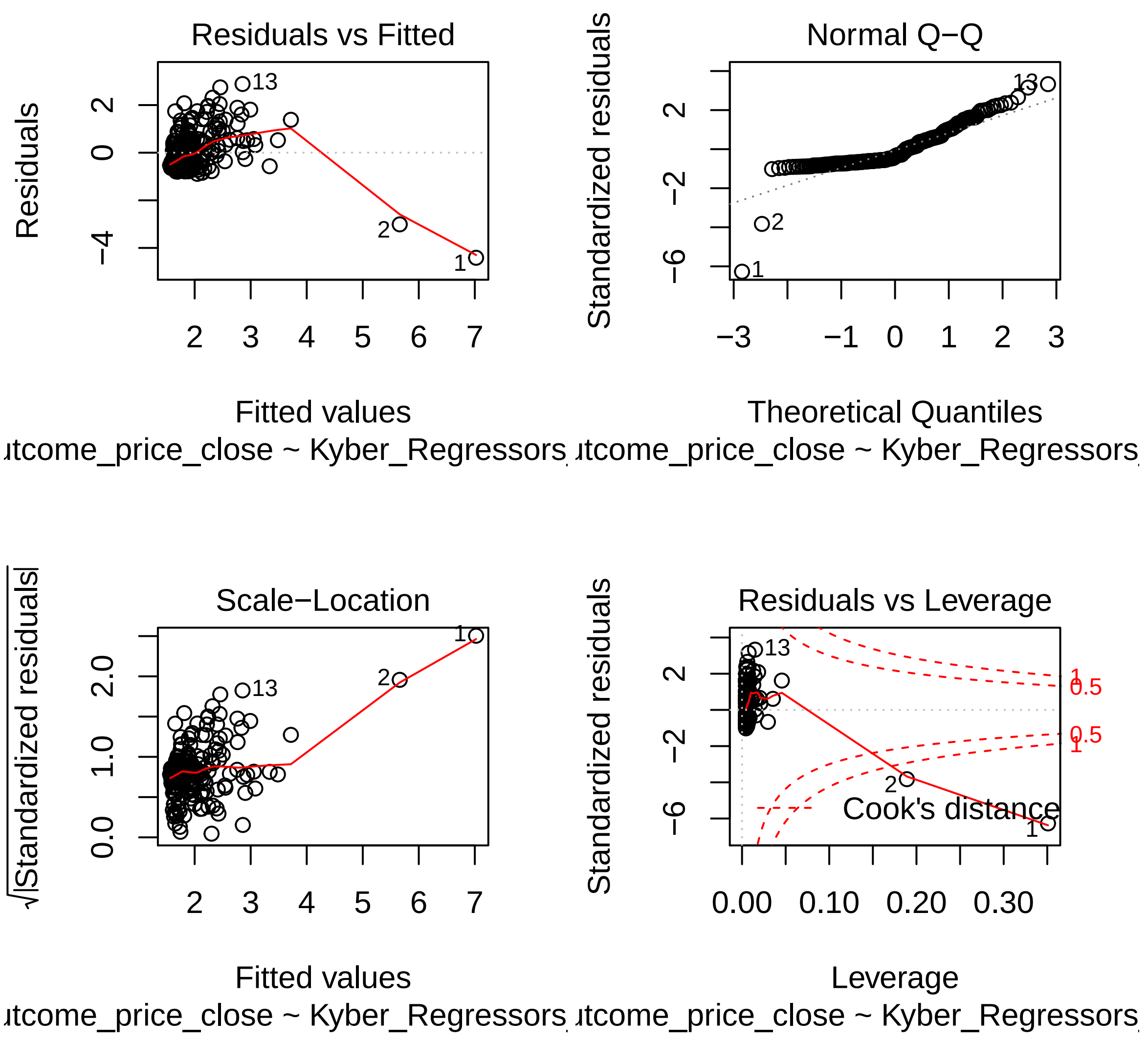


Regression model for kyber network

```r
summary(kyber_model_close)
```

```
##
## Call:
## lm(formula = kyber_Outcome_price_close ~ Kyber_Regressors_Amount_of_buy,
##     data = kyber_top_unique_vol)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4126 -0.5809 -0.3319  0.4701  2.8855
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.526e+00  7.419e-02  20.571   <2e-16 ***
## Kyber_Regressors_Amount_of_buy 3.548e-08  3.614e-09   9.817   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8732 on 223 degrees of freedom
## Multiple R-squared:  0.3018, Adjusted R-squared:  0.2986
## F-statistic: 96.38 on 1 and 223 DF,  p-value: < 2.2e-16
```

```r
plot(kyber_model_close)
```

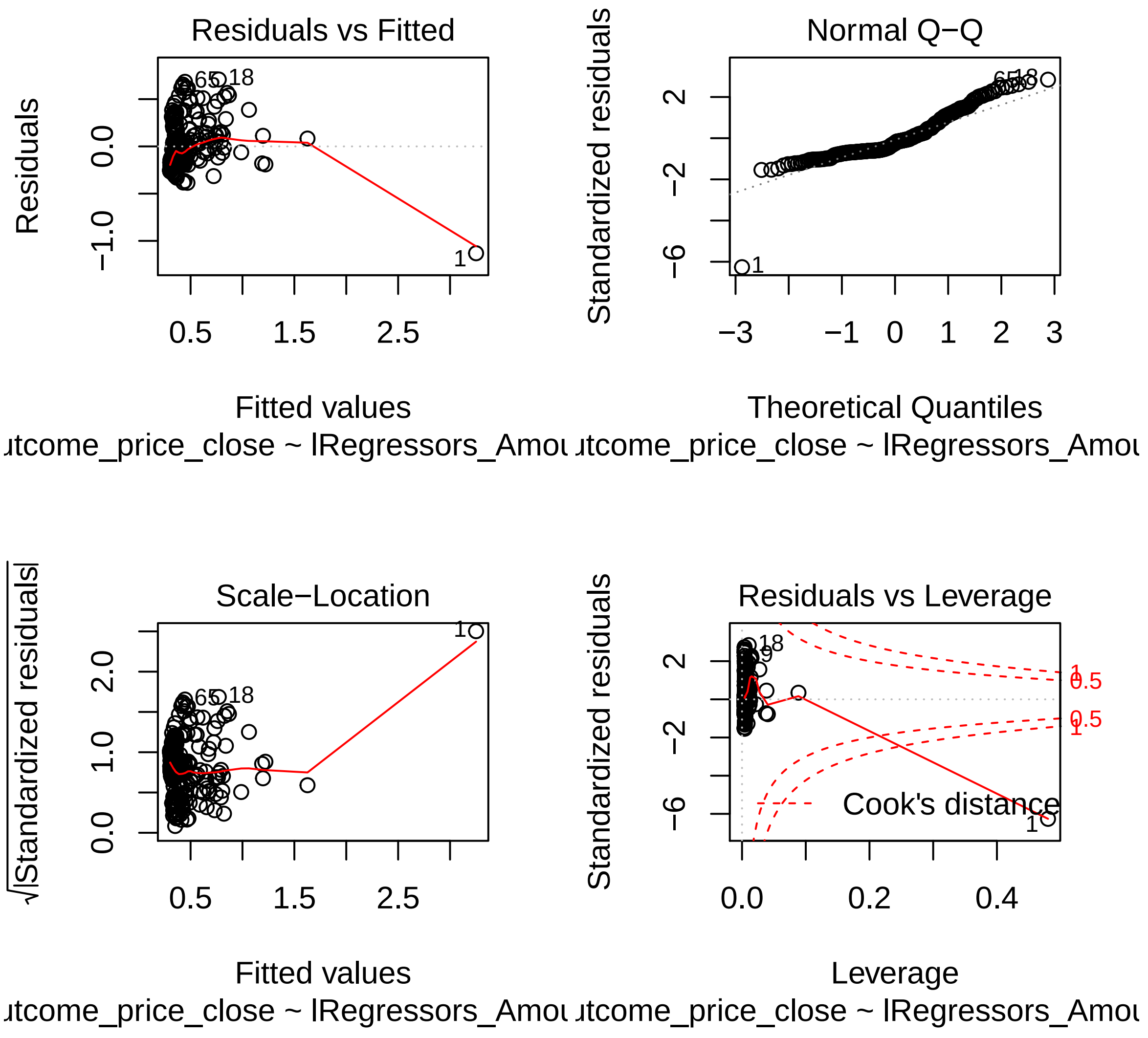


Regression model for loopring

```
summary(lmodel)
```

```
##
## Call:
## lm(formula = lOutcome_price_close ~ lRegressors_Amount_of_buy,
##     data = lunique_order_by_vol)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.13102 -0.16437 -0.06149  0.12247  0.70825
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.025e-01  1.808e-02   16.73   <2e-16 ***
## lRegressors_Amount_of_buy 1.701e-08  1.047e-09   16.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2505 on 248 degrees of freedom
## Multiple R-squared:  0.5154, Adjusted R-squared:  0.5135
## F-statistic: 263.8 on 1 and 248 DF,  p-value: < 2.2e-16
```

```r
plot(lmodel)
```



Conclusion of Question 2

From the above summary & plot,we see that the Adjusted R Square is more and Standard error is less for the taken top k buyers for the tokens. Thus, we have fitted the linear regression model.