

ÜSKÜDAR ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI  
2022-2023  
FİNAL PROJE RAPORU

ÖĞRENCİ NUMARASI: 214327027

ÖĞRENCİ ADI-SOYADI: HAMZA TALİP EKŞİ

PROJE BAŞLIĞI: Microsoft Stok Verileri Tahmini

## 1. GİRİŞ

Bu çalışmada , RNN ile Microsoft şirketinin stok verileri analiz edilip tahmin edildi.

RNN'ler genelde bir sonraki adımı tahmin etmek için kullanılan bir çeşit Derin Öğrenme yapılarıdır. Diğer derin öğrenme yapılarından en büyük farkları ise hatırlamalarıdır. RNN'ler bir sonraki adımı takip edebilmek için girdiler arasında ilişki kurarlar ve eğilirken tüm ilişkilerini hatırlarlar.

LSTM yapısı içerisindeki kapılar (gate) neyin hatırlanacağını, neyin unutulacağını belirler. Yani gelen girdi önemsizse unutulur, önemliyse bir sonraki aşamaya aktarılır.

## 2. MATERYALLER VE YÖNTEMLER

### Microsoft Corporation (MSFT)

NasdaqGS - NasdaqGS Real Time Price. Currency in USD

☆ Follow

**331.21** -1.68 (-0.50%)

At close: May 30 04:00PM EDT

**332.39** +1.18 (+0.36%)

Pre-Market: 8:10AM EDT

[Summary](#) [Chart](#) [Conversations](#) [Statistics](#) [Historical Data](#) [Profile](#) [Financials](#) [Analysis](#) [Options](#) [Holders](#) [Sustainability](#)



Time Period: May 31, 2022 - May 31, 2023

Show: Historical Prices

Frequency: Daily

Apply

Currency in USD

Download

Date	Open	High	Low	Close*	Adj Close**	Volume
May 26, 2023	324.02	333.40	323.88	332.89	332.89	36,600,900
May 25, 2023	323.24	326.90	320.00	325.92	325.92	43,301,700
May 24, 2023	314.73	316.50	312.61	313.85	313.85	23,384,900
May 23, 2023	320.03	322.72	315.25	315.26	315.26	30,797,200
May 22, 2023	318.60	322.59	318.01	321.18	321.18	24,115,700
May 19, 2023	316.74	318.75	316.37	318.34	318.34	27,529,500
May 18, 2023	314.53	319.04	313.72	318.52	318.52	27,276,000
May 17, 2023	312.29	314.43	310.74	314.00	314.00	24,315,000

Yahoo finance üzerinden alınan Microsoft'un 30 Mayıs 2010-30 Mayıs 2021 arasındaki stok verileri eğitim verisi, 31 Mayıs 2021-30 Mayıs 2023 arasındaki stok verileri test verisi olarak kullanıldı

### 3. ARAÇLAR

Analizin yapılması için gerekli kod Google Colaboratory üzerinde python dili ile yazılmıştır.

### 4. KODLAR

```
[ ] import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

-Dizilerle çalışmak için numpy,veri analizi için pandas,grafik elde etmek için plt kullanıldı.

```
from google.colab import drive
drive.mount('/content/gdrive')
```

-Veri setini çekebilmek için drive ,colab'a bağlandı.

```
[ ] %cd /content/gdrive/MyDrive/Colab Notebooks/

/content/gdrive/MyDrive/Colab Notebooks
```

-Veri setinin çekileceği klasör girildi.

```
!ls
```

```
Amazon.csv
'A TALE OF TWO CITIES.txt'
movie.csv
MSFT_test.csv
MSFT_training.csv
NFLX.csv
NlpFinal.ipynb
NlpVize
'Topic models with Gensim.ipynb adlı not defterinin kopyası'
word2vec.ipynb
word2vec.model
yapayzekafinal.ipynb
```

- Klasördeki dosyalar listelendi.

```
[ ] dataset_train = pd.read_csv('MSFT_training.csv')
    training_set = dataset_train.iloc[:, 1:2].values
```

-Eğitim için kullandığımız veri seti okundu.Sadece “Open” sütunu ile ilgilendiğimiz için önceki sütunları dikkate almıyoruz.

```
[ ] from sklearn.preprocessing import MinMaxScaler
    sc = MinMaxScaler(feature_range= (0,1))
    training_set_scaled = sc.fit_transform(training_set)
```

-Eğitim verisi, MinMaxScaler ile 0-1 aralığına normalize edildi.

```
▶ x_train = []
  y_train = []
  for i in range (60, len(training_set_scaled)) :
      x_train.append(training_set_scaled[i-60:i,0])
      y_train.append(training_set_scaled[i,0])
  x_train, y_train = np.array(x_train), np.array(y_train)
```

-Döngü içerisinde ilk satırda her 60 güne bakıp 61.günü tahmin etmesi isteniyor.İkinci satırda 1.günü dışarda bırakıp 2. Günden itibaren sonraki günü tahmin etmeye çalışması isteniyor.Böylece bellekli bir veri setine sahip olunmuş olur.

```
[ ] x_train = np.reshape(x_train, newshape = (x_train.shape[0], x_train.shape[1], 1))
```

-x\_train 3 boyutlu bir yapı haline getirildi.

```
[ ] from keras.models import Sequential
    from keras.layers import Dense
    from keras.layers import LSTM
    from keras.layers import Dropout
```

-Yapay sinir ağı oluşturacağımız için Sequential, katmanlarını oluşturmak için Dense,hücrelere bilgi iletimi için LSTM, aşırı öğrenmeyi (overfitting) engellemek için dropout import ediyoruz.

```
[ ] regressor = Sequential()
```

-Sayılarla çalışıldığı için bir regressor kullanıldı.

```
[ ] regressor.add(LSTM(units = 50 , return_sequences = True , input_shape = (x_train.shape[1] , 1)))  
    regressor.add(Dropout(0.2))
```

```
[ ] regressor.add(LSTM(units = 50 , return_sequences = True))  
    regressor.add(Dropout(0.2))
```

```
[ ] regressor.add(LSTM(units = 50 , return_sequences = True))  
    regressor.add(Dropout(0.2))
```

```
[ ] regressor.add(LSTM(units = 50 , return_sequences = False))  
    regressor.add(Dropout(0.2))
```

```
[ ] regressor.add(Dense(units = 1))
```

-Sinir ağının , 50 nöronlu LSTM katmanları oluşturuldu.Her bir katmanın bir sonraki katmana aktarımı “return\_sequences=True” ile sağlanıyor.Bu işlemin son katmanda bitmesi için bu ifade son katmanda “False” olarak kullanıldı.Overfitting’i önlemek için dropout kullanıldı.En sonunda aktarım Dense katmanındaki tek nörona yani sonuca ulaşıyor.

```
[ ] regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')
```

-RNN’i compile etmek için adam optimizer ve sayılarla çalıştığımızdan kayıp fonksiyonu olarak mse kullanıldı.

```
[ ] regressor.fit(x_train , y_train , epochs = 100 , batch_size=32)
```

```
Epoch 1/100
85/85 [=====] - 24s 163ms/step - loss: 0.0098
Epoch 2/100
85/85 [=====] - 14s 166ms/step - loss: 0.0018
Epoch 3/100
85/85 [=====] - 14s 163ms/step - loss: 0.0017
Epoch 4/100
85/85 [=====] - 14s 164ms/step - loss: 0.0014
Epoch 5/100
85/85 [=====] - 14s 169ms/step - loss: 0.0014
Epoch 6/100
85/85 [=====] - 13s 155ms/step - loss: 0.0015
Epoch 7/100
85/85 [=====] - 13s 149ms/step - loss: 0.0015
Epoch 8/100
85/85 [=====] - 13s 156ms/step - loss: 0.0014
Epoch 9/100
85/85 [=====] - 14s 163ms/step - loss: 0.0016
Epoch 10/100
85/85 [=====] - 14s 163ms/step - loss: 0.0011
Epoch 11/100
85/85 [=====] - 14s 163ms/step - loss: 0.0010
Epoch 12/100
85/85 [=====] - 15s 179ms/step - loss: 0.0012
Epoch 13/100
85/85 [=====] - 14s 161ms/step - loss: 0.0010
Epoch 14/100
85/85 [=====] - 14s 161ms/step - loss: 0.0012
Epoch 15/100
```

-RNN , eğitim veri setine fit edildi ve eğitime başlandı.

```
[ ] dataset_test = pd.read_csv('MSFT_test.csv')
    real_stock_price = dataset_test.iloc[:, 1:2].values
```

-Test verisi okundu. Gerçek stok fiyatları alınmış oldu.Sadece “Open” sütunu ile ilgilendiğimiz için önceki sütunları dikkate almıyoruz.

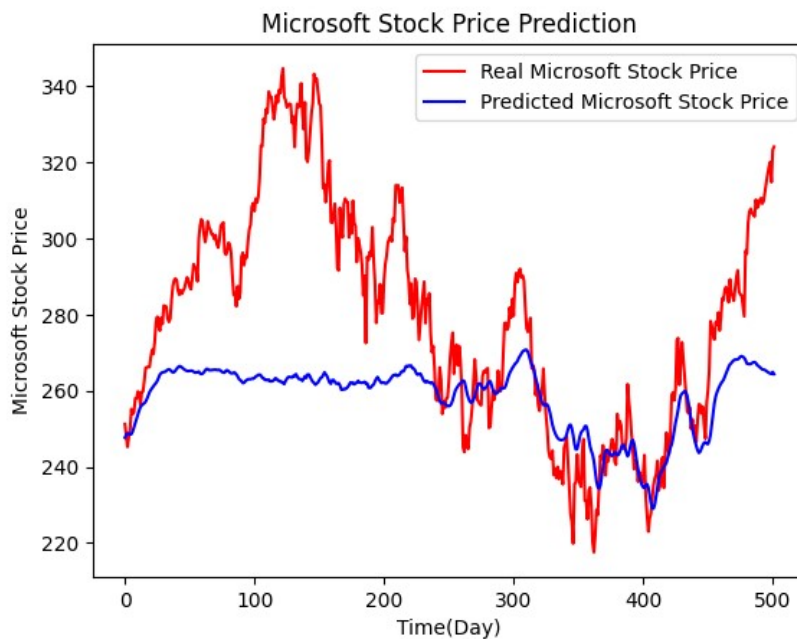
```
[ ] dataset_total = pd.concat((dataset_train['Open'], dataset_test['Open']), axis = 0)
inputs = dataset_total[len(dataset_total) - len(dataset_test) - 60:].values
inputs = inputs.reshape (-1,1)
inputs = sc.transform(inputs)
x_test = []
for i in range(60,len(inputs)):
    x_test.append(inputs[i-60:i , 0])
x_test = np.array(x_test)
x_test = np.reshape(x_test, newshape = (x_test.shape[0],x_test.shape[1], 1))
predicted_stock_price = regressor.predict(x_test)
predicted_stock_price = sc.inverse_transform(predicted_stock_price)
```

16/16 [=====] - 3s 38ms/step

-Birleştirme işlemi yaparak eğitim ve test verisi birleştirildi.Bu birleşimden test verisini ve son 60 günü çıkararak başlangıç noktamızı elde etmiş olduk.Bu değerleri inputs'a tanımlayıp daha sonra reshape ve transform ederek istenilen yapıya getirmiş olduk.Döngü içerisinde inputs'un ilk 60 gününü elde etmiş olduk.Elden edilen sonuç bir diziye ve reshape ile 3 boyutlu hale dönüştürüldü.Predict ile verinin 61.günü tahmin etmesi sağlandı.

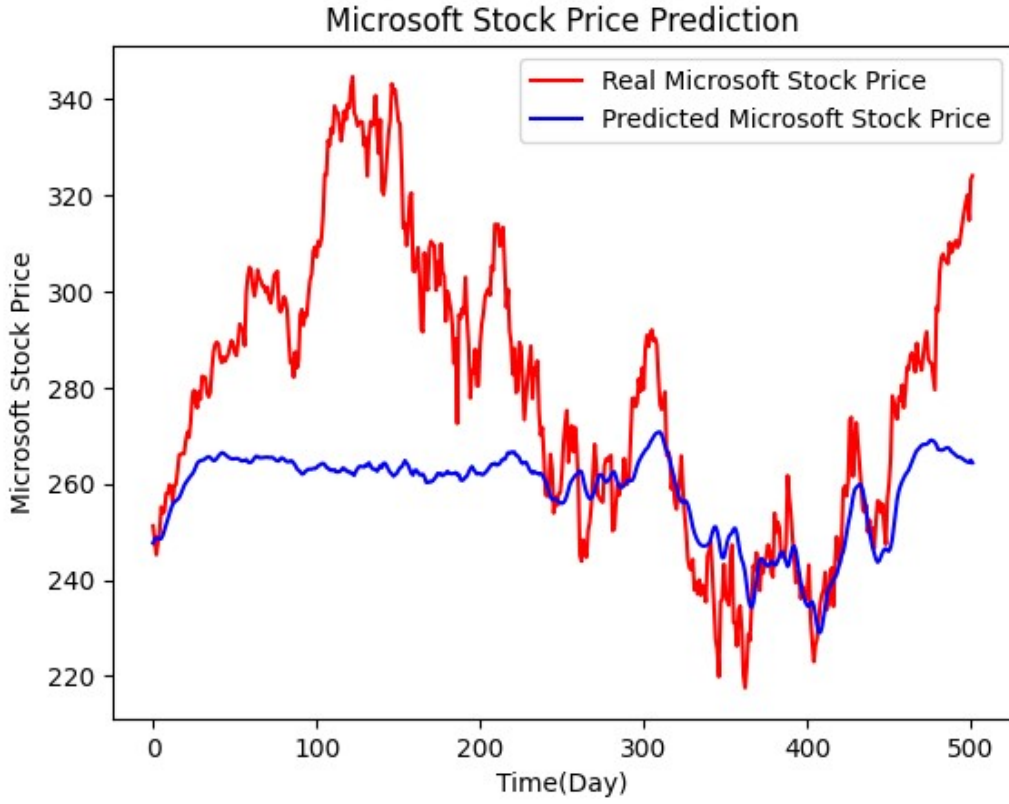
```
[ ] plt.plot(real_stock_price, color = 'red' , label = 'Real Microsoft Stock Price')
plt.plot(predicted_stock_price, color = 'blue' , label = 'Predicted Microsoft Stock Price')
plt.title('Microsoft Stock Price Prediction')
plt.xlabel('Time(Day)')
plt.ylabel('Microsoft Stock Price')
plt.legend()
plt.show()
```

-Gerçek ve tahmini stok fiyatlarının stok-zaman grafiği elde edildi.





## 5. SONUÇLAR



-Gerçek değerler çok keskin yükseliş ve düşüş miktarına sahipken tahmini değerler daha dengeli şekilde ilerlemiş.Yükselme ve düşüş miktarının ne kadar olduğu konusunda başarılı olamasa da yükselme ve düşme trendlerinin ne zaman gerçekleşeceğini öngörebilmiş.Bu durum al-sat konusunda kullanıcıya yarar sağlar.Ne zaman al-sat yapacağını bilmek kar miktarını bilmekten daha önemlidir.

## 6. ÖNERİ VE TARTIŞMALAR

Loss oranı daha 13-14.epoch'ta 0,0010-0,0012'ye geldiğinden bir değişikliğe gerek yok gibi duruyor.Ancak veri seti büyütülebilir.Borsayı etkileyen çok fazla etken olduğundan yüzde yüz sağlıklı bir tahmin yapmak pek mümkün değil.



## KAYNAKLAR

<https://finance.yahoo.com/quote/MSFT/history?period1=1622419200&period2=1685404800&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true>

<https://mfakca.medium.com/lstm-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-326866fd8869>

<https://medium.com/deep-learning-turkiye/rnn-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-9e5d572689e1>

## KONTROL LİSTESİ

	EVET /HAYIR
Raporunuzu şablonda belirtildiği gibi hazırladınız mı?	Evet
Çalışmanızın sonuçlarını (print screen) rapora eklediniz mi?	Evet
Rapor dosyanızı şablondaki gibi yeniden adlandırdınız mı?	Evet
Raporu sisteme yüklediniz mi?	Evet
Kodları sisteme yüklediniz mi?	Evet