

Hazırlayanlar
Rabia Kaşıkçı, Hakan Everdi

Tarih:
22.07.21

Akbank Teknoloji Okuryazarlığı Bootcamp

Özet:

Bu rapor, hisse senedi piyasalarının analizini ve tahminini Python programlama dili kullanarak gerçekleştirilmiş bir çalışmayı sunmaktadır. Çalışma, çeşitli zaman serisi modelleme tekniklerini kullanarak hisse senedi fiyatlarını tahmin etmeyi amaçlamakta ve her bir yöntemin performansını değerlendirmektedir.

Anahtar Kelimeler:

Hisse Senedi, Zaman Serisi Analizi, Python, ARIMA, LSTM, Holt-Winters

1. GİRİŞ

Bu rapor, belirli bir borsa endeksi üzerindeki hisse senedi verilerini analiz etmek ve tahminlerde bulunmak için Python kodu kullanılarak yapılan çalışmayı özetlemektedir. Analiz, temel değerlerin görselleştirilmesi, çeşitli modelleme tekniklerinin uygulanması ve bu modellerin performanslarının değerlendirilmesini kapsamaktadır. Kullanılan teknikler arasında Doğrusal Regresyon (LR), Otoregresif Entegre Hareketli Ortalama (ARIMA), Uzun Kısa Süreli Bellek (LSTM) ve Holt-Winters yöntemleri bulunmaktadır. Projenin amacı, BIST 30 endeksindeki hisse senetlerinin geçmiş fiyat verilerini kullanarak, gelecekteki fiyat hareketlerini tahmin edebilecek etkili modeller geliştirmektir. Bu amaç doğrultusunda, çeşitli makine öğrenimi algoritmaları ve zaman serisi analiz teknikleri kullanılmıştır.

2. VERİ TOPLAMA ve İŞLEME

2.1 Veri Toplama ve Hazırlık

Bu projede kullanılan veriler, Yahoo Finance platformundan alınmış olup, Borsa İstanbul (BIST) 30 endeksindeki hisse senetlerine aittir. Veriler, Python programlama dili ve yfinance kütüphanesi kullanılarak elde edilmiştir.

2.1.1 Veri Kaynağı

Projedeki veri kaynağı, Yahoo Finance platformudur. Bu platformdan hisse senedi fiyatları ve işlem hacmi gibi önemli finansal veriler çekilmiştir. Veri seti, 2023-2024 yılları arasındaki tarihleri kapsamaktadır.

2.1.2 Veri İşleme ve Temizleme

Veri toplama süreci tamamlandıktan sonra, veri setinde eksik veya hatalı değerlerin bulunması durumunda uygun yöntemlerle düzeltme ve temizleme işlemleri yapılmıştır. İşleme adımları şu şekilde özetlenebilir:

- **Verilerin İndirilmesi:** Yfinance kütüphanesi kullanılarak hisse senedi fiyat verileri indirilmiştir.
- **Verilerin Düzenlenmesi:** İndirilen veriler, pandas kütüphanesi aracılığıyla işlenmiş, tarih sütunları datetime formatına dönüştürülmüş ve eksik veriler ortalama değerlerle doldurulmuştur.
- **Verilerin Ayrılması:** Eğitim ve test veri setleri oluşturulmuş, veriler %70 eğitim ve %30 test olacak şekilde ayrılmıştır.



Resim 1. Veriler HeatMap

2.2 Veri Ön İşleme

Toplanan veriler üzerinde eksik değerler doldurulmuş ve tarih sütunu uygun formatta dönüştürülmüştür. Veriler, zaman serisi analizine uygun hale getirilmiştir. Ayrıca, bazı modellerin gerektirdiği formatlara dönüştürülmüş ve ölçeklendirilmiştir.

3. MAKİNE ÖĞRENMESİ MODELLEMESİ

3.1 Doğrusal Regresyon (LR)

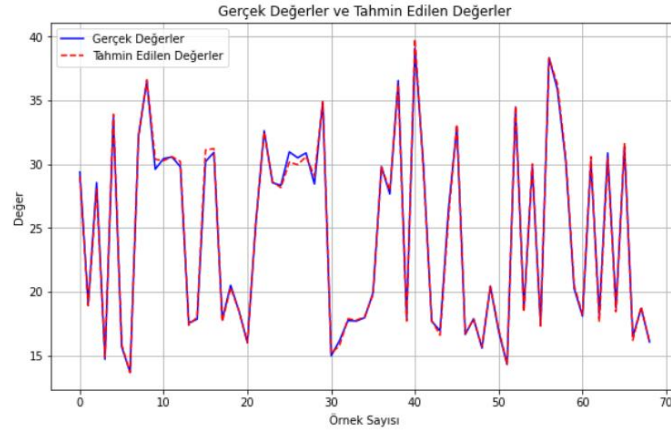
Doğrusal regresyon, hisse senedi fiyatlarını tahmin etmek için kullanılan temel bir modeldir. Model, tarihsel verilerden öğrenir ve test verileri üzerinde tahminlerde bulunur. Bu çalışma kapsamında kullanılan kod aşağıda verilmiştir.

```
# Doğrusal Regresyon
model_lnr = LinearRegression()
model_lnr.fit(x_train, y_train)
y_pred_lr = model_lnr.predict(x_test)

MSE_lr = round(mean_squared_error(y_test, y_pred_lr), 3)
RMSE_lr = round(np.sqrt(MSE_lr), 3)
MAE_lr = round(mean_absolute_error(y_test, y_pred_lr), 3)
MAPE_lr = round(mean_absolute_percentage_error(y_test, y_pred_lr), 3)
R2_Score_lr = round(r2_score(y_test, y_pred_lr), 3)
```

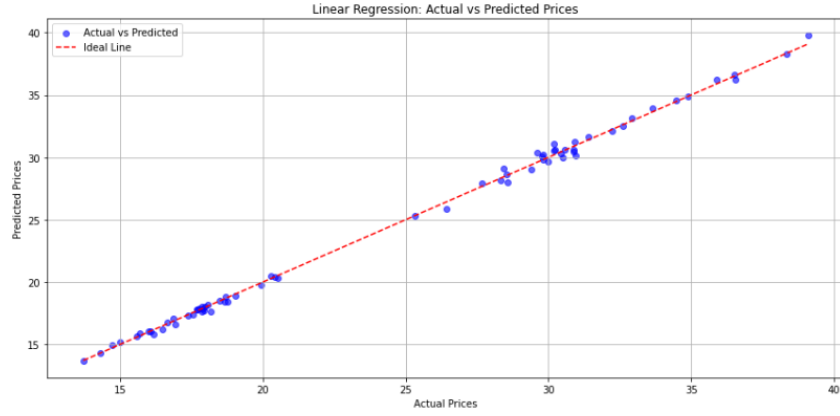
Resim 2. Lineer Reg. Kodu

Tahminler sonucunda, karşılaştırma yapılmıştır. Gerçek değerler ile tahmin değerleri arasında ilişkiye bakılıp modelin performansı incelenmiştir.



Resim 3. Lineer Reg. Gerçek ve Tahmin Değerleri

Bu incelemeden sonra tahminler ve gerçek değerler arasındaki dağılım incelenmiştir. İncelemek için elde edilen grafik aşağıda verilmiştir.



Resim 4. Lineer Reg. Gerçek ve Tahmin Değerleri dağılımı

3.2 ARIMA Modeli

ARIMA modeli, zaman serisi verileri için yaygın olarak kullanılan bir modeldir. Bu model, trend ve mevsimselliği göz önünde bulundurarak gelecekteki değerleri tahmin eder. Model için kullanılan kod aşağıda verilmiştir.

```
# ARIMA Modeli
close_value_arma = prices_train_without_last_year['Close']
p_value = adfuller(close_value_arma)[1]
if p_value > 0.05:
    close_value_arma = close_value_arma.diff().dropna()

model_arma = auto_arma(close_value_arma, seasonal=False, stepwise=True, trace=True)
model_fit = model_arma.fit(close_value_arma)
forecast = model_fit.predict(n_periods=10)

y_test_arma = prices_train_without_last_year['Close'][-10:]
MSE_arma = round(mean_squared_error(y_test_arma, forecast), 3)
RMSE_arma = round(np.sqrt(MSE_arma), 3)
MAE_arma = round(mean_absolute_error(y_test_arma, forecast), 3)
MAPE_arma = round(mean_absolute_percentage_error(y_test_arma, forecast), 3)
R2_Score_arma = round(r2_score(y_test_arma, forecast), 3)
```

Resim 5. Arıma Model Kod

ARIMA modelinin sonuçları incelendiğinde, istenilen performansın elde edilmediği gözlemlenmiştir. Bu nedenle, sonraki adımlarda bu model kullanılmamıştır.

3.3 LSTM Modeli

Uzun Kısa Süreli Bellek (LSTM) ağları, özellikle zaman serisi tahminleri için etkili olan derin öğrenme modelleridir. Bu model, geçmiş verileri kullanarak gelecekteki değerleri tahmin eder.

Model için kullanılan kod aşağıda verilmiştir.

```
# LSTM Modeli
data_lstm = prices_train['Close'].values.reshape(-1, 1)
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data_lstm)

def create_dataset(data, time_step=1):
    X, Y = [], []
    for i in range(len(data) - time_step - 1):
        X.append(data[i:(i + time_step), 0])
        Y.append(data[i + time_step, 0])
    return np.array(X), np.array(Y)

time_step = 10
X, y = create_dataset(scaled_data, time_step)
train_size = int(len(X) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

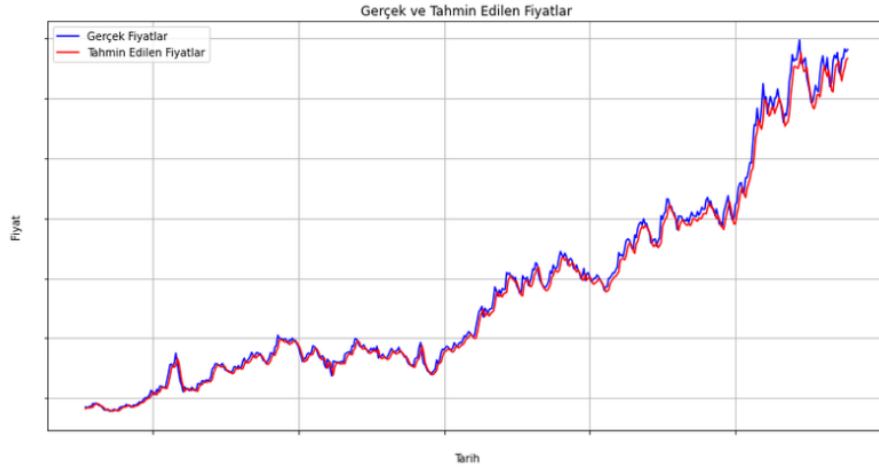
model_lstm = Sequential()
model_lstm.add(LSTM(100, return_sequences=True, input_shape=(time_step, 1)))
model_lstm.add(Dropout(0.2))
model_lstm.add(LSTM(100, return_sequences=False))
model_lstm.add(Dropout(0.2))
model_lstm.add(Dense(50))
model_lstm.add(Dense(1))
model_lstm.compile(optimizer='adam', loss='mean_squared_error')

history = model_lstm.fit(X_train, y_train, batch_size=32, epochs=80, validation_split=0.1, verbose=1,
                        callbacks=[EarlyStopping(monitor='val_loss', patience=10)])

predictions = model_lstm.predict(X_test)
predictions = scaler.inverse_transform(predictions)
y_test = scaler.inverse_transform(y_test.reshape(-1, 1))
```

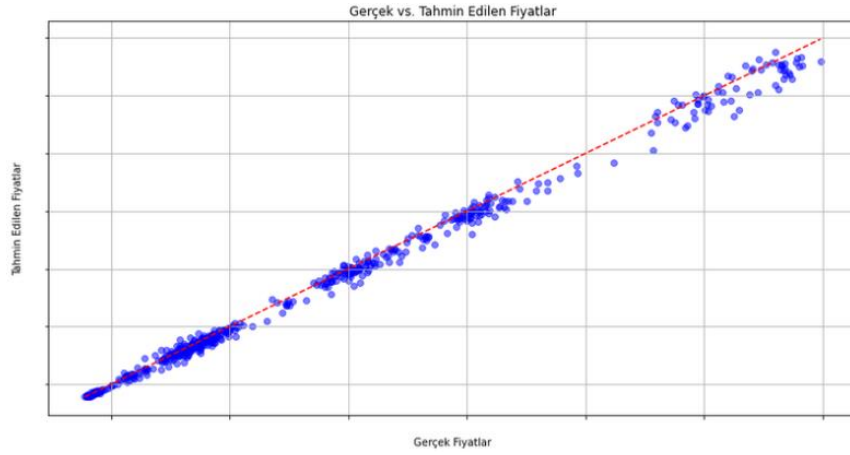
Resim 6. LSTM Model Kod

Modelleme sonucunda elde edilen tahminler ile gerçek değerler karşılaştırılmış ve sonuçlar grafikler aracılığıyla görselleştirilmiştir.



Resim 7. LSTM Model Gerçek ve Tahmin Değerleri Arasındaki İnceleme

İnceleme sonucunda modeldeki gerçek ve tahmin değerlerin nasıl dağıldığına bakılmıştır.



Resim 8. LSTM Model Gerçek ve Tahmin Değerleri Arasındaki Dağılımı

3.4 ExponentialSmoothing Yöntemi

ExponentialSmoothing yöntemleri, trend ve mevsimselliği modelleyerek gelecekteki değerleri tahmin eder. Bu yöntem, özellikle yıllık döngülerin olduğu veriler için uygundur. Bu model için en uygun parametreler dataset içinde incelenerek bulunmuştur.

```
# Modeli oluřtur
prices_train.set_index('Date', inplace=True)

# Modeli oluřtur
model_ets = ExponentialSmoothing(prices_train['Close'], trend=None, seasonal='mul', seasonal_periods=30)
results_ets = model_ets.fit()
forecast_steps = 10
forecast_ets = results_ets.forecast(steps=forecast_steps)

# Son 10 gn iin gerek ve tahmin edilen deęerleri karřılařtır
y_true = prices_train['Close'][-forecast_steps:]
y_pred = forecast_ets.values

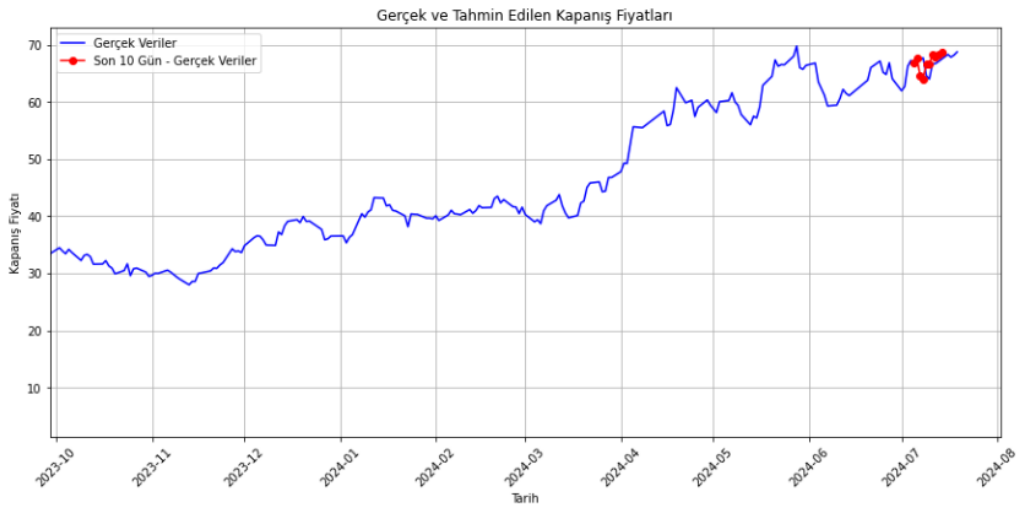
# İndeksleri oluřturma
forecast_index = pd.date_range(start=prices_train.index[-1] + pd.Timedelta(days=1), periods=forecast_steps, freq='B')

# Karřılařtırma iin DataFrame oluřturma
comparison_df = pd.DataFrame({'True': y_true.values, 'Predicted': y_pred}, index=prices_train.index[-forecast_steps:])
print(comparison_df)

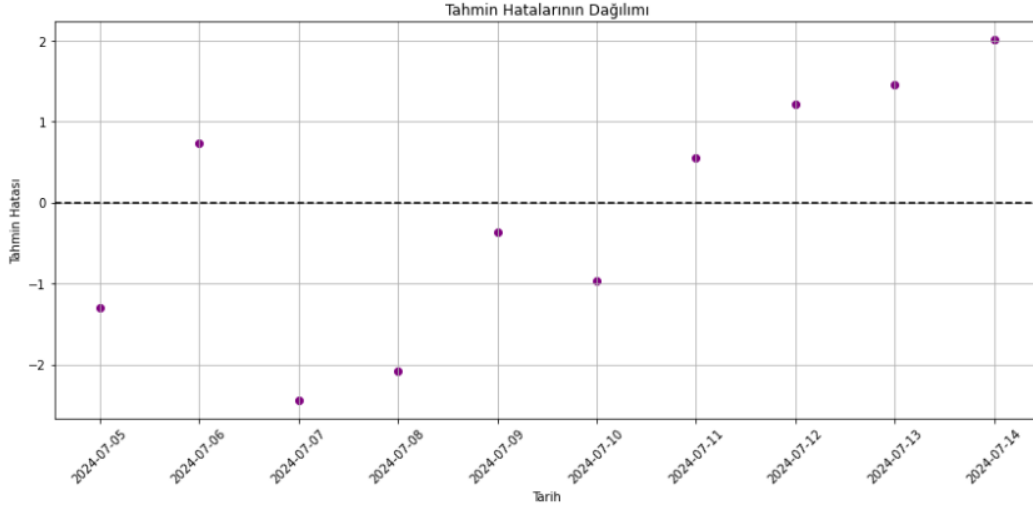
# Performans metriklerini hesapla
mse_ets = mean_squared_error(y_true, y_pred)
mae_ets = mean_absolute_error(y_true, y_pred)
rmse_ets = np.sqrt(mse_ets)
r2_ets = r2_score(y_true, y_pred)
```

Resim 9. ExponentialSmoothing Model Kod

Modelleme sonucunda elde edilen tahmin e gerek deęerler incelenmiř ve grafik elde edilmiřtir. İnceleme sonucunda modeldeki gerek ve tahmin deęerlerin nasıl daęıldığına bakılmıřtır.



Resim 10. ExponentialSmoothing gerek ve tahmin deęerleri



Resim 10. ExponentialSmoothing gerçek ve tahmin dağılımı

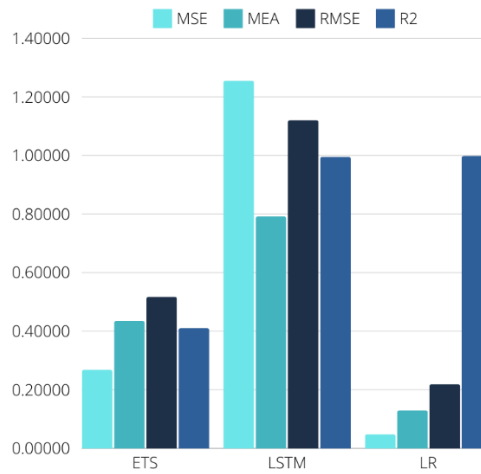
Verilerin görselleştirilmesi, hisse senedinin tarihsel performansını anlamak için yapılmıştır. Açılış, yüksek, düşük, kapanış fiyatları ve işlem hacmi gibi değerler zaman serisi grafikleri olarak sunulmuştur. Bu grafikleri oluşturan kod parçacıkları aşağıda özetlenmiştir:

4. SONUÇLAR

4.1 Model Performansı

Modellerin performansları çeşitli metriklerle değerlendirilmiştir:

- Doğrusal Regresyon: Ortalama Kare Hatası (MSE) hesaplanmıştır.
- ARIMA Modeli: Tahminlerin gerçek verilerle karşılaştırılması yapılmıştır.
- LSTM Modeli: Ortalama Kare Hatası (MSE) hesaplanmıştır ve sonuçlar karşılaştırılmıştır.
- ExponentialSmoothing Yöntemi: Tahmin sonuçları ve MSE raporlanmıştır.



Resim 11. Performans Metriklerin İncelenmesi

4.2 Görselleştirme

Tahmin sonuçları, hisse senedinin gelecekteki performansını göstermek için görselleştirilmiştir. Bu görselleştirmeler, kullanıcıların tahminlerin doğruluğunu görsel olarak değerlendirmelerine yardımcı olmuştur.

5.GELECEK TAHMİNİ

Modellerin incelemesi sonucunda, gelecekteki tahminler için en uygun modellerin seçimi üzerine çalışılmıştır. Liner Regresyon, özneliliklere dayalı olarak çok iyi performans göstermiş olmasına rağmen, gelecekteki tahminler için yeterli bulunmamıştır. Modelin gelecekteki değer tahmini sırasında overfitting yaptığı ve kendini tekrar ettiği gözlemlenmiştir.

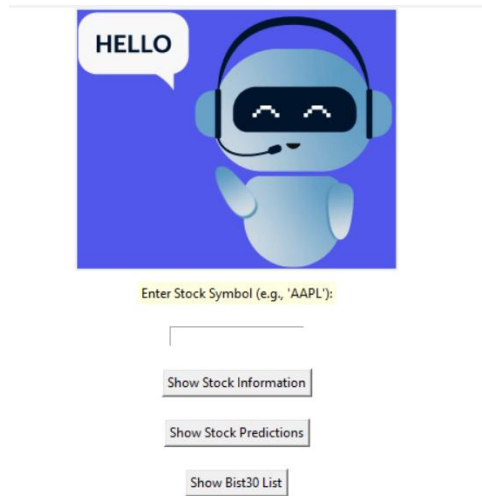
ARIMA modeli ise performans açısından beklentileri karşılamadığı için bu çalışmada kullanılmamıştır.

Sonuç olarak, gelecekteki 10 gün tahmini için LSTM ve ETS modelleri seçilmiştir. Burada, hangi algoritmanın veri seti için daha iyi performans gösterdiği temel alınarak tahminler yapılmıştır.

6.ARAYÜZ

Modelin kullanımını artırmak amacıyla bir arayüz tasarımı yapılmıştır. Bu arayüzde kullanıcıdan, incelemek ve bilgi almak istediği sembolün adını girmesi istenmektedir. Arayüzdeki ana fonksiyonlar şunlardır:

- **Show Stock Information:** Bu seçenek, girilen hisse senedine ait bugüne kadar olan açık (Open), yüksek (High), düşük (Low), kapanış (Close) ve işlem hacmi (Volume) verilerini getirmektedir.
- **Show Stock Prediction:** Bu seçenek, seçilen hisse senedi için yapılan gelecekteki tahminleri sunmaktadır.
- **Show BIST30 List:** Bu seçenek, kullanılabilecek sembol listelerini göstermektedir.



Resim 12. Arayüz Resmi

7. Tartışma

Bu çalışma, çeşitli modelleme tekniklerinin hisse senedi tahminlerinde nasıl kullanılabileceğini göstermektedir. Modellerin her birinin avantajları ve sınırlamaları bulunmaktadır. Doğrusal regresyon, basit ve anlaşılır bir model sunarken, LSTM ve ARIMA gibi daha karmaşık modeller daha hassas tahminler sağlayabilmektedir.

8. Gelecekteki Çalışmalar

Bu projede elde edilen bulgular, hisse senedi fiyatlarının tahmin edilmesinde makine öğrenimi ve zaman serisi analiz tekniklerinin etkinliğini ortaya koymaktadır. Gelecekteki çalışmalarda aşağıdaki alanlara odaklanılması önerilmektedir:

- **Veri Kapsamının Genişletilmesi:** Daha geniş veri setleri ve uzun zaman dilimlerini kapsayan analizler yapılabilir.
- **Algoritma Çeşitlendirmesi:** Diğer makine öğrenimi algoritmaları ve derin öğrenme modellerinin değerlendirilmesi, tahmin performansını artırabilir.
- **Ek Veri Entegrasyonu:** Makroekonomik göstergeler ve şirketlerin finansal raporları gibi ek veri kaynaklarının kullanılması, modellerin doğruluğunu ve güvenilirliğini artırabilir.

9. REFERANSLAR

- 1) Neptune.ai. (n.d.). Time Series Prediction vs Machine Learning. Retrieved from <https://neptune.ai/blog/time-series-prediction-vs-machine-learning>
- 2) Neptune.ai. (n.d.). ARIMA vs Prophet vs LSTM. Retrieved from <https://neptune.ai/blog/arima-vs-prophet-vs-lstm>
- 3) Tran Phuoc, P., Pham Thi Kim Anh, T., Phan Huy Tam, T., & Chien V. Nguyen, N. (n.d.). Applying machine learning algorithms to predict the stock price trend in the stock market – The case of Vietnam.
- 4) Machine Learning Türkiye. (2021). Zaman Serileri Analizi 2. Retrieved from <https://medium.com/machine-learning-t%C3%BCrkiye/zaman-serileri-analizi-2-e4f6a750cd5d>