

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



UZUN VE KISA TATİL GÜNLERİ İÇİN TRAFİK TAHMİNİ:
ÖZEL YAKLAŞIMLAR

19011093 – Fatih ECEVİT

19011042 – Yusuf Mert ÇELİKARSLAN

BİLGİSAYAR PROJESİ

Danışman
Doç. Dr. Mehmet Amaç GÜVENSAN

Ocak, 2024

TEŞEKKÜR

Bir projenin başarılı bir şekilde hayata geçirilmesi, sadece teknik bilgi ve beceriyle değil, aynı zamanda yürekten verilen destek ve vizyonla mümkündür. Bu nedenle, Sayın Doç.Dr. Mehmet Amaç GÜVENSAN'ın projemizin gerçekleşmesindeki katkılarından dolayı teşekkür ederiz.

Projemizin her aşamasında bize verdığınız değerli geri dönüşler, vizyonumuzu şekillendirdi ve bizi daha da ileri taşıdı. Ayrıca yol göstericiliği, bize sadece akademik bilgi değil, aynı zamanda sektörde nasıl başarılı olabileceğimiz konusunda da kıymetli ipuçları sundu. Bu vesileyle, sadece projemize katkılarınız için değil, bize kattığınız değerler, öğretikleriniz için de teşekkür etmek istiyoruz.

Umarız ki gelecekte de sizinle çalışma fırsatı bulabiliriz.

Ayrıca bu süreçte manevi desteklerini her an hissettiğimiz ailemize ve arkadaşlarımıza teşekkürü bir borç biliriz.

Fatih ECEVİT
Yusuf Mert ÇELİKARSLAN

İÇİNDEKİLER

KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
TABLO LİSTESİ	ix
ÖZET	x
ABSTRACT	xii
1 Giriş	1
1.1 Trafik Tahmin Probleminin Tarihçesi	1
1.2 Trafik Tahmin Probleminde Kullanılan Veri Türleri	1
1.3 Trafik Tahmin Modelleri ve Hata Metrikleri	2
1.4 Karşılaşılan Sorunlar	2
2 Ön İnceleme	3
2.1 Zaman Serileri	3
2.2 Probleme Farklı Yaklaşımlar	4
2.3 Veri	4
3 Fizibilite	5
3.1 Teknik Fizibilite	5
3.1.1 Yazılım Fizibilitisi	5
3.1.2 Donanım Fizibilitesi	6
3.2 Ekonomik Fizibilite	7
3.3 Legal Fizibilite	7
3.4 İşgücü ve Zaman Fizibilitesi	8
4 Sistem Analizi	9
4.1 Veri Seti Analizi	9
4.1.1 Mevsimsel Ayrışma	9
4.1.2 Geçmiş Verilerle Korelasyon	10
4.1.3 Özellik Seçimi ve Farklı Bayramlardaki Kullanımları	10

4.1.4	Test Seti Olarak Kullanılan Segmentler ve Özellikleri	12
4.2	Uzun Tatiller	17
4.2.1	Kurban Bayramı	17
4.2.2	Ramazan Bayramı	18
4.3	Kısa Tatiller	19
4.3.1	Haftaiçine Denk Gelen Tatiller	22
4.3.2	Haftasonuna Denk Gelen Tatiller	23
4.3.3	Farklı Karakteristik İzleyen Tatiller	23
5	Sistem Tasarımı	25
5.1	Algoritmalar ve Model Eğitimi	25
5.2	Karar Ağacı Modelleri	25
5.3	Derin Öğrenme Modelleri	27
5.4	Birleşik Model	28
6	Uygulama	29
6.1	Fast Fourier Transform Ön İşlemesi	29
6.2	Kayıp Değerleri Doldurma	32
6.3	Random Forest Regressor	32
6.4	XGBoost	35
6.5	Gradient Boosting	38
6.6	CatBoost	39
6.7	LGBMRegressor	42
6.8	DNNs	45
6.9	Ensemble	48
6.10	Hata Oranları	49
6.11	Deneysel Sonuçlar	53
6.12	Performans Analizi	58
7	Sonuç	60
	Referanslar	63
	Özgeçmiş	64

KISALTMA LİSTESİ

ANN	Artifical Neural Networks
CNN	Convolutional Neural Networks
DFT	Discrete Fourier Transform
GBM	Gradient Boosting Model
LGBM	Light Gradient Boosting Model
LSTM	Long-short Term Memory
MA	Moving Average
MAD	Mean Accuracy Decrease
MDI	Mean Decrease in Impurity
RFR	Random Forest Regressor
RGSCV	Randomized Grid Search CV
RNN	Artifical Neural Networks
SVR	Support Vector Regression

ŞEKİL LİSTESİ

Şekil 4.1 5 dk. aralıklarla	10
Şekil 4.2 Günlük	10
Şekil 4.3 Korelasyon	11
Şekil 4.4 Kısmi Korelasyon	11
Şekil 4.5 Değişken Önem Yatay Bar Grafiği	12
Şekil 4.6 MDI Özellik Önemi	13
Şekil 4.7 Permüstasyon Özellik Önemi	13
Şekil 4.8 E-5 Ana Arter	14
Şekil 4.9 Kurbağalıdere Ana Arter [7]	15
Şekil 4.10 Kurbağalıdere Yol Çalışması Sonucu Etkilenen 431 Segmenti	15
Şekil 4.11 Şehir İçi Yol	16
Şekil 4.12 Bağdat Caddesi 2018 Kurban Bayramı	18
Şekil 4.13 Bağdat Caddesi 2019 Kurban Bayramı	18
Şekil 4.14 Bağdat Caddesi 2018 Ramazan Bayramı	19
Şekil 4.15 Bağdat Caddesi 2019 Ramazan Bayramı	19
Şekil 4.16 Bağdat Caddesi 2018 Ramazan Bayramı	22
Şekil 4.17 1 Ocak 2018	24
Şekil 4.18 1 Ocak 2019	24
Şekil 5.1 Sistem Tasarım Modeli	28
Şekil 6.1 Düşük Bant Arahklı FFT	30
Şekil 6.2 Yüksek Bant Aralıklı FFT	30
Şekil 6.3 Düşük Frekans Filtreleme Uygulanmış Veri ve Gerçek Veri	31
Şekil 6.4 Yüksek Frekanslı Veri	31
Şekil 6.5 1 Mayıs 2019 Random Forest	34
Şekil 6.6 30 Ağustos 2019 Random Forest	34
Şekil 6.7 23 Nisan 2019 Random Forest	34
Şekil 6.8 29 Ekim 2019 Random Forest	34
Şekil 6.9 19 Mayıs 2019 Random Forest	34
Şekil 6.10 15 Temmuz 2019 Random Forest	34
Şekil 6.11 Ramazan Bayramı 2019 Random Forest	35
Şekil 6.12 Kurban Bayramı 2019 Random Forest	35

Şekil 6.13 1 Mayıs 2019 XGBoost	37
Şekil 6.14 30 Ağustos 2019 XGBoost	37
Şekil 6.15 23 Nisan 2019 XGBoost	37
Şekil 6.16 29 Ekim 2019 XGBoost	37
Şekil 6.17 19 Mayıs 2019 XGBoost	37
Şekil 6.18 15 Temmuz 2019 XGBoost	37
Şekil 6.19 Ramazan Bayramı 2019 XGBoost	38
Şekil 6.20 Kurban Bayramı 2019 XGBoost	38
Şekil 6.21 1 Mayıs 2019 Gradient Boosted	39
Şekil 6.22 30 Ağustos 2019 Gradient Boosted	39
Şekil 6.23 23 Nisan 2019 Gradient Boosted	39
Şekil 6.24 29 Ekim 2019 Gradient Boosted	39
Şekil 6.25 19 Mayıs 2019 Gradient Boosted	40
Şekil 6.26 15 Temmuz 2019 Gradient Boosted	40
Şekil 6.27 Ramazan Bayramı 2019 Gradient Boosted	40
Şekil 6.28 Kurban Bayramı 2019 Gradient Boosted	40
Şekil 6.29 1 Mayıs 2019 Catboost	42
Şekil 6.30 30 Ağustos 2019 Catboost	42
Şekil 6.31 23 Nisan 2019 Catboost	42
Şekil 6.32 29 Ekim 2019 Catboost	42
Şekil 6.33 19 Mayıs 2019 Catboost	43
Şekil 6.34 15 Temmuz 2019 Catboost	43
Şekil 6.35 Ramazan Bayramı 2019 Catboost	43
Şekil 6.36 Kurban Bayramı 2019 Catboost	43
Şekil 6.37 1 Mayıs 2019 LightGBM	44
Şekil 6.38 30 Ağustos 2019 LightGBM	44
Şekil 6.39 23 Nisan 2019 LightGBM	45
Şekil 6.40 29 Ekim 2019 LightGBM	45
Şekil 6.41 19 Mayıs 2019 LightGBM	45
Şekil 6.42 15 Temmuz 2019 LightGBM	45
Şekil 6.43 Ramazan Bayramı 2019 LightGBM	46
Şekil 6.44 Kurban Bayramı 2019 LightGBM	46
Şekil 6.45 1 Mayıs 2019 DNNs	47
Şekil 6.46 30 Ağustos 2019 DNNs	47
Şekil 6.47 23 Nisan 2019 DNNs	47
Şekil 6.48 29 Ekim 2019 DNNs	47
Şekil 6.49 19 Mayıs 2019 DNNs	47
Şekil 6.50 15 Temmuz 2019 DNNs	47
Şekil 6.51 1 Mayıs 2019 Ensemble	48

Şekil 6.52 30 Ağustos 2019 Ensemble	48
Şekil 6.53 23 Nisan 2019 Ensemble	48
Şekil 6.54 29 Ekim 2019 Ensemble	48
Şekil 6.55 19 Mayıs 2019 Ensemble	48
Şekil 6.56 15 Temmuz 2019 Ensemble	48
Şekil 6.57 Ramazan Bayramı 2019 Ensemble	49
Şekil 6.58 Kurban Bayramı 2019 Ensemble	49
Şekil 6.59 Eğitim Setinde MDI kullanılması	53
Şekil 6.60 Test Setinde Permüstasyon Kullanılması	53
Şekil 6.61 Eğitim Setinde MDI kullanılması	53
Şekil 6.62 Test Setinde Permüstasyon Kullanılması	53
Şekil 6.63 23 Nisan 2019	54
Şekil 6.64 1 Mayıs	54
Şekil 6.65 19 Mayıs	54
Şekil 6.66 15 Temmuz	54
Şekil 6.67 30 Ağustos	54
Şekil 6.68 29 Ekim	54
Şekil 6.69 Eğitim Seti Nesnesi	55
Şekil 6.70 Kadıköy (312 Numaralı Segment)	55
Şekil 6.71 Maltepe (321 Numaralı Segment)	55
Şekil 6.72 Bakırköy (420 Numaralı Segment)	55
Şekil 6.73 Üsküdar (614 Numaralı Segment)	55
Şekil 6.74 Beşiktaş (64 Numaralı Segment)	55
Şekil 6.75 Kadıköy (312 Numaralı Segment)	56
Şekil 6.76 Maltepe (321 Numaralı Segment)	56
Şekil 6.77 Bakırköy (420 Numaralı Segment)	56
Şekil 6.78 Üsküdar (614 Numaralı Segment)	56
Şekil 6.79 Beşiktaş (64 Numaralı Segment)	56
Şekil 6.80 Random Forest	57
Şekil 6.81 XGB Regressor	57
Şekil 6.82 CatBoost	57
Şekil 6.83 Ensemble	57
Şekil 6.84 Random Forest	57
Şekil 6.85 XGB Regressor	57
Şekil 6.86 CatBoost	57
Şekil 6.87 Ensemble	57

TABLO LİSTESİ

Tablo 4.1 Tatil Günlerinin Türlerine Özel Tahmin Setleri	20
Tablo 6.1 23 Nisan 2019 Hata Oranları Set-1: Aynı Yıl Geçen Hafta Set-2: Set1 + Geçmiş yıl milli bayramlar ve bir önceki günlerinin birleşimi Set-3: Set1 + FFT Set-4: Set2 + FFT	49
Tablo 6.2 1 Mayıs 2019 Hata Oranları Set-1: Geçen Sene Tek Günlük Bayramlar Set-2: Set1 + Aynı Sene Geçen Hafta Set-3: Set2 + 23 Nisan 2019 Set-4: Set3 + FFT	50
Tablo 6.3 19 Mayıs 2019 Hata Oranları Set-1: Geçen Sene Tek Günlük Tatiller ve önceki bir gün Set-2: Aynı Yıl Bir Hafta Önce Aynı Gün Set-3: Set1 + FFT Set-4: Set2 + FFT	50
Tablo 6.4 15 Temmuz 2019 Hata Oranları Set-1: Geçen Sene Tek Günlük Tatiller ve önceki bir gün Set-2: Set1 + FFT Set-3: Set1 + Aynı yıl önceki haftasonu ve önceki hafta aynı gün + FFT	50
Tablo 6.5 30 Ağustos 2019 Hata Oranları Set-1: Geçen Sene Tek Günlük Tatiller ve önceki bir gün + Geçen Hafta Set-2: Set1 + FFT Set-3: Set2 + Aynı Sene Tek Günlük Bayramlar(23 Nisan, 1 Mayıs) . .	51
Tablo 6.6 29 Ekim 2019 Hata Oranları Set-1: Geçen Sene Tek Günlük Tatiller ve önceki günleri Set-2: Set1 + Aynı Sene Geçen Hafta Set-3: 2 Hafta 2018 + Önceki 2 Hafta 2019 Set-4:,Set3 + FFT .	51
Tablo 6.7 Kurban Bayramı 2019 Hata Oranları Set-1: Geçen Sene Kurban Bayramı 2 Hafta 13 - 27 Ağustos Set-2: Set1 + Ramazan Bayramı Set-3: Set1 + Aynı Sene Önceki Hafta sonu + FFT	52
Tablo 6.8 Ramazan Bayramı 2019 Hata Oranları Set-1: Geçen Sene Ramazan Bayramı 2 Hafta 11 - 25 Haziran Set-2 : Set1 + Geçen Sene Haftasonu + Aynı Sene Bir Önceki Haftasonu Set-3: Set1 + Kurban Bayramı	52
Tablo 6.9 İstanbul Geneli Segmentlerin MAPE Ortalaması	58
Tablo 6.10 Örnek Segmentlere Göre MAPE Hata Oranı	59
Tablo 7.1 Önerilen Model ve Halihazırdağı diğer modellerin MAPE oranı açısından karşılaştırılması	61

ÖZET

UZUN VE KISA TATİL GÜNLERİ İÇİN TRAFİK TAHMİNİ: ÖZEL YAKLAŞIMLAR

Fatih ECEVİT

Yusuf Mert ÇELİKARSLAN

Bilgisayar Mühendisliği Bölümü

Bilgisayar Projesi

Danışman: Doç. Dr. Mehmet Amaç GÜVENSAN

Trafik analizi ve araç hız tahmini günümüzde bireylerin günlük yaşantısına önemli ölçüde etki etmektedir. 21. Yüzyılın başlangıcından günümüze dünya nüfusundaki yaklaşık %33'lük artış ve araç sayılarındaki yaklaşık %43'lük artış hem insan hem araç sayılarındaki artış hızının yüksek olduğunu, insan ve araç sayılarındaki bu artışın da beraberinde trafik yoğunluğu getirdiği görülmektedir. Analiz ve tahmin süreçlerinin başarılı bir şekilde ilerlemesi ve sonuçlanması için trafikteki araçların hız verilerinin eksiksiz ve doğru olması gerekmektedir. Bu projede İstanbul sınırları içerisinde hız tahmini yapılması planlanıp, İstanbul Büyükşehir Belediyesi tarafından tutulan veriler kullanılarak sayısal verilerin görselleştirilmesi, analiz edilmesi ve yorumlanması adımları sonrasında uzun ve kısa tatil günleri için trafik tahmini yapan bir model geliştirilmiştir. Modelde önişlem metodu olarak Discrete Fourier Transform (DFT) algoritmasının hesaplama maliyeti daha düşük olan bir versionu olan Fast Fourier Transform(FFT) kullanılmıştır. Ön işlemenden geçirilen veri analiz adımdında incelenerek doğru kesme frekansı belirlenmiş ve tahmin edilecek günlere özel eğitim setleri oluşturulmuştur. Oluşturulan eğitim setleri ilk önce parametre değişimine toleransı nedeniyle Random Forest modelinde denenmiş ve Ortalama Safsızlık Düşüsü (MDI) ve Tam Model Permütasyonu (MAD) algoritmalarıyla öznitelik önemi hesaplanarak grafikleştirilmiş, ortaya çıkan sonuçlar yorumlanarak öznitelik seçimi yapılmıştır. Bu seçim sonrasında diğer GradientBoosting, CatBoost, LGBM, XGBRegressor, DNN modelleri de denenerek aralarında daha iyi sonuç veren modeller seçilerek FFT-Ensemble adı verilen yeni bir model ortaya çıkarılmıştır. Yeni ortaya

çıkan model sonucunda sadece kısa ve uzun tatil günlerini içeren tahminde ortalama %11.22'lik MAPE oranı elde edilerek literatürdeki[1] %11.60'lık hata oranı geçilerek 0.38'lik düşüş sağlanmıştır. Ayrıca kısa tatillerin bir gün uzun tatillerin ise iki gün öncesi ve sonrası test setine ekledikten sonra genişletilen yeni modelde %12.63 hata oranı ile hız tahmini gerçekleştirilebilmektedir. Bu model kullanılarak temel model olarak baz alıp 1 hafta öncesini eğitim seti olarak kullanan Tarihi Ortalama (HA) modelinin verdiği %15.51 MAPE oranında 2.88'lik bir düşüş sağlanmıştır.

Anahtar Kelimeler: Trafik hız tahmini, uzun süreli, tatil, XGBoost, denetimli, zaman serisi, DFT

ABSTRACT

TRAFFIC FORECASTING FOR LONG AND SHORT HOLIDAYS: SPECIAL APPROACHES

Fatih ECEVİT

Yusuf Mert ÇELİKARSLAN

Department of Computer Engineering

Computer Project

Advisor: Assoc. Prof. Dr. Mehmet Amaç GÜVENSAN

Traffic analysis and vehicle speed estimation have a significant impact on the daily lives of individuals today. Since the beginning of the 21st century, approximately 33% increase in the world population and approximately 43% increase in the number of vehicles shows that the rate of increase in the number of both people and vehicles is high, and this increase in the number of people and vehicles brings traffic density. In order for the analysis and prediction processes to proceed and result successfully, the speed data of the vehicles in traffic must be complete and accurate. In this project, a model that forecasts traffic for long and short holidays was developed after the visualization, analysis, and interpretation steps of the numerical data using the data kept by the Istanbul Metropolitan Municipality. Fast Fourier Transform (FFT), which is a lower computational cost version of the Discrete Fourier Transform (DFT) algorithm, is used as a preprocessing method in the model. The preprocessed data were analyzed in the analysis step to determine the correct cut-off frequency and training sets were created for the days to be predicted. The training sets were first tested in the Random Forest model due to its tolerance to parameter variation. The feature importance was calculated and graphed with the Mean Impurity Decrease (MDI) and Mean Accuracy Decrease (MAD) algorithms, and the results were interpreted and feature selection was made. After this selection, other GradientBoosting, CatBoost, LGBM, XGBRegressor, and DNN models were also tested and a new model called FFT-Ensemble was created by selecting the models with better results. As a result of the new model, an average MAPE rate of 11.22% is obtained in the prediction

including only short and long vacation days, which is the better result in the literature than the SVR-HA[1] model. The error rate of 11.60% was exceeded and a decrease of 0.38 was achieved. In addition, the new model, which was expanded after adding one day before and after short holidays and two days before and after long holidays to the test set, can predict speed with an error rate of 12.63%. Using this model, a decrease of 2.88 was achieved in the MAPE rate of 15.51% given by the Historical Average (HA) model, which is based on the base model and uses 1 week before as the training set.

Keywords: Traffic speed prediction, long term, holiday, XGBoost, supervised, time-series, DFT

1 Giriş

1.1 Trafik Tahmin Probleminin Tarihçesi

Varolan makaleler tarandığında, 1960'lardan beri trafik tahmini üzerine birçok model geliştirilmiştir. İlk olarak istatistiksel modellerden olan ARIMA ve onun geliştirilmiş versiyonları olan SARIMA ve SARIMAX modelleri, trafik tahmini problemlerinde kullanılmıştır. Daha sonraları, makine öğrenmesi modellerinin regresyon verileri üzerindeki başarısı, bu modellerin zaman serileri içinde kullanılmasına neden olmuştur. Makine öğrenme modelleri, istatistiksel modellere oranla veride oluşan anlık değişimleri yakalamada daha başarılıdır; ancak mevsimsel değişimleri tahmin etmede daha düşük bir performans sergilemektedir.

Günümüzde ise derin öğrenme modellerinin gelişmesiyle zaman serisi problemlerinde çeşitli uygulamalar denenmiştir. Nöral ağların, makine öğrenme modellerine göre avantajı uzun süreli tahminlerde geçmiş verileri saklama becerisidir.

1.2 Trafik Tahmin Probleminde Kullanılan Veri Türleri

Her zaman serisinde olduğu gibi, trafik verisi de univariate ve multivariate olarak iki ana kategoriye ayrılmaktadır. Trafik verisinin hava durumu, sıcaklık, yol durumu gibi farklı özelliklerle desteklenmesi, elimizdeki modelin doğruluk oranını artırlabilir. Trafik verisi kendi içinde de farklılaşmaktadır. Literatürdeki tahmin modelleri genellikle trafik akışı, trafik yoğunluğu, yoldaki araç sayısı ve trafik hızını kullanmaktadır.

Bu projede, İstanbul Büyükşehir Belediyesi'nde 5 dakika aralıklarla örneklenen 2018 ve 2019 yıllarına ait trafik hızı verisi kullanılmaktadır.

1.3 Trafik Tahmin Modelleri ve Hata Metrikleri

Yapılan literatür taramasında, DFT-SVR modeli [2], XGBoost modeli [3] ve derin öğrenme modelleri arasında LSTM, CNN, ANN gibi modellerin uzun vadeli trafik tahmininde daha başarılı olduğu gözlemlenmektedir [4]. SVR modeli ve lineer regresyon modelleri üzerinde yapılan çeşitli testler sonucunda beklenen başarı elde edilememiştir. Kısa vadeli tahminlerde kabul edilebilir sonuçlar elde edilmiş olmasına rağmen, uzun vadeli tahminlerde ve tatil günlerinin değişken karakteristiği göz önüne alındığında, SVR modelinin bu araştırma konusu için yeterli olmadığı kanısına varılmıştır. Bu nedenle, araştırma XGBoost ve derin öğrenme algoritmaları üzerinden ilerlemeye odaklanmıştır.

Hata ölçme metriği olarak MAPE, modellerin sonuçlanma hatalarını hesaplamak için kullanılmıştır. Ayrıca MAE, yapılan tahminlerdeki en kötü ve en iyi tahminleri belirlemek ve buna göre parametreler belirlemek için tercih edilmiştir.

1.4 Karşılaşılan Sorunlar

Grafikler incelendiğinde, en sorunlu ve yoğun günlerin uzun süreli tatil günleri olduğu görülmektedir. Bir ve iki günlük resmi tatillerin, trafik verisinin olağan akışına pek bir etki etmediği gözlemlenirken, hafta sonuyla birleşen resmi tatiller veya kurban ve ramazan bayramı gibi uzun süreli tatillerin, birkaç günlük trafik yoğunluğuna sebep olduğu gözlemlenmektedir. Bu gözlem ışığında, iki önemli konu göze çarpmaktadır: yılbaşı gibi anlık yoğunluktan kaynaklanan çok ani hız düşüsleri veya uzun tatillerde meydana gelen kademeli düşüşler.

2 Ön İnceleme

Zaman serisi tahmin problemleri doğası gereği diğer makine öğrenme problemlerinden daha zorludur ve doğru sonuçlar elde edilebilmesi için eldeki verinin analizin doğru yapılması, veriye uygun bir model seçilmesi ve verinin kullanılmadan önce işlemenden geçilmesi gerekmektedir.

2.1 Zaman Serileri

Trafik hız zaman serisi, finans, tüketim ve diğer zaman serilerinden farklıdır. Bu farklardan ilki, trafik zaman serisinin belirgin bir trend eğrisi izlememesidir. İstanbul özelinde düşünüldüğünde, TÜİK'in 2022 Mart ayında açıkladığı verilere göre, İstanbul trafiğine günlük 585 araç, toplamda ise 213,486 yeni araç katılmıştır [5]. Ancak, bu yeni araçların trafik hız verisine 1 yıllık bir değişimde büyük ölçüde etkisi olmamakta ve grafiklerde yıllık olarak trafik hız verisinde dikkate değer bir artış veya azalış gözlemlenmemektedir. Zaman serilerindeki durağanlığın bir ölçüsü olan ADF testi, bu gözlemi desteklemektedir. Test, eşik p değeri olan 0.05'e uygulandığında, 0.0 sonucunu vermektedir, bu da zaman serisinin durağan olduğunu gösterir.

İkinci bir fark da dış etkenlerin, diğer veri türlerine kıyasla trafik verisinde belirgin bir biçimde etkilerinin görülmemesidir. Hava durumu, sıcaklık, tatiller, yol çalışmaları, insanların o anki tutumları gibi birçok etken, trafik verisini doğrudan etkileyebilir. Bu nedenle, projenin ilerleyen aşamalarında bu dış etkenlerin ekstra değişkenler olarak veriye eklenmesi düşünülebilir. Literatürde, hava durumu üzerinden trafik tahmini yapılan çalışmalara dair örnekler bulunmaktadır [6].

2.2 Probleme Farklı Yaklaşımlar

Literatür incelendiğinde, uzun süreli tatil günleri konusunda yapılan çalışma sayısının, makine öğrenmesi ve derin öğrenme konularının geniş yelpazesi düşünüldüğünde az olduğu görülmüştür. Bu nedenle, DFT-SVR modeli üzerine yapılan çalışmada, MAPE değerleri incelendiğinde, ARIMA ve diğer istatistiksel modellerden daha tutarlı sonuçlar verdiği gözlemlenmektedir [2]. Ayrıca, SVR modelinin basit doğası, makalenin ortaya çıkardığı sonuçları daha kolay yorumlanabilir kılmaktadır. Model, Çin'in Jiangsu şehrindeki National günü ve Tomb-sweeping gününde test edilmiştir. İstanbul'un nüfusu ve bu tatil günlerinin Türkiye'deki resmi tatillerin uzunluklarına benzerliği düşünüldüğünde, çalışmadaki model seçiminde etkisi olmuştur.

Diğer bir makalede [4], LSTM kullanılarak bir günlük araç yoğunluk tahmini yapılmış, geçmiş günlerin tahmine olan etkisi grafiklendirilmiştir.

2.3 Veri

İstanbul Büyükşehir Belediyesi (İBB), trafik tahmini için gerekli verileri sağlamak amacıyla CSV formatında bilgiler sunmaktadır. Bu veriler, 2018 ve 2019 yıllarına ait en az 2000 farklı lokasyondan gidiş ve dönüş yollarındaki araçların ortalama hızlarını içermektedir. Ortalama hız verileri her 5 dakikada bir alınmaktadır.

Trafik akış verileri, her lokasyon için ayrı bir CSV dosyası olarak Google Drive'da saklanmaktadır.

Ayrıca, her segmentin konum bilgisi, enlem ve boylam verileriyle birlikte bir JSON dosyasında tutulmaktadır. Bu bilgiler, belirli bir segmentin hangi konumu temsil ettiğini göstermektedir. Bu kapsamlı veri seti, trafik tahmin modelleri oluşturmak ve şehirdeki trafik durumunu analiz etmek için kullanılabilir.

3

Fizibilite

3.1 Teknik Fizibilite

Projenin amacına uygun olarak gereksinimlerin belirlenmesi ve planlanması yapılmıştır. Bu aşamada, veri çeşitliliği ve miktarı detaylı bir şekilde incelenerek, bellek gereksinimleri ve işlem süreleri gibi teknik detaylar belirlenmiştir. Bu adımlar, projenin başarılı bir şekilde yürütülmesi ve hedeflenen sonuçların elde edilmesi için önemlidir.

Gereksinim belirleme aşamasında, proje kapsamında ele alınacak konular, kullanılacak yöntemler, veri kaynakları, hedeflenen çıktılar ve sonuçlar net bir şekilde tanımlanmıştır. Planlama sürecinde ise bu gereksinimlere yönelik ayrıntılı bir proje planı oluşturulmuş ve bu planın başarıyla uygulanması için gerekli kaynaklar ve zaman çizelgeleri belirlenmiştir.

Ayrıca, veri çeşitliliği ve miktarı değerlendirilerek, projenin bellek gereksinimleri ve işlem süreleri önceden belirlenmiştir. Bu adım, proje sırasında karşılaşılabilen teknik zorluklara hazırlıklı olunmasını sağlamaktadır.

Bu önemli aşamaların tamamlanmasıyla, proje daha sağlam bir temel üzerine oturtulmuş ve başarıyla yürütülebilir hale gelmiştir

3.1.1 Yazılım Fizibiletisi

Projenin belirlenen aşamaları için uygun teknolojiler ve programlar seçilmiştir. Bu kapsamında, Python programlama dili tercih edilmiş ve özellikle veri analizi, makine öğrenimi modeli geliştirme ve test etme süreçlerinde yaygın olarak kullanılan kütüphaneler kullanılmıştır.

Aşağıda belirtilen teknolojiler ve programlar kullanılmıştır:

Python Programlama Dili:

Veri önişleme, analiz, ve model geliştirme süreçlerinde Python programlama dili kullanılmıştır. Python, veri bilimi ve makine öğrenimi alanlarında yaygın olarak tercih edilen bir dil olmasına bilinir.

Veri Önişleme ve Analizi:

Pandas: Veri manipülasyonu ve analizi için kullanılmıştır. NumPy: Bilimsel hesaplamalar ve veri üzerinde matematiksel işlemler için kullanılmıştır. Matplotlib ve Seaborn: Veri görselleştirmesi ve grafik oluşturmak için kullanılmıştır.

Makine Öğrenimi Modelleri:

Scikit-learn: Genel amaçlı bir makine öğrenimi kütüphanesidir. Birçok öğrenme algoritması içermektedir. XGBoost, LightGBM, CatBoost: Gradient Boosting algoritmaları, özellikle regresyon ve sınıflandırma problemlerinde kullanılmıştır. TensorFlow ve Keras: Derin öğrenme modelleri için kullanılmıştır.

Geliştirme Ortamı:

Anaconda ve Spyder: Python kodlarının geliştirilmesi ve çalıştırılması için kullanılmıştır. İşletim sistemi olarak Windows üzerinde proje hazırlanmış ve test edilmiştir. Python(3.11) kullanılarak, Anaconda ile spyder üzerinden algoritma çalışmaları yapılmıştır. Bu çerçevede, proje sürecinde kullanılan sistem, performans açısından yeterli bulunmuştur.

3.1.2 Donanım Fizibilitesi

Veri boyutu ve işleme hızı göz önüne alındığında, projenin gereksinimlerini karşılamak için donanımsal olarak optimize edilmiş bir bilgisayar kullanılması önemlidir. Mevcut donanımlarınızın bazı kısıtlamaları olduğu görülmüyor; ancak Google Colab gibi bulut tabanlı hizmetlerden yararlanarak bu kısıtlamaları aşmak mümkündür. İşte bu süreçte donanım ve hizmet seçimi ile ilgili öneriler:

Bilgisayar Donanımı:

RAM Kapasitesi: Veri setinizin büyüklüğü göz önüne alındığında, 16 GB RAM yetersiz olabilir. Özellikle büyük veri setlerini işlerken daha fazla belleğe ihtiyaç duyabilirsiniz. 32 GB veya daha fazla RAM kapasitesine sahip bir bilgisayar düşünülebilir. İşlemci: Yüksek frekansta ve çok çekirdekli bir işlemci, veri işleme süreçlerini hızlandırmaya yardımcı olabilir. 2.60 GHz işlemci hızı, daha hızlı bir işlemci ile değiştirilebilir.

Depolama:

SSD Kapasitesi: 500 GB SSD, veri setiniz büyükçe yetersiz kalabilir. Veri setinizin büyüklüğüne göre daha büyük bir SSD düşünülebilir veya harici depolama çözümleri kullanılabilir.

Ekran Kartı:

GPU: Veri işleme süreçlerinde GPU hızlandırma kullanılabileceği için daha güçlü bir GPU düşünülebilir. Özellikle büyük veri setlerinde makine öğrenimi modellerinin eğitimi GPU tarafından hızlandırılabilir.

Bulut Tabanlı Hizmetler:

Google Colab: Veri analizi, işleme ve model geliştirme süreçlerinde, Google Colab gibi bulut tabanlı hizmetlerden yararlanmak, donanım kısıtlamalarını aşmanın etkili bir yoludur. GPU ve TPU gibi kaynaklara ücretsiz olarak erişim sağlayabilir ve büyük veri setlerini işleyebilirsiniz.

Düzenli Veri Temizliği ve Saklama Stratejisi:

Büyük veri setlerini işlerken düzenli olarak gereksiz verileri temizlemek ve sadece gerekli olanları saklamak, hem depolama alanından tasarruf sağlar hem de işlemleri hızlandırır. Donanım ve hizmet seçiminde, bütçe, proje öncelikleri ve kullanım alışkanlıkları gibi faktörleri dikkate almak önemlidir.

3.2 Ekonomik Fizibilite

Donanım imkanları örnek veri setleri için şuanlık yeterli olduğundan yeni bir donanıma veya bulut geliştirme ortamına ücret ödenmemiştir.

3.3 Legal Fizibilite

Veri toplama ve işleme faaliyetleri için yasal gerekliliklerin belirlenmesi ve buna uygun hareket edilmesi (örneğin, veri gizliliği, kişisel verilerin korunması vb.) trafik yönetiminde kullanılabilecek verilerin uygunluk ve güvenilirlik açısından incelenmesi ve proje sürecinde elde edilen sonuçların kullanımı veya paylaşımı için yasal gerekliliklerin belirlenmesi adımlarının takip edilmesi sonucunda projenizin herhangi bir yasal sorun teşkil etmediği tespit edilmiştir.

3.4 İşgücü ve Zaman Fizibilitesi

Projenin gereksinimlerine uygun olarak bir proje yönetim planı hazırlanmış ve süreç içinde aşağıdaki adımlardan geçilmiştir:

Proje Yönetim Planı Hazırlama:

Proje gereksinimleri belirlenmiş ve bu gereksinimlere uygun bir proje yönetim planı oluşturulmuştur. Plan içinde, ekip üyelerinin görevleri ve sorumlulukları net bir şekilde tanımlanmıştır.

Ekip Üyeleri ve Görev Tanımları:

Ekip üyelerinin yetkinlikleri, uzmanlıkları ve ilgi alanları dikkate alınarak, her bir ekip üyesine uygun görevler ve sorumluluklar atanmıştır. Ekip içindeki iletişim ve işbirliği süreçleri planlanmıştır.

Proje Süresinin Belirlenmesi:

Proje süresi, yapılacak işlerin karmaşıklığına ve projenin genel hedeflerine göre belirlenmiştir. Milestone'lar ve takvimlerle proje süreci ayrıntılı bir şekilde planlanmıştır.

İş Planlaması ve Ayrıntılı Çalışma:

Proje süresince yapılacak olan işler detaylı bir şekilde planlanmış ve görevlerin zaman çizelgesi oluşturulmuştur. Gereksinimlerin değişimeceği durumlar göz önüne alınarak, esneklik sağlanmış ve planlamalarda uyum sağlanabilirlik öngörülmüştür.

Literatür Taraması ve Ön Çalışma:

Proje öncesinde, literatür taraması ve ön çalışma süreçleri gerçekleştirilmiştir. Bu aşama, mevcut bilgi birikimini artırmak ve projenin temelini sağlam bir şekilde oluşturmak amacıyla gerçekleştirilmiştir.

Model Belirleme ve Çalışmaların İlerlemesi:

Gerekli modeller belirlenmiş ve bu alandaki çalışmalar ilerletilmiştir. Bu aşama, projenin ana hedeflerine ulaşmak için uygun metodolojilerin seçilmesi ve uygulanmasını içermiştir. Bu adımlar, proje yönetimini sağlam temellere oturtarak, proje sürecinin etkin ve başarılı bir şekilde yürütülmesine olanak tanımiştı

4

Sistem Analizi

4.1 Veri Seti Analizi

Trafik tahmini için gerekli olan veriler İstanbul Büyükşehir Belediyesi (İBB) tarafından temin edilmiştir. Temin edilen veriler CSV formatında olup Google Drive'da saklanmaktadır. CSV dosyaları 2018 ve 2019 yılına ait en az 2000 farklı lokasyondan gidiş ve dönüş yolları için geçen araçların ortalama hızlarını tutmaktadır. Ortalama hız verilerini her 5 dakikada bir almaktadır.

Trafik akış verileri CSV formatında, her lokasyon için farklı CSV dosyası ve numarası olacak şekilde zaman bilgisi ve ortalama hız bilgisi ile birlikte tutulmaktadır. Hangi segmentin hangi konumu belirttiği bir JSON dosyası içerisinde enlem ve boylam bilgisi olarak tutulmaktadır.

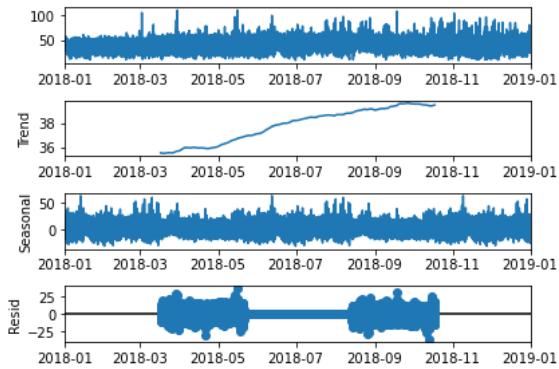
Veri incelendiğinde, bazı aralıklarda verilerin -1 değeri aldığı görülmüştür. Bu verilerin eğitim ve test adımlarında sorun teşkil etmesi nedeniyle, bir fonksiyon çağrısı ile ortalama değerler kullanılarak -1 değerler doldurulmuştur. Fonksiyon içeriği daha sonra "Kayıp değerleri doldurma" başlığı altında detaylı açıklanacaktır.

4.1.1 Mevsimsel Ayışma

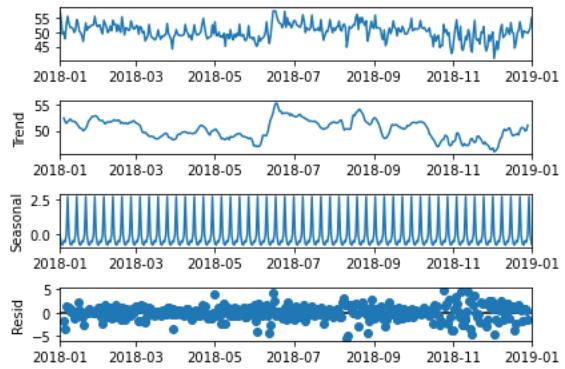
Veriyi mevsimsel olarak ayırttığımızda, trend, seasonal ve residual eğrilerini görmemiz mümkün olmuştur. Bu sayede veri konusunda daha iyi bir anlayış geliştirilmiştir. Şekil 4.1'de görüldüğü gibi belirgin bir trend eğrisi yoktur, fakat seasonal eğri belirgin biçimde kendini göstermektedir. Yapılan ADF testi de gözlemlerimizin sonucunu doğrulamıştır.

Veri günlük olarak yeniden örneklenliğinde, veri gürültüsü azaldığı için daha belirgin mevsimsel eğriler gözlemlenmiştir. Şekil 4.2'de eğrinin mevsimsel değişimi görülebilmektedir.

Sonuç olarak, bu iki grafik bize uzun tatil günlerinde, eğer mevsimsel tekrar



Şekil 4.1 5 dk. aralıklarla



Şekil 4.2 Günlük

Trend Verisinin, Mevsimsel Verinin ve Kalan Veririn Ayışması

zamanlarındaki kısım eğitim veri seti olarak alınırsa, tahmin sonucunun daha doğru çıkacağı öngörüsüne ulaştırmıştır.

4.1.2 Geçmiş Verilerle Korelasyon

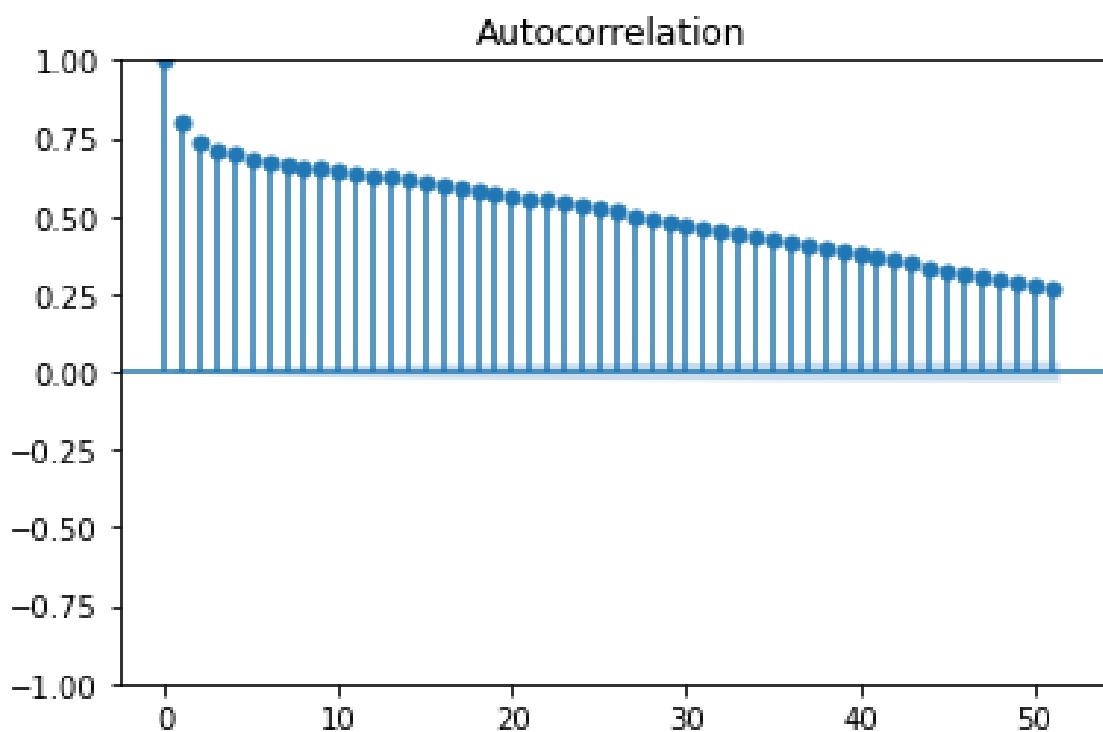
Bir zaman serisi probleminde geçmiş verilerin modelin yaptığı tahmine etkisi en önemli konulardan birisidir. Elimizdeki veri 5 dakika aralıklarla olduğu için 5 dakika önceki bir artış ya da düşüş eğilimini korumaktadır. Bu nedenle korelasyon ve kısmi korelasyon eğrilerini incelememiz gerekmektedir.

Şekil 4.3'de korelasyon eğrisine baktığımızda geçmiş örneklerle doğru azalarak devam eden bir korelasyon görülmektedir. Geçmişteki hangi örneklerin sonuca daha çok etki ettiğini anlamak için Şekil 4.4'deki kısmi korelasyon eğrisini incelemek gerekmektedir. Bu grafikte görüldüğü üzere 0 ve 10 adım önceki verilerin korelasyonu göze çarpmaktadır. Bu nedenle tatil verileri incelenirken tatil gününden 1-10 örnek öncesi ve sonrası göz önüne alınmıştır.

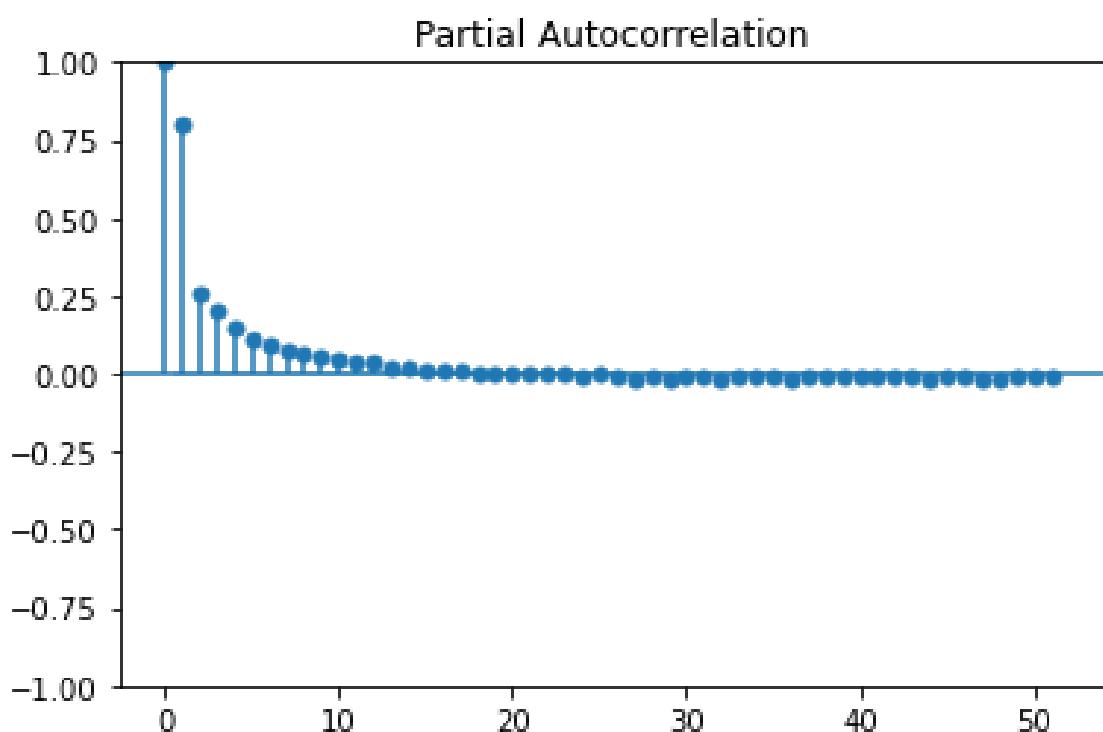
4.1.3 Özellik Seçimi ve Farklı Bayramlardaki Kullanımları

XGBoost kullanılarak 1 hafta öncesinden alınan örneklerin 1 hafta sonrasına tahmin etmekte olan etkisini gözlemlemek amacıyla hız verisi, hız verisi arasındaki saatlik fark, hız verisi arasındaki yüzdesel fark ve hız verisinin MA metoduyla smoothing işlemi yapılmasıyla oluşan değişkenler model eğitiminde kullanılmıştır. Model Bağdat Caddesi segmenti 2018 verisi kullanılarak eğitilmiştir. Bu model tatil tahmini yapmak için değil, veri üzerinde geçmiş verilerin ve zaman değişkenlerinin etkisini görmek için geliştirilmiştir.

Şekil 4.5'te sonuçlar Bağdat Caddesi hariç beş farklı segment genelinde

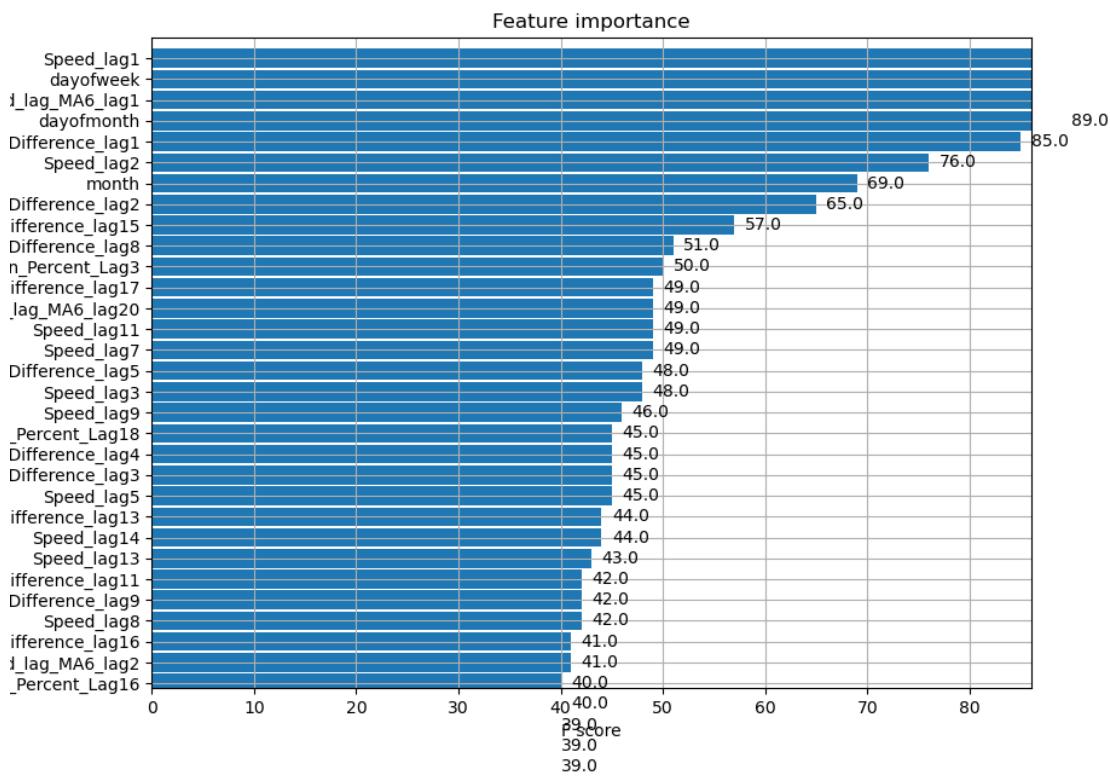


Şekil 4.3 Korelasyon



Şekil 4.4 Kısmi Korelasyon

incelediğinde, 1 hafta önceki verinin, haftanın günlerinin ve ayın günlerinin tahmin sonucunda en fazla etkiye sahip olduğu gözlemlenmiştir.



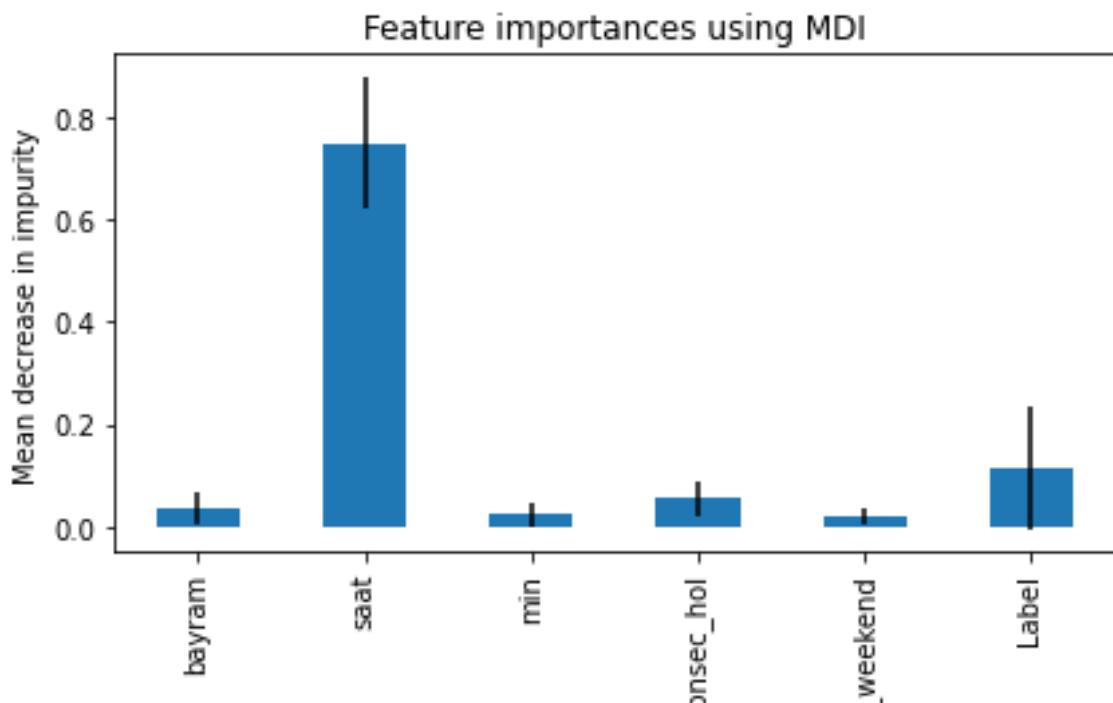
Şekil 4.5 Değişken Önem Yatay Bar Grafiği

XGBoost ile yapılan analizde, 1 hafta öncesinin 1 hafta sonrasında olan etkiyi değerlendirmek için özelliklerin önemini belirlemek amacıyla önce eğitim setini kullanarak MDI (Mean Decrease in Impurity) modelini incelemiştir (4.6 Şekil). Ancak, MDI modelinin eğitim setini baz aldığı için yanıldıcı sonuçlar verebileceği fark edilmiştir. Bu nedenle, aynı zamanda 4.7 Şekil'de gösterildiği gibi permütasyon modeli de kullanılarak özellik önemleri çıkarılmıştır.

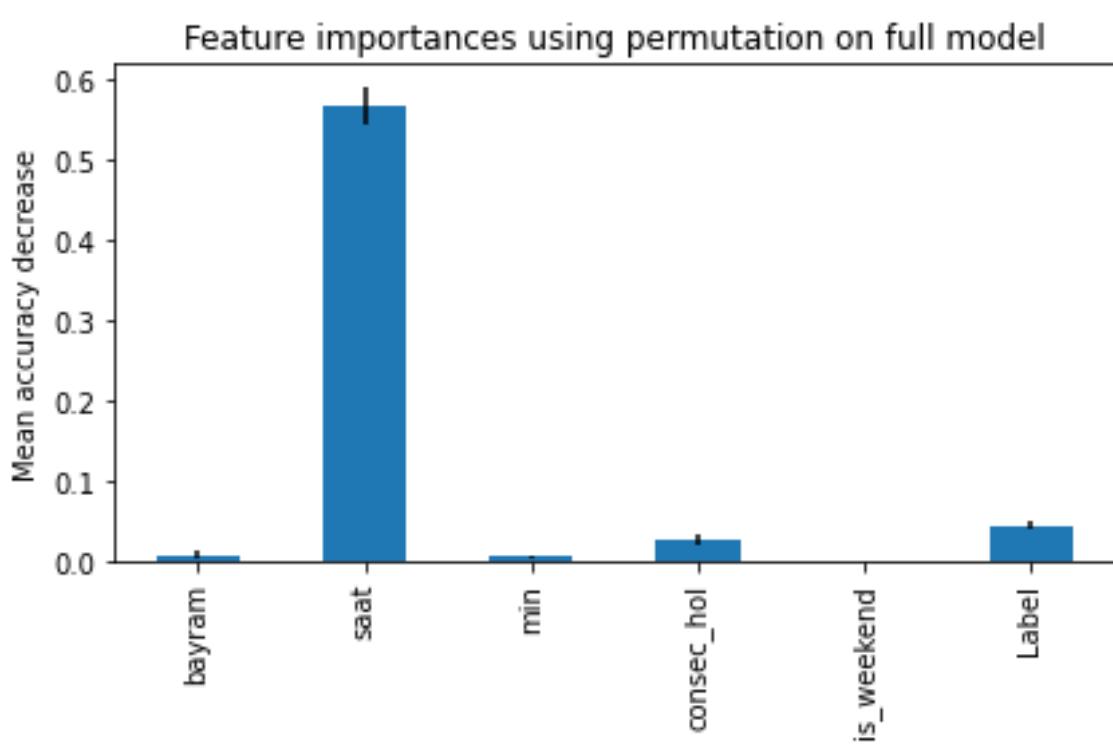
Şekil 4.6 ve Şekil 4.7'de, bu iki modelin farklı sonuçlar verdiği görülmektedir. Eğitim setinin doğası gereği, bazı parametrelerin sonucu olumsuz etkilediğini ve bazı parametrelerin ise hiçbir işe yaramadığı görülmektedir. Bu çift yönlü analiz, modelin güvenilirliğini ve genel performansını daha iyi değerlendirmemize olanak tanımaktadır.

4.1.4 Test Seti Olarak Kullanılan Segmentler ve Özellikleri

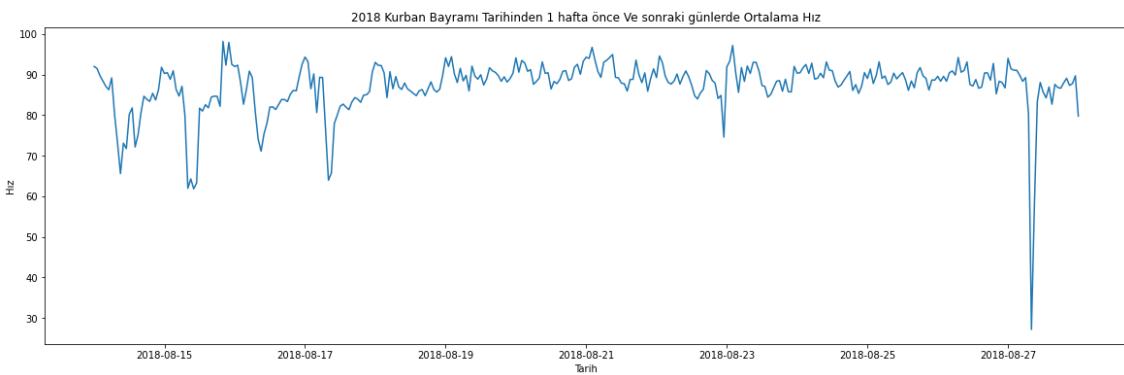
İstanbul genelinde 2200'e yakın segment ve bu segmentlerin gidiş ve geliş şeritleri olduğu için testleri bu 2200 segmentte yapmak yerine seçilen 106 segmentte yapılması tercih edilmiştir. Bu tamamen zaman ve işlem gücü kısıtı nedeniyle yapılmasına rağmen bu 106 segment İstanbuldaki trafiğin genel karakteristiğini yansıtmaktadır.



Şekil 4.6 MDI Özellik Önemi



Şekil 4.7 Permüstasyon Özellik Önemi



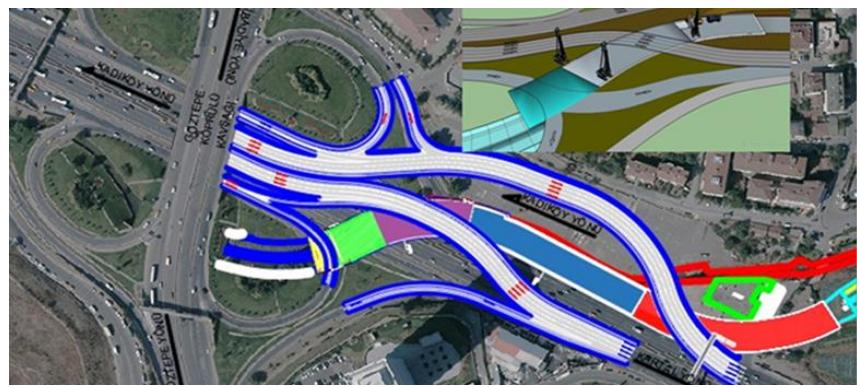
Şekil 4.8 E-5 Ana Arter

Ana Arterler

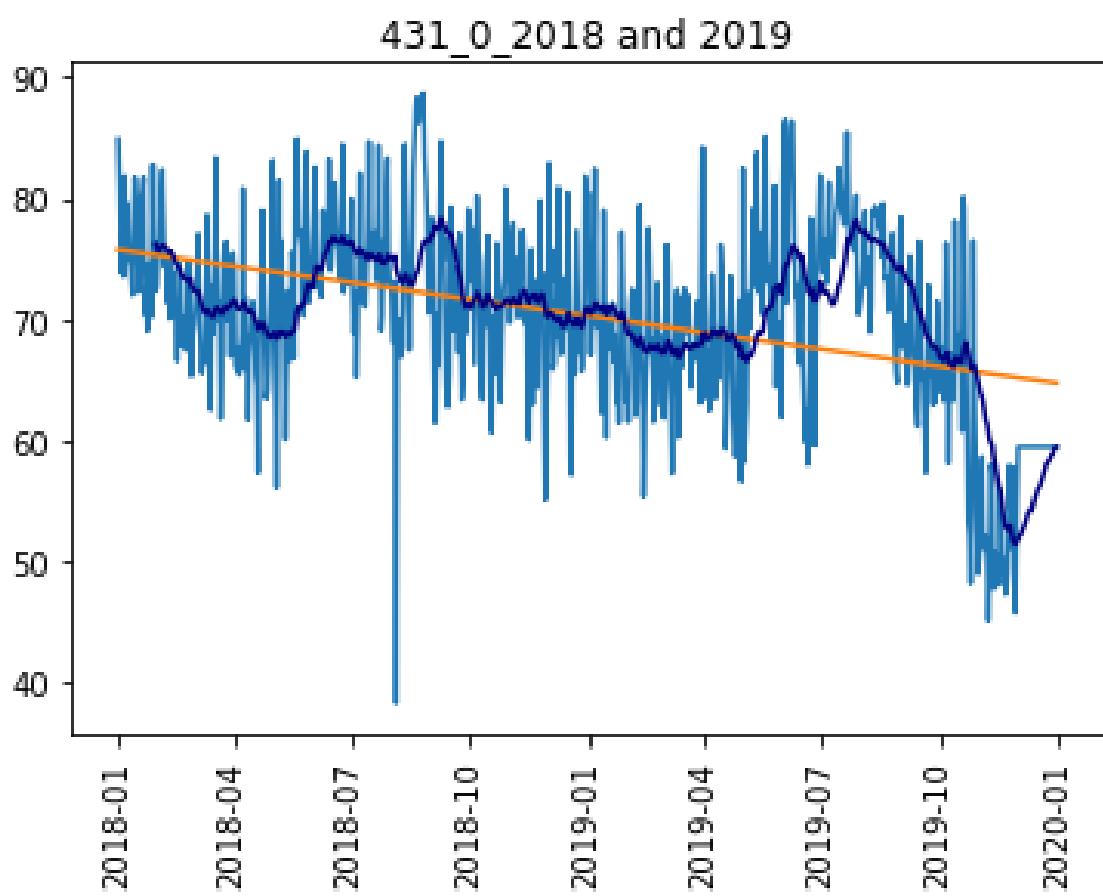
İstanbul'un genel yol yapısına bakıldığından, Avrasya Yolu, E5 gibi çok şeritli ana yollar bulunduğu görülmektedir. Bu yolların 4.9 hız grafiklerine bakıldığından, yerel kaynaklı olaylardan kaza, etkinlik vb. daha az etkilendiği görülmektedir. Çünkü bu yollar anlık değişimleri, şerit çokluğundan kaynaklı daha rahat kaldırılabilmektedir. Aynı zamanda tatil günlerinde de bu yolların şehir içindeki yollara göre daha az etkilendiği görülebilmektedir. Şekil 4.9 ve Şekil 4.11 farka bakıldığından, ana arterlerin o yolun maksimum hız limitine ne kadar yakın ve düz bir grafik izlediğini, diğer dar yolun ise ne kadar değişken bir grafik izlediğini görebilmektedir. Fakat ana arterlerin en çok etkilendiği dış etkenin yol çalışmaları olduğu görülmektedir. Kadıköy E-5 no'lu segmente bakıldığından bir ana yol bulunmakta; bu segmentte yapılan tahminler, ortalama hatanın çok üstünde bir hata oranı verdiği için 2018 ve 2019 hız verileri bu segmente özel olarak incelenmiştir. Grafiklere bakıldığından, Şekil 4.10'te 2019'da hızın ortalama olarak düşürüldüğü görülmektedir. Bu düşüş, Kurbağalıdere yenileme nedeniyle yapılan altyapı çalışmasının yola etki etmesi sonucu olduğu fark edilmiştir. Yol çalışması tam olarak 7 Ekim 2019 tarihinde başladığı için 29 Ekim için yapılacak olan tahminde 2018 verisinin kullanılması ve 7 Ekimden öncesinin kullanılmasını engellemektedir, çünkü trafik karakteristiği bu çalışmaya beraber tamamen değişmektedir. Problemi çözmek için yol bakım ve kaza verilerinin modele entegre edilebileceği düşünülmektedir.

Şehir İçi Yollar

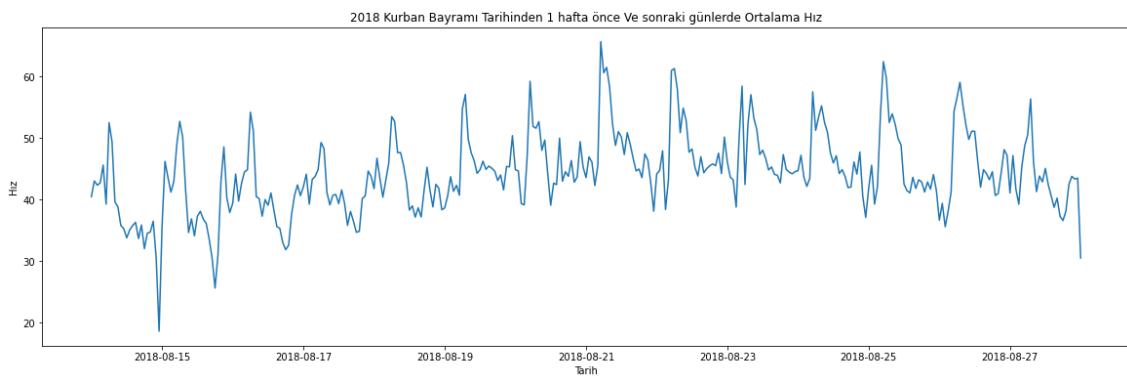
Lokal faktörlerin asıl etkili olduğu yollar ilçelerde mahalleleri bağlayan yollardır. Bu yolların tahliye yolları olmadığı için kazalar ve yerel etkinlikler anlık trafik sıkışıklıklarını yaratıbmektedir. Bu nedenle bu yolların arterlerden farklı şekilde incelenmiştir.



Şekil 4.9 Kurbagalidere Ana Arter [7]



Şekil 4.10 Kurbagalidere Yol Çalışması Sonucu Etkilenen 431 Segmenti



Şekil 4.11 Şehir İçi Yol

4.2 Uzun Tatiller

Resmi tatil günlerindeki değişimi incelemek için Ramazan ve Kurban Bayramı tatil günleri 2018 ve 2019 yıllarında ayrı ayrı görselleştirilmiş ve karşılaştırmalar yapılmıştır. Konum seçimlerinde İstanbul trafiğinin genel karakteristiğini yansitan 106 segment teste tabi tutulmuştur. Test sonuçları, İstanbul'un geneli için uygun bir sonuç vermektedir.

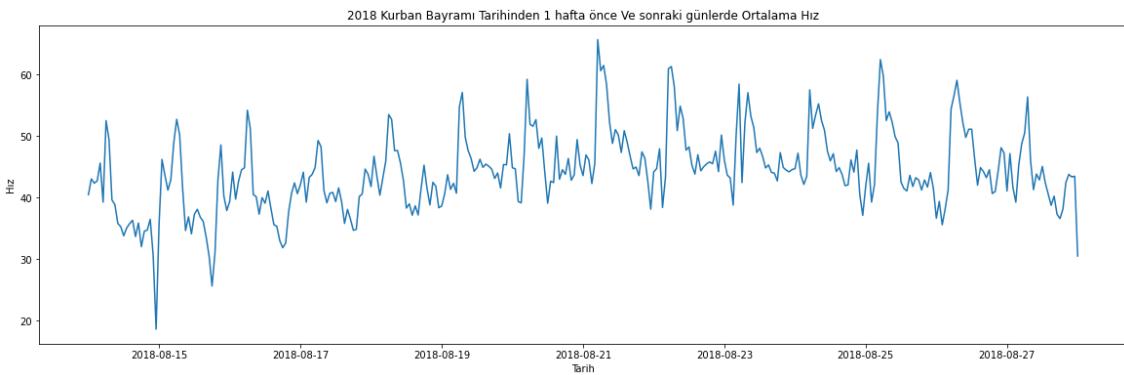
Gözlemlenen konumlar

- Bağdat Caddesi
- Londra Asfaltı(Büyükçekmece)
- Haliç
- Fatih Sultan Mehmet Köprüsü
- Dolmabahçe Sarayı
- Üsküdar-Ümraniye TEM
- Kuzey Marmara Otoyolu
- E5

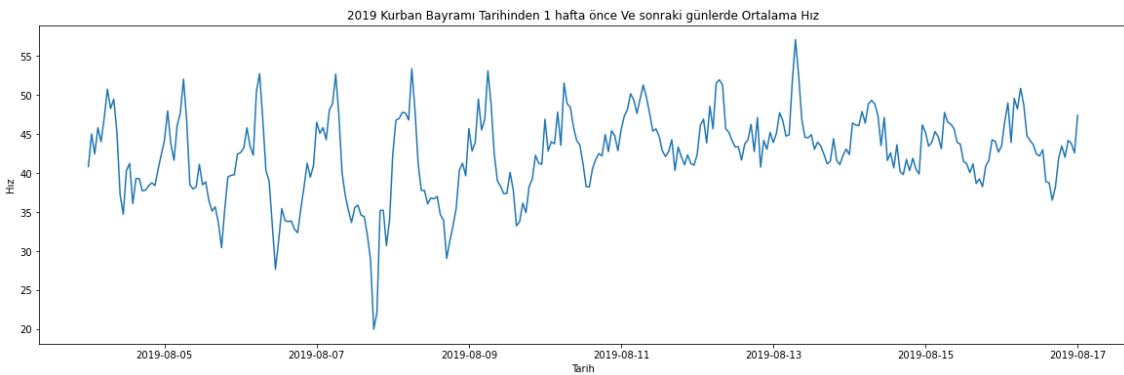
4.2.1 Kurban Bayramı

Kurban bayramlarında genellikle insanlar uzun süreli şehir dışına çıktıkları için özellikle hafta sonuyla birleştiğinde İstanbul trafiğinde bir yoğunluk yaşanmaktadır. Şekil 4.12'de görüldüğü gibi, 2018 yılında bayram hafta sonuyla birleştiğinden 15 Ağustos ayında trafik hızında bir düşüş gerçekleşmektedir. Sonrasında İstanbul'daki insan sayısının azalması nedeniyle trafik hızı normal değerlerin üstüne çıkmaktadır. Bir sonraki yılda ise arife günü 10 Ağustos'a denk gelmektedir, bu nedenle 7-9 Ağustos tarihleri arasında trafik yoğunlığında bir artış gözlemlenmektedir. 2018'de de olduğu gibi uzun süreli tatil, İstanbul trafik yoğunlığında azalmaya sebep olmaktadır.

Kurban bayramı için eğitim günleri olarak 18 - 26 Ağustos 2018 arasını ve 3, 4 Ağustos 2019 günlerini seçilmiştir. Cumartesi ve pazar günleri tatil günlerine benzer bir karakteristik gösterdiği için 2019 yılında test seti olarak kullanılan 10 - 14 Ağustos'a bir hafta önceki haftasonunu da eğitim setine dahil edilmiştir. Bu şekilde, bayram tatilinin ve hafta sonlarının etkilerini daha iyi anlamak ve modelin bu durumları doğru bir şekilde değerlendirmesini sağlamak amaçlanmıştır.



Şekil 4.12 Bağdat Caddesi 2018 Kurban Bayramı



Şekil 4.13 Bağdat Caddesi 2019 Kurban Bayramı

4.2.2 Ramazan Bayramı

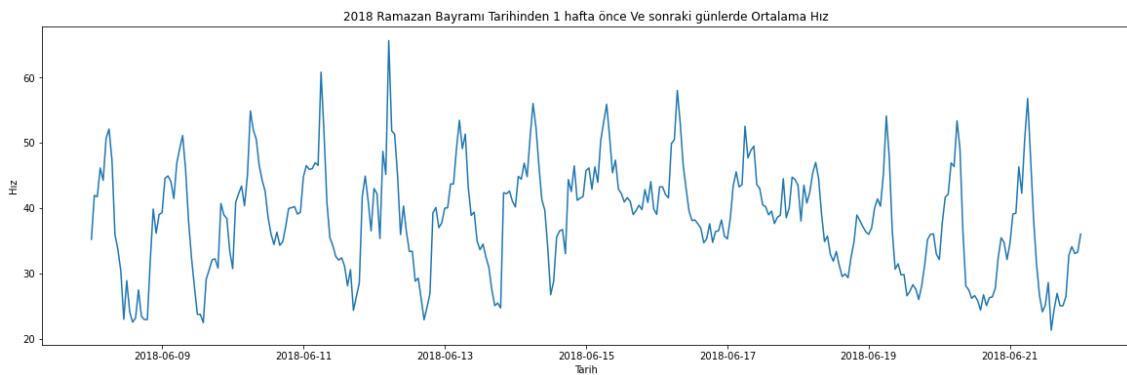
Ramazan Bayramı incelendiğinde, trafik yoğunlığında kayda değer bir artış gözlemlenmemiştir, ancak 2018'de Şekil 4.14'de görüldüğü gibi tatilin olduğu 15-17 Haziran tarihleri arasında trafik yoğunlığında azalma gözlemlenmiştir. Ek olarak, Şekil 4.15'da 2019 yılında tatil süresinin hafta sonuyla birleştiği için 1-9 Haziran tarihleri arasında trafik yoğunlığında bir miktar düşüş gözlemlenebilir.

Ramazan Bayramı genelde uzun tatiller olduğunda insanların ilk günlerde şehri terk etme eğilimi gösterirken, kurban bayramında insanların tatile gidişleri bayramın geneline yayıldığı gözlemlenmiştir. Bu nedenle, Kurban Bayramı grafiklerinde birden fazla düşüş ve yükseliş gözlemlenirken, Ramazan Bayramı grafiklerinde genelde bir düşüş ve yükseliş gözlemlenmiştir.

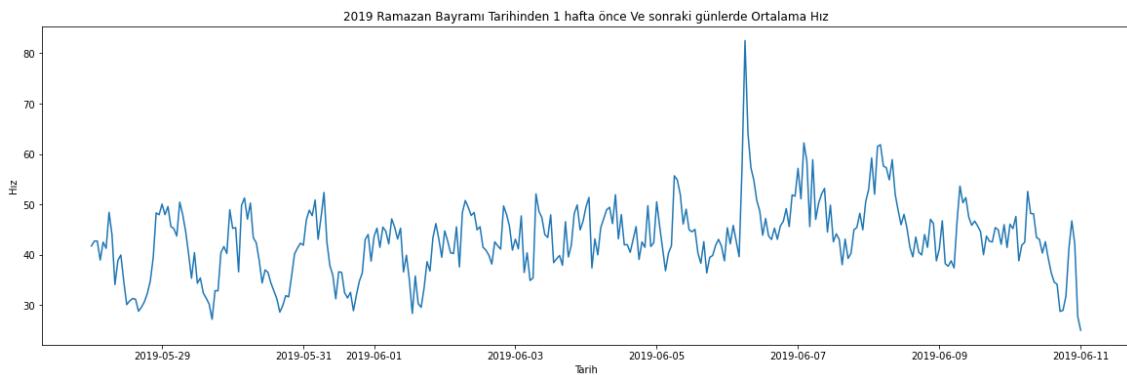
Ramazan Bayramı verilerine baktığımızda, 2018'de hafta sonuna denk geldiği için 15-17 Haziran arasında 3 günlük bir tatilken, 2019'da 1 - 9 Haziran arasında 9 günlük bir tatil oluyor. Eğitim setinde 2018 verisini kullanarak yapılan tahminlerde %18 gibi kötü bir sonuç alındığı için, kurban bayramından farklı bir yaklaşım izlenmesi gerektiği fark edilmiştir. Kurban Bayramı 2018 yılında 2019 yılındaki Ramazan gibi uzun bir dönemi kapsadığı için, kurban bayramını kullanarak tahmin etmeyi denedigimizde

daha kötü sonuçlar alındığını fark edilmiştir.

Bu sorunu çözmek için, bayram günlerinin korelasyonuna bakıldığından yüksek bir korelasyon olduğu için kurban bayramının bayram günlerini eğitim setine dahil edilmiştir. Ancak arife ve ağustos ayının diğer günleri, haziran ve ağustos aylarındaki karakteristik farkından dolayı eklenmemiştir. Sonuç olarak, eğitim setinde bir hafta öncesinin 2019 hafta sonu, Ramazan Bayramı ve Kurban Bayramı günleri yer almıştır. Bu eğitim setiyle %13'e kadar hata oranını indirilebilmiştir. Hata oranlarına daha detaylı olarak Performans Analizi bölümünde değinilecektir.



Şekil 4.14 Bağdat Caddesi 2018 Ramazan Bayramı



Şekil 4.15 Bağdat Caddesi 2019 Ramazan Bayramı

4.3 Kısa Tatiller

Günlük resmi tatillerin, uzun süreli tatillere nazaran daha kısa süreli olması nedeniyle trafik karakteristiği üzerinde benzer bir etki göstermediği gözlemlenmektedir. Burada, yol ve gün özelinde karakteristiklerin etkisinin daha belirgin olduğu görülebilir. Resmi tatiller, genelgeçer ve ülkelere özgü olmak üzere farklı gruplara ayrılabilir. Genelgeçer bir bakış açısıyla günlük tatillerin haftasonuna denk gelmesi ile haftaiçine denk gelmesi arasında belirgin bir fark gözlenmektedir. Haftasonu ve haftaiçi tatillerinin farklı bir şekilde incelenmesi gereklidir. Tablo 4.1'de bu gruplara denk gelen günler görülmektedir.

Tablo 4.1 Tatil Günlerinin Türlerine Özel Tahmin Setleri

Tatil (2019)	Tatil Türü (2019)	Tatil Türü (2018)	Tahmin Seti
23 Nisan Ulusal Egemenlik ve Çocuk Bayramı	Kısa Tatil - Hafta İçin-Tek Günlük	Kısa Tatil - Hafta İçin-Üç Günlük	1 Mayıs 2018 + 30 Ağustos 2018 + 28-29 Ekim 2018 + 21 Nisan 2019 (Bir önceki pazar)
1 Mayıs Emek ve Dayanışma Günü	Kısa Tatil - Hafta İçin-Tek Günlük	Kısa Tatil - Hafta İçin-Tek Günlük	30 Nisan-1 Mayıs 2018+ 18-19 Mayıs 2018+ 29-30 Ağustos 2018 + 28-29 Ekim 2018 + Son 1 hafta 2019
19 Mayıs Atatürk'ü Anma, Gençlik ve Spor Bayramı	Kısa Tatil - Hafta Sonu	Kısa Tatil - Hafta Sonu	12 Mayıs 2019 (Geçen hafta aynı gün)
15 Temmuz Demokrasi ve Millî Birlik Günü	Kısa Tatil - Farklı Karakteristik	Kısa Tatil - Farklı Karakteristik	2018 Milli(2) + 2019 15 Temmuza kadarki milli bayramlar + 8 Temmuz 2019 (1 Hafta önce aynı gün) + Bir önceki Cumartesi-Pazar
30 Ağustos Zafer Bayramı	Kısa Tatil - Hafta İçin-Üç Günlük	Kısa Tatil - Hafta İçin-Tek Günlük	30 Nisan-1 Mayıs 2018+ 18-19 Mayıs 2018+ 29-30 Ağustos 2018 + 28-29 Ekim 2018 + 23 Nisan 2019 + 1 Mayıs 2019 + Son 1 hafta 2019
30 Ağustos Zafer Bayramı	Kısa Tatil - Hafta İçin-Tek Günlük	Kısa Tatil - Hafta İçin-Üç Günlük	22 Ekim-4 Kasım 2018 + 23 Nisan 2019 + Son 1 hafta 2019
Ramazan Bayramı	Uzun Tatil - 9 Günlük	Uzun Tatil - 3 Günlük	2 Haziran 2018 + 1 Haziran 2019(Önceki cumartesi) + 14-15-16-17 Haziran 2018 (Ramazan 2018) + Kurban 2018
Kurban Bayramı	Uzun Tatil - 4 Günlük	Uzun Tatil - 9 Günlük	2019 bayramdan önceki haftasonu + 18 Ağustos-26 Ağustos 2018

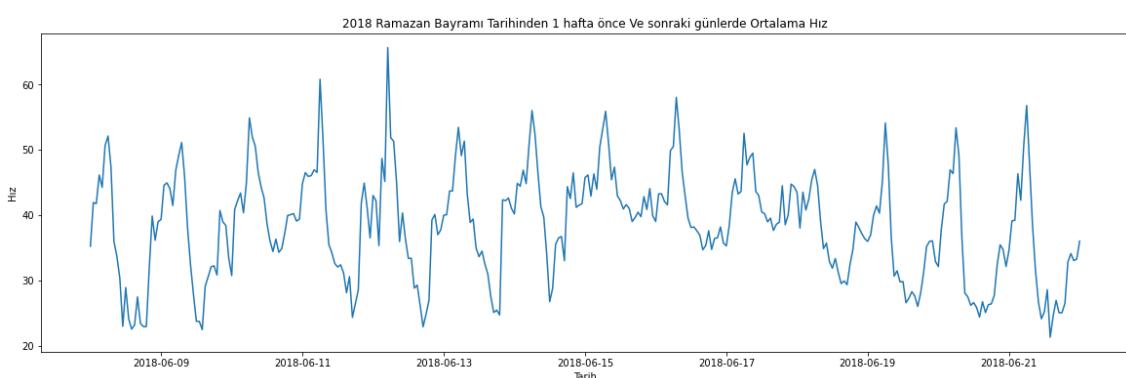
Ülkelere göre değişen durumlar ise insanların tatillere gösterdiği ilgi çerçevesinde gerçekleşmektedir. Örneğin, 29 Ekim günü her sene belirli bir yoğunluk gözlenmesine rağmen, Türkiye Cumhuriyeti'nin 100. yılında çok daha farklı bir trafik durumu yaşanmış olabilir. Ayrıca, resmi tatillerin denk geldiği mevsimler de bu durumu etkileyebilir. Bu nedenle, farklı karakteristik gösteren bu tatilleri ayrı bir grup olarak incelemek uygun görülmüştür.

4.3.1 Haftaiçine Denk Gelen Tatiller

2018 yılında 23 Nisan pazartesi, 1 Mayıs salı, 30 Ağustos perşembe, 29 Ekim pazartesi gününe denk gelirken, bir sonraki yıl tek günlük bu tatiller bir gün sonrasında denk gelmektedir. Eğer veri setinde bir önceki yılda cuma gününe denk gelen bir tatil olsaydı, tahminlere göre bu gün eğitim setinde kullanıldığında kötü bir sonuç elde edebilirdi çünkü haftaiçi ve haftasonuna denk gelen tatil günlerinde farklılık gözlenmektedir. Cuma ve pazartesine denk gelen günlük tatiller haftasonuyla birleştiği için, insanların davranışlarına daha fazla etki edebilir.

Örneğin, yukarıda bahsedildiği gibi 23 Nisan tatili pazartesi gününe denk gelmektedir. İnsanlar 3 günlük tatil nedeniyle İstanbul'a yakın şehirlere veya aile ziyaretlerini 3 günlük tatil nedeniyle tercih edebilmektedir. İncelenen sonuçlarda da bu görülmüştür. İki yıl arasındaki bu fark, 23 Nisan tahminlerinin diğer haftaiçine denk gelen tatillerden daha kötü olması sonucunu doğurmaktadır çünkü haftaiçine denk gelen tatillerde aynı yaklaşım izlenmiştir. Bu grup, 3 günlük ve tek günlük tatiller olarak incelense de hata bazında 3 günlük tatilleri bu gruptaki tek günlük tatiller seviyesine indirmeyi şu anda gerçekleştirmek mümkün değildir.

Şekil 4.16 bu durumun nedenini açıkça göstermektedir. Şekil 4.16 incelediğinde, 23 Nisan tatilinin diğer haftaiçi tatillerinden daha farklı bir karakteristik sergilediği görülmektedir. Bu durum, tahminlerin bu tatili diğerlerinden daha kötü bir şekilde anamasına neden olmaktadır. Bu konudaki daha detaylı çözümler ve iyileştirmeler, gelecekteki analizler ve model güncellemeleri ile mümkün olabilir.



Şekil 4.16 Bağdat Caddesi 2018 Ramazan Bayramı

Tek Günlük Tatiller

Veri setinde 2018 yılında yukarıda bahsedilen günlerden 1 Mayıs ve 30 Ağustos günleri bu kategoriye girmektedir. Bir sonraki yılda ise sadece 30 Ağustos Zafer Bayramı harici diğer bayramlar bu kategoridedir. İki yıl arasında 1 Mayıs'ın belirlediğimiz gruppala göre karakteristik değiştirmediği görülmektedir. Bu nedenle 1 Mayıs için kullanılan

eğitim setinde 4.1 tablosunda görüldüğü gibi tek günlük bayramlar ve bir önceki gün, 1 Mayıs 2019'dan önceki son bir hafta ve 1 Mayıs 2018 haftasının haftasonu kullanılmıştır. 30 Ağustos Zafer Bayramı'nda ise 1 Mayıs eğitim setindeki aynı mantık uygulanmakla beraber, 2019 senesinde 30 Ağustos'tan önce gelen tek günlük bayramlar da eğitim setine eklenmiştir. Aynı seneden önceki bayramları ekleme yaklaşımı genel olarak bütün bayramlar için yapılmıştır. Daha detaylı sonuçlar 4.1 tablosunda görülebilmektedir. 30 Ağustos tahmininde hata oranlarının ortalaması, 1 Mayıs'tan yüzde 1 oranında daha yüksektir. Bunun nedeni, yukarıda bahsedildiği gibi 30 Ağustos'un 2019 yılında karakteristik değiştirmesi ve 3 günlük tatil kategorisine girmesidir.

Üç Günlük Tatiller

29 Ekim Cumhuriyet Bayramı ve 23 Nisan Ulusal Egemenlik ve Çocuk Bayramı 2018 yılında bu kategoriye girmektedir, ancak 2019 yılında karakteristik değiştirerek tek günlük tatil kategorisine dahil olmaktadır. Ancak hata oranlarına bakıldığında, 23 Nisan'ın ortalama olarak yaklaşık yüzde 2 oranında daha fazla hata verdiği görülmektedir. Aslında, her ne kadar iki yılda da aynı karakteristiği izleyen tatiller olarak gözükseler de, 22 Nisan 2019 pazartesi günü tatil değilken, 28 Ekim pazartesi günü yarı� gün tatil edilmektedir. Bu nedenle, 29 Ekim'in 3 günlük tatil durumunu bir yandan devam ettirdiği söylenebilir.

4.3.2 Haftasonuna Denk Gelen Tatiller

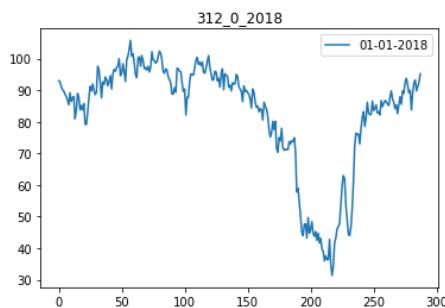
23 Nisan Gençlik ve Spor Bayramı bu kategoriye girmektedir. Veri setinde bulunan iki yılda da haftasonuna denk geldiği için tek günlük tatillere uygulanan yaklaşım burada %15 gibi kötü bir sonuç vermektedir. Önemli olan insanların haftasonuna denk gelen tatillerde nasıl bir davranışta bulunduklarıdır. Herhangi bir haftasonundan farkı olmayan bu günlerde haftaiçinde oluşan sabah 8 akşam 5 trafiği görülmemektedir. Ancak trafikte her ne kadar bir azalma olsada tek günlük tatillerde yaşanan belirli bölgelerdeki yoğunlukta gözlenmemektedir. Bunun sonucunda haftasonuna denk gelen tatilleri eğitirken aynı yılın bir önceki haftasonu verisini kullanmanın daha isabetli olduğu sonucuna varılmıştır. Bu yaklaşım izlenerek hata oranı yüzde 2 oranında düşürülerek %13 seviyesine indirilebilmektedir.

4.3.3 Farklı Karakteristik İzleyen Tatiller

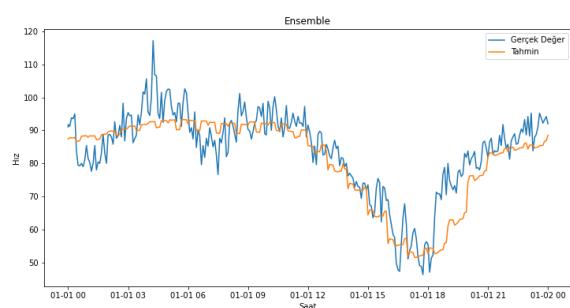
Yılbaşı ve 15 Temmuz Demokrasi Bayramı diğer kategorilere girmeyen bir davranış izlemektedir. Yılbaşı için bunun nedeni 31 Aralık gecesi artan yoğunluk ve 1 Ocak gününün bundan etkilenmesidir. Yılbaşı günü incelendiğinde Şekil 4.17 ve 4.18

grafiklerinde görüleceği üzere gece saatlerinde aşırı trafik yoğunluğu gözlemlenmiştir. Demokrasi bayramına baktığımızda ise 2018 yılında haftasonuna, 2019 yılında da pazartesi gününe denk gelerek 3 günlük bir tatil oluşturmakdadır. Bakıldığından 2018 yılındaki 23 Nisan ve 29 Ekim tatilleriyle aynı kategoriye girse de aynı karakteristiği göstermemektedir.

15 Temmuz eğitim setinde 2018 tek günlük bayramları ve önceki günü, bir hafta önce aynı gün, aynı yılın bir önceki haftasonunu ve aynı yılda o güne kadar olan bayramlar kullanılmıştır. Tek günlük tatillerde izlenen 2018 verisini kullanma yaklaşımında %15 olan hata oranı bu yeni yaklaşımın sayesinde %13 seviyesine kadar gerilemiştir.



Şekil 4.17 1 Ocak 2018



Şekil 4.18 1 Ocak 2019

2018 ve 2019 Yıllarında 1 Ocak Hız Grafikleri

5

Sistem Tasarımı

5.1 Algoritmalar ve Model Eğitimi

Çeşitli makaleler ve modeller incelendiğinde karar ağaçlarının başarısı görülmüştür. Özellikle ani değişim gösteren verilerde XGBoost modeli klasik makine öğrenimi modellerine göre başarı sağlamaktadır. Modele verilecek parametreler, parametre istatistiksel testleri kullanılarak çıkarılmıştır. Veri üzerindeki mevsimsel değişimleri algılamak için parametrelerin sin ve cos değerleri hesaplanmıştır ve bu değerler parametre olarak modellere verilmiştir. Ayrıca, modeller 2018 ve 2019 resmi tatil günleri veriden ayrılarak bu ayırtırılan veri üzerinde eğitilmiştir. Modellerin tatil günlerinde normal günlere oranla çok daha iyi performans verdiği söylenebilir.

5.2 Karar Ağacı Modelleri

Karar ağacı modelleri, birçok makine öğrenimi probleminde kullanılan ve yüksek başarı oranlarına sahip bir sınıflandırma ve regresyon yöntemidir. Karar ağaçları, verileri işlemek ve sonuçları tahmin etmek için bir ağaç yapısı kullanmaktadır.

1. Sınıflandırma Ağaçları: Sınıflandırma ağaçları, verileri sınıflandırmak için kullanılmaktadır. Girdi verileri, ağaç yapısının kök düğümüne girmekte ve her düğümde, bir karar kuralı ile dallara ayrılmaktadır. Bu karar kuralı, bir özellik değeri ile bir eşik değeri arasındaki bir karşılaştırmaya dayanabilmektedir. Bu işlem yaprak düğümüne ulaşınca kadar devam etmektedir. Sonuçta girdi verilerinin sınıflandırılması, yaprak düğümlerinde tanımlanmış olan sınıflara göre belirlenmektedir.
2. Regresyon Ağaçları: Regresyon ağaçları, verileri tahmin etmek için kullanılmaktadır. Girdi verileri, kök düğümünden başlayarak ağaç yapısında ilermektedir. Her düğümde, bir özellik değeri ile bir eşik değeri arasındaki bir karşılaştırmaya dayanan bir karar kuralı kullanılmaktadır. Bu işlem, yaprak düğümüne ulaşınca kadar devam etmektedir. Sonuçta, girdi verilerinin

tahmini, yaprak düğümlerinde tanımlanmış olan sayısal değerlere göre belirlenmektedir.

Karar ağaç modelleri, ayrik veya sürekli özellikleri olan verileri işleyebilirler. Ayrıca, çoklu sınıflandırma veya çoklu regresyon problemlerine de uygulanabilirler. Karar ağaç modelleri, yüksek doğruluk, hızlı tahmin ve kolay yorumlanabilirlik gibi avantajlara sahiptirler.

Regresyon ağaçları, veri setindeki özellikleri kullanarak bağımlı değişkenin (hedef değişken) sürekli bir değerini tahmin etmek için kullanılan bir makine öğrenmesi algoritmasıdır. Bu modeller, karar ağaç algoritmasının bir türüdür ve ağaç yapısı, veri setindeki özelliklerin belirli değerleri tarafından belirli bir hedef değişkeni de ile sonuçlanan karar düğümleri tarafından temsil edilir. Bu karar düğümleri, veri setindeki özelliklerin değerlerine göre sıralanır ve hangi özelliklerin hedef değişkeni üzerinde en büyük etkiye sahip olduğunu belirlemeye yardımcı olur.

Regresyon ağaçları, diğer makine öğrenimi modellerine göre bazı avantajlara sahiptir. Özellikle, bu modeller, veri setindeki özellikler arasındaki etkileşimleri ve doğrusal olmayan ilişkileri belirlemekte iyidirler. Ayrıca, bu modeller, çok sayıda özellik içeren veri setlerinde çalışabilir ve genellikle sonuçları yorumlamak kolaydır. Regresyon ağaçları aynı zamanda birçok farklı varyasyona sahiptir. Bazı örnekler, CART (Classification and Regression Trees), Random Forest ve Gradient Boosting Trees , XGBRegressor gibi algoritmaları içerir. Projemizde model olarak regresyon ağaçları varyasyonlarından RandomForestRegressor, Gradient Boosted Trees ve XGBRegressor kullanılmıştır. Random Forest Regressor, Gradient Boosted Trees (GBM) ve XGBRegressor, üç farklı makine öğrenimi modelidir. Tümü ağaç tabanlı modellerdir ve çeşitli özellikleri ve kullanımı vardır. Random Forest Regressor, birden fazla karar ağaçını eğiterek çalışır. Bu ağaçlar, birbirinden bağımsız olarak oluşturulur ve ardından tahminleri bir araya getirerek son bir tahmin yapılır. Random Forest Regressor, yüksek varyanslı modellerle başa çıkmak için kullanılır.

GBM, Gradient Boosted Trees, tek bir karar ağaçının üzerinde çalışır ve ardışık ağaçlar ekler. Bu nedenle, Random Forest Regressor'dan farklı olarak, GBM aşamalı bir öğrenme modelidir. GBM, düşük biaslı modellerle başa çıkmak için kullanılır. XGBRegressor, GBM'ye benzer şekilde çalışır, ancak öğrenme sürecinde farklı bir teknik kullanır. XGBRegressor, GBM'nin aksine, eğitim örneğine ağırlıklar ekleyerek öğrenir. Bu, modelin daha az hata yapmasına ve daha az ağırlığa sahip örnekleri göz ardı etmesine neden olur. XGBRegressor, GBM'ye kıyasla daha hızlı ve daha doğru sonuçlar verir. Tüm bu modeller ağaç tabanlı olmakla birlikte, eğitim yöntemleri ve özellikleri farklıdır. Seçim, veri setine ve probleme bağlıdır. Random Forest Regressor yüksek

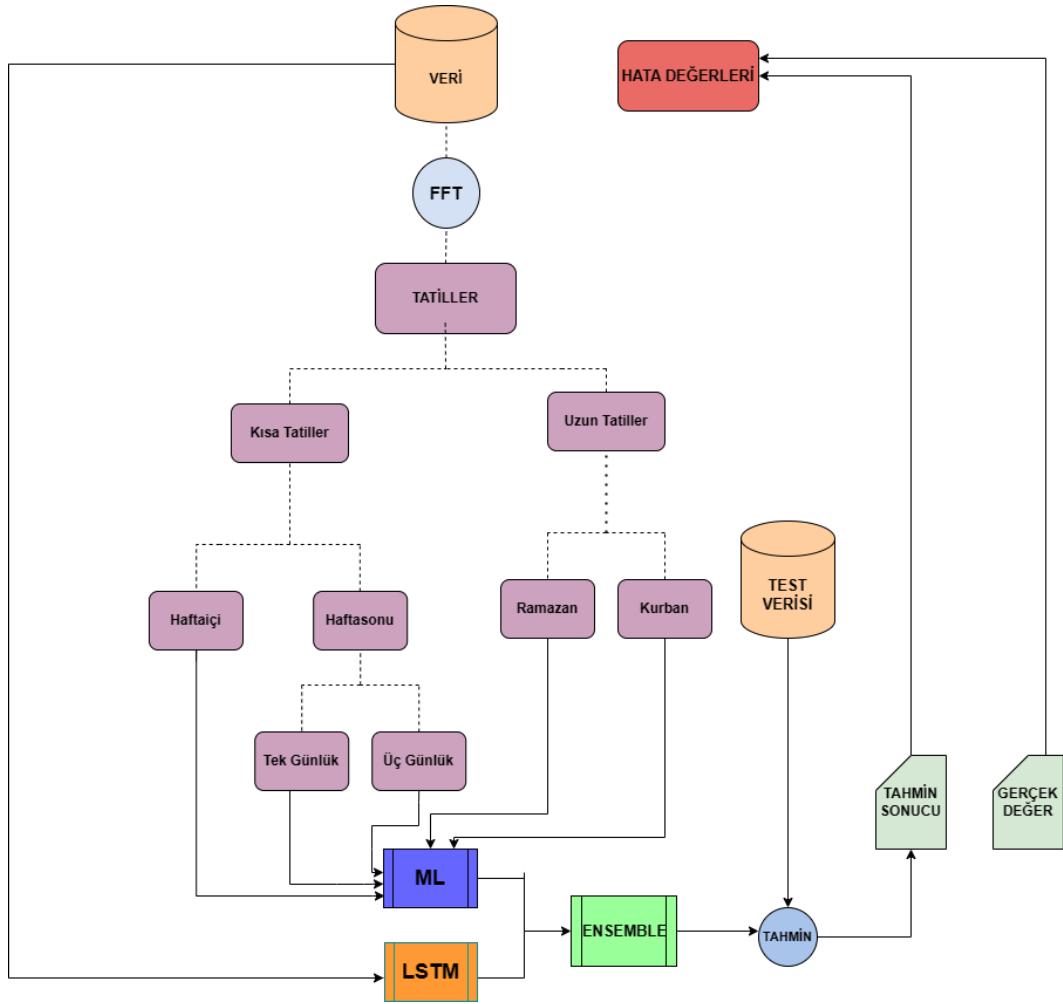
varyanslı verilerle çalışmak için uygunken, GBM düşük biaslı verilerle çalışmak için uygun olabilir ve XGBRegressor daha hızlı ve doğru sonuçlar verir.

5.3 Derin Öğrenme Modelleri

Derin öğrenme, yapay sinir ağlarının daha karmaşık yapılarını kullanarak, büyük veri setlerindeki kompleks yapıları algılama, sınıflandırma, desen tanıma, tahminleme gibi problemleri çözmek için kullanılan bir makine öğrenmesi yöntemidir. Derin öğrenme, sinir ağı mimarileri aracılığıyla, veri setindeki özellikleri otomatik olarak öğrenerek, daha önceki örneklerden öğrendiği bilgiyi yeni veriler üzerinde uygulayarak sonuç üretir. Derin öğrenme yöntemleri, resim ve video işleme, doğal dil işleme, otomatik araba sürüsü, ses tanıma ve makine çevirisi gibi alanlarda başarılı sonuçlar vermektedir. Derin öğrenme modelleri arasında Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Autoencoder, ve Generative Adversarial Networks (GAN) , Deep Neural Networks (DNNs) gibi popüler mimariler yer almaktadır. DNN'ler, diğer derin öğrenme modelleri arasında en genel ve temel yapılardan biridir. Diğer modeller, DNN'lerin farklı varyasyonları olarak kabul edilebilir.

DNN'ler, girdi katmanı, birden fazla gizli katman ve çıkış katmanı gibi temel yapıya sahip sinir ağı modelleridir. Bu katmanlar, birbirleriyle bağlı olan çok sayıda nöron içerir ve birbirlerine ağırlıklı bağlarla bağlanır. Bu ağırlıklar, modelin öğrenme sürecinde belirlenir.

DNN'lerin faydaları, özellikle büyük veri kümeleri üzerinde çalışırken etkileyicidir. Derin sinir ağları, daha önce insanlar tarafından tanımlanmamış özellikleri algılayabilen ve bu özellikleri yorumlayabilme yetisine sahiptir. Derin öğrenme modelleri genel olarak elimizdeki verinin az olmasından kaynaklı olarak klasik makine öğrenimi modellerinden daha kötü bir sonuç vermiştir. Klasik makine öğreniminde verileri özel olarak günlere göre seçme yaklaşımını izlerken derin öğrenme modellerinin çok veriyi daha iyi işleyebilme yeteneğini göz önünde bulundurarak eğitim veri setini daha geniş tutma yaklaşımı izlenmiştir. Sonuç olarak klasik ve derin öğrenme modellerinin farklı şekilde eğitilmeleri gerektiğini düşünüyoruz. bu bilgiler doğrultusunda projede derin öğrenme modeli olarak DNN tercih edilmiştir.



Şekil 5.1 Sistem Tasarım Modeli

5.4 Birleşik Model

Karar ağıacı ve derin öğrenme modeli diyagramda 5.1'deki şekilde birleştirilerek ensemble model haline getirilmiştir. Karar ağıacı modellerine verilen eğitim seti FFT ön işleminden geçirilirken derin öğrenme modeline verilen kirliliği azaltılmış verinin doğrusal modellere çok benzer grafiksel sonuçlar üretmesi ve verideki dalgalanmayı yakalayamaması nedeniyle ön işleme tabi tutulmamış veri ile beslenmiş ve sonuçlar birleştirilmiştir. Daha sonra test verisinde tahmin sonuçları hesaplanıp. Gerçeve veriyle karşılaştırılması yapılarak hata değerleri elde edilmiştir.

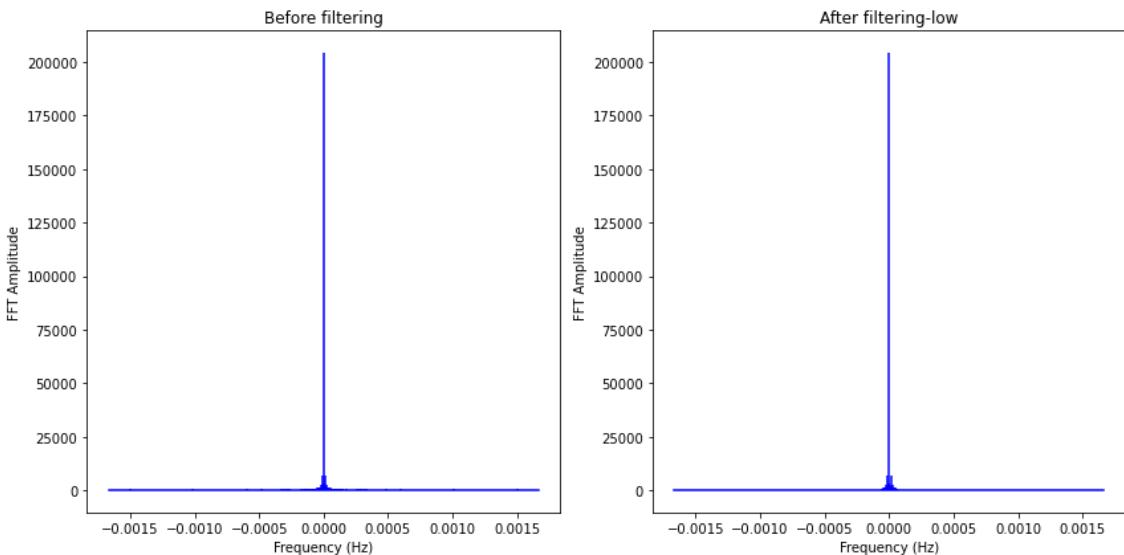
6

Uygulama

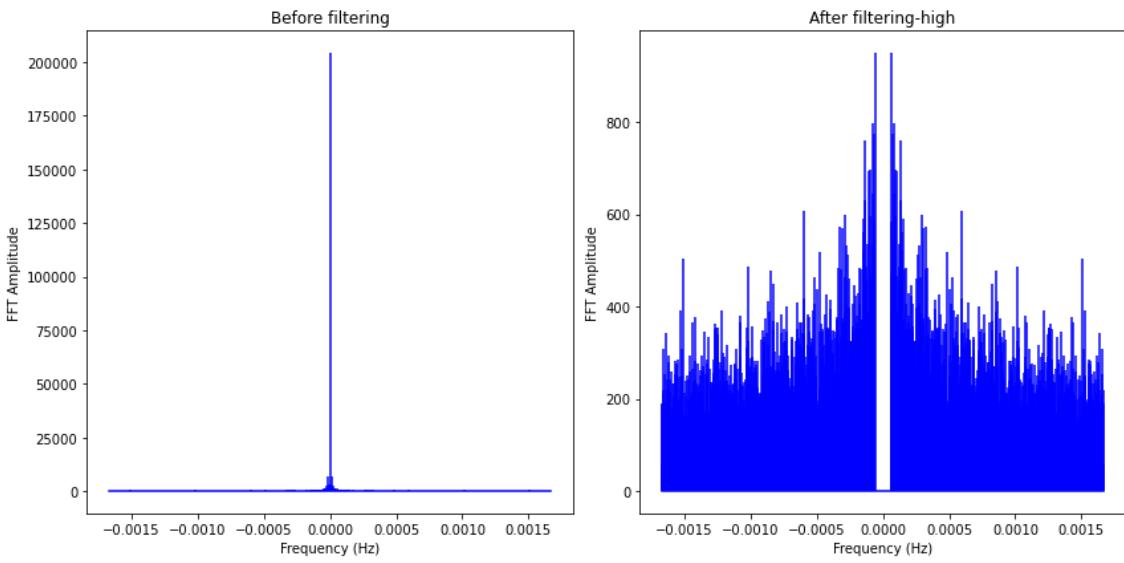
6.1 Fast Fourier Transform Ön İşlemesi

İncelenen DFT-SVR [2] görüldüğü gibi SVR modelini kullanırken eğitim verisine DFT ön işleminin uygulanması eğitim veri setindeki gürültüyü ortadan kaldırmaktadır. Makalede görüldüğü üzere bu işlem SVR modelindeki %13 oranındaki MAPE hata değerini vermektedir, veri setine DFT uygulandığında ise bu oran %10 seviyesine kadar düşmektedir. Bu iyileştirmeden yola çıkarak ensemble modelini DFT uygulananlı veri setiyle eğitmenin iyi sonuçlar vereceği düşünülmüştür. Ancak DFT modelini kullanmak yerine DFT'nin daha hızlı bir versiyonu olan FFT ön işlem metodu bu ensemble modeline uygulanmıştır. Ayrık zamanlı filtrelemede düşük bant aralıklı, yüksek bant aralıklı ve orta bant aralıklı filtreleme yöntemleri literatüre bakıldığından görülebilmektedir. Trafik verisi incelendiğinde düşük bant aralıklı filtrelemenin belirli bir kesme frekansı seçilerek (Şekil 6.1) verideki gürültü çıkarılabilmiştir. Belirli bir bayram aralığı incelendiği için hız verisinin belirgin bir trend eğrisi bulunmamaktadır. Bu nedenle yapılan bu düşük bantlı filtreleme yüksek frekanslı verinin asıl veriden ayrılmasını sağlamaktadır.

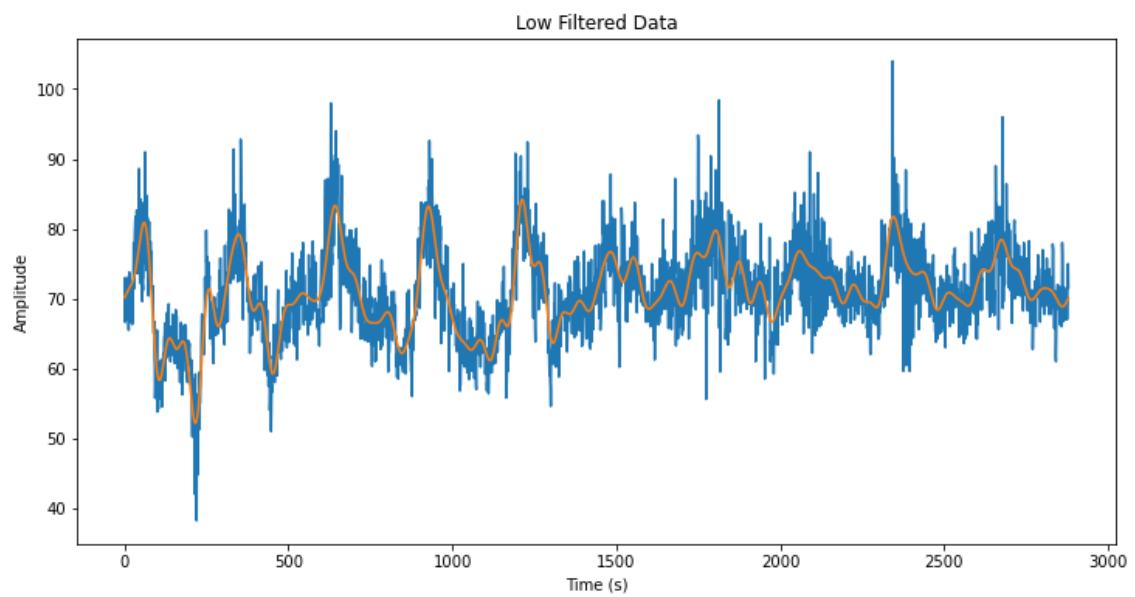
Bu yüksek frekanslı veriye bakıldığından 6.4 istenilen veri setinin ortalamasının 0'a olabildiğince yakın olması gereklidir. Eğer yapılan filtreleme sonucunda yüksek frekanslı veride 0'dan farklı bir ortalama bulunursa bu ortalama anlamlı verinin filtreleme sonucunda kaybedilmiş olduğunu gösterir. Kaybı en aza indirmek için yüksek frekanslı verinin de ayrı bir SVR modelinde eğitilerek ana ensemble modeline eklenmesi planlanmaktadır. Düşük frekanslı veriye bakıldığından Şekil 6.3 verinin ana hatlarının yakalandığı görülebilmektedir.



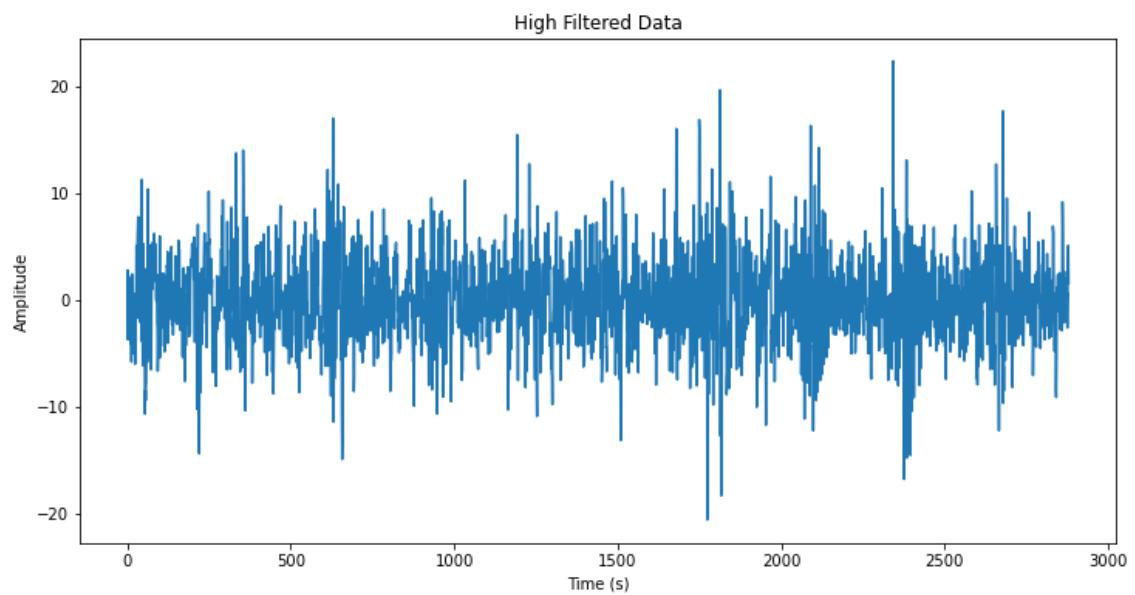
Şekil 6.1 Düşük Bant Aralıklı FFT



Şekil 6.2 Yüksek Bant Aralıklı FFT



Şekil 6.3 Düşük Frekans Filtreleme Uygulanmış Veri ve Gerçek Veri



Şekil 6.4 Yüksek Frekanslı Veri

6.2 Kayıp Değerleri Doldurma

İstanbul'un farklı lokasyonlarından alınan hız verileri tablolarında eksik değerler bulunmaktadır. Tabloda '-1' olarak ifade edilen bu eksik değerler tahminleme sürecinde hata oranının artmasına sebep olmaktadır. Eğitim setinde olan eksik değerler modelin yanlış öğrenme sürecine girmesine ve tahminleme sırasında gerçek değerlerden uzak sonuç üretmesine neden olmaktadır. Test setinde olan eksik değerler ise modelimizin ne kadar iyi çalıştığını anlamamıza engel olmaktadır. Tahminleme sonucunda oluşan değerleri test setindeki veriler ile karşılaştırduğumda hata oranı yüksek çıkmaktadır. Bu sorunların giderilmesi adına test setinden eksik değerlerin çıkarılması, eğitim setinde ise eksik değerlerin doldurulması gerektiği düşünülmüştür. Eğitim setindeki eksik değerlerin doldurulması için ön işleme kısmına fonksiyon yazılmıştır. Fonksiyonunun yazılma süreci ve son adımda çalışması şu şekildedir;

1-Öncelikle train veri setinde -1 ile karşılaşıldığında son 5 günün aynı saatindeki hızların ortalamasını alarak -1 olan kısma set edilmesi denenmiştir. Fakat son 5 gün içerisinde veya tamamında yine aynı saatte -1 değer olma durumunda da hata ile karşılaşılmıştır.

2-Onceki adımda alınan(-1) değer olmayan son 5 günü bulana kadar while ile geriye gidilerek değer bulma ve ortalama alarak set etme işlemi yapılmıştır. Fakat (-1) değerinin olduğu gün yılın ilk 5 günü olması veya geriye gitme işlemi yaparken yılın ilk gününe giderek daha geriye gidememe sorunu ile karşılaşılmıştır.

3-Son adımda çözüm olarak (-1) değeri bulunan zaman yılın ilk 5 günü ise ileri doğru gidilerek (-1) olmayan 5 günün ortalaması alınmıştır, yılın ilk 5 günü değil ise geriye giderek (-1) olmayan 5 gün bulunamadığında ileri doğru gidilerek (-1) olmayan 5 günün ortalaması alınmıştır.

6.3 Random Forest Regressor

RandomForestRegressor, bir makine öğrenimi algoritması olan Random Forest'un bir varyasyonudur ve regresyon problemlerinde kullanılmaktadır. Bu algoritma, birden fazla karar ağacını (decision tree) eğiterek tahmin yapmaktadır. Random Forest algoritması, öncelikle birbirinden bağımsız ve rastgele özellikler kümesi oluşturarak her bir karar ağacını eğitir. Bu rastgele özellikler kümesi, tüm özellikler kümesinden örneklem alınarak oluşturulur. Daha sonra, her bir karar ağacı, rastgele seçilen bir alt kümesi ile örneklem alınmış veri kümesini kullanarak eğitilir. Bu sayede, farklı alt örneklem ve özellik kümeleri kullanarak eğitilmiş birden fazla karar ağacı elde edilir. Random Forest'un tahmin yapmak için kullandığı yöntem, tüm karar ağaçlarının

ürettiği tahminleri bir araya getirerek (ensemble learning) ortalama bir tahmin değeri oluşturmaktır. Bu şekilde, tek bir ağacın hatalarından etkilenmeyi engelleyerek daha doğru bir tahmin elde edilir. RandomForestRegressor, regresyon problemlerinde kullanılan bir varyasyondur. Bu algoritma, aynı şekilde birden fazla karar ağacını kullanarak tahmin yapar ve tüm ağaçların tahminlerini bir araya getirerek ortalama bir tahmin değeri oluşturur. Ancak bu tahminler artık sınıflar değil, gerçek sayısal değerlerdir.

Projede Random Forest ana model oalrak kullanılmıştır. Bu kararın alınmasındaki asıl neden random forestin parametre değişimine olan toleransı ve verdiği hata değerlerinin güvenilirliği olmasıdır. Asıl amaç tatil günlerini etkileyen günlerin belirlenmesi olduğu için random forest temel model olarak iyi bir sonuç vermektedir.

Randem Forest için en uygun parametre seçimi yapılarak mevcut model içim daha iyi sonuçların elde edilmesi hedeflenmiş, bu doğrultuda 106 segment için sırası ile Rastgele Arama Çapraz Doğrulama yöntemi kullanılmış, belirtilen hiperparametre dağılımlarından rastgele seçilen 10 farklı kombinasyonu değerlendirilmiştir. Bu, 10 farklı hiperparametre setini deneyerek en iyi yapılandırma bulmunya çalışılmıştır. Belirtilen hipermatreler aşağıdaki gibidir ;

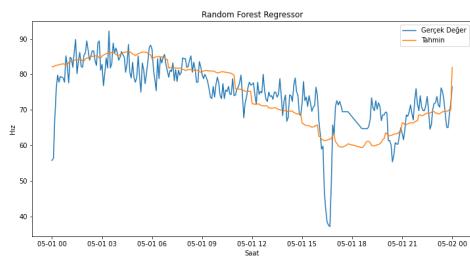
```
n_estimators: [100, 500, 1000]
max_features: ['sqrt', 'log2', 1.0]
max_depth: randint(2, 10)
min_samples_split: randint(2, 20)
min_samples_leaf: randint(1, 20)
bootstrap: [True, False]
```

Sonuç olarak en iyi parametre değerleri aşağıdaki şekilde belirlenmiştir;

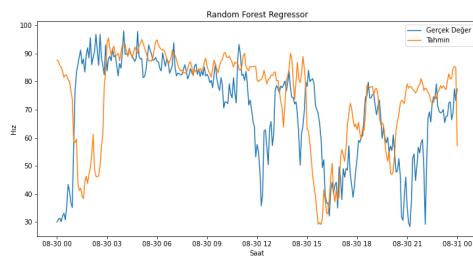
```
n_estimators: [500]
max_features: [1.0]
max_depth: [19]
min_samples_split: [11]
min_samples_leaf: [19]
bootstrap: [True]
```

Kısa Bayramlar

Kısa bayram olarak sınıflandırılan 23 Nisan, 1 Mayıs, 19 Mayıs, 15 Temmuz, 30 Ağustos, 29 Ekim tarihlerindeki tatillerin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;

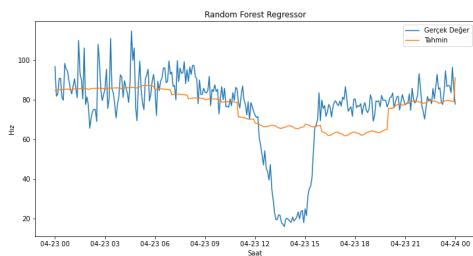


Şekil 6.5 1 Mayıs 2019 Random Forest

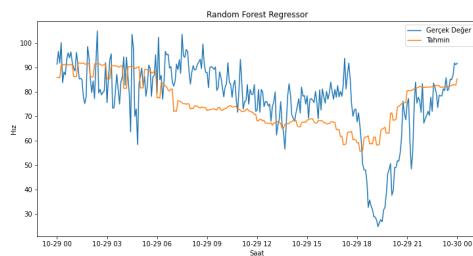


Şekil 6.6 30 Ağustos 2019 Random Forest

Avcılar 414 Nolu Segment Haftaiçine Denk Gelen Bayramlar

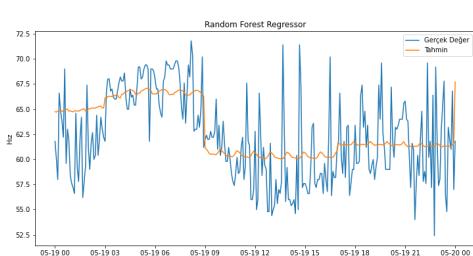


Şekil 6.7 23 Nisan 2019 Random Forest

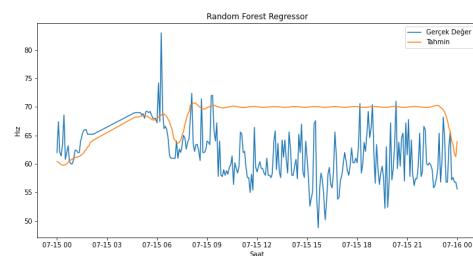


Şekil 6.8 29 Ekim 2019 Random Forest

Ataşehir 316 Nolu Segment Haftasonu İle Birleşen Bayramlar



Şekil 6.9 19 Mayıs 2019 Random Forest

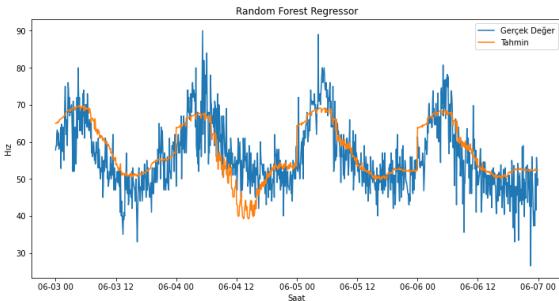


Şekil 6.10 15 Temmuz 2019 Random Forest

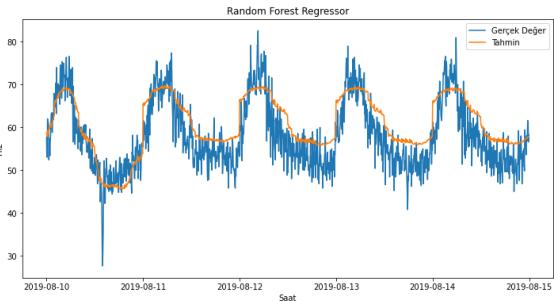
Kadıköy 2102 Nolu Segment Farklı Karakteristik İzleyen Bayramlar

Uzun Bayramlar

Uzun bayram olarak sınıflandırılan ramazan bayramı ve kurban bayramı tatillerinin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;



Şekil 6.11 Ramazan Bayramı 2019
Random Forest



Şekil 6.12 Kurban Bayramı 2019 Random
Forest

Kadıköy 2102 nolu Uzun Bayramlar

6.4 XGBoost

XGBoost (eXtreme Gradient Boosting), açık kaynak kodlu bir makine öğrenimi kütüphanesidir ve Gradient Boosting yöntemini kullanarak yüksek performanslı tahmin modelleri oluşturmayı hedefler. XGBoost, düzenlileştirme (regularization), paralelleştirme ve hızlı ölçeklenebilirlik özellikleri sayesinde, özellikle yapısal olarak büyük veri kümeleri üzerinde yüksek performanslı modelleme yapmak için tercih edilir.

En uygun parametre değerlerini belirlemek için genellikle çapraz doğrulama (cross-validation) kullanılır. Bu işlem, veri kümesini birden fazla parçaya ayırarak (genellikle 5 veya 10 parçaaya bölünür), her parçanın sırayla test seti olarak kullanılması ve diğer parçaların eğitim seti olarak kullanılması yoluyla yapılır. Bu yöntem, modelin performansını değerlendirmek ve aynı zamanda en uygun parametre değerlerini belirlemek için kullanılır.

Bu amaçla, GridSearchCV veya RandomizedSearchCV gibi bir hiperparametre ayarlama kütüphanesi kullanılabilir. Örneğin, aşağıdaki kod, n-estimators için farklı değerlerin denendiği ve en iyi değerin belirlendiği bir GridSearchCV örneği göstermektedir: Bizim projemizde param-grid aşağıdaki gibi tanıplanıp, param-grid = 'n-estimators': [50, 100, 150, 200], Burada param-grid değişkeninde belirtilen n-estimators değerleri denenecek ve en iyi değer GridSearchCV(grid-search = GridSearchCV(xgb-model, param-grid, cv=5, verbose=2)) yöntemi kullanılarak belirlenmiştir. cv parametresi, çapraz doğrulama için kaç parçaya bölüneceğini belirtir. Bu örnekte, 5 kullanılmıştır. verbose parametresi, çıktıının ayrıntı düzeyini belirtir. Bu örnekte, 2 kullanılmıştır.

XGBRegressor, XGBoost adı verilen açık kaynaklı bir makine öğrenimi kütüphanesi tarafından sağlanan bir algoritmadır. XGBoost, ağaç tabanlı bir öğrenme

algoritmasıdır ve Gradient Boosting yöntemini uygular. Bu yöntem, zayıf öğrenicileri (ağaçları) birden fazla kez eğiterek güçlü bir öğrenici oluşturur. XGBRegressor, verilerin özelliklerini ve hedef değişkenlerini kullanarak ağaç tabanlı bir model oluşturur. Model, verilerdeki özelliklerin birleşimlerine dayalı olarak kararlar alır. XGBRegressor, Gradient Boosting yöntemini uygularken aynı zamanda Regularization ve Shrinkage gibi teknikleri de kullanır. Bu teknikler, modelin overfitting yapmasını öner ve daha iyi genelleştirme yapmasını sağlar. XGBRegressor, birçok hiperparametreye sahiptir. Bu parametreler arasında ağaçın derinliği, ağaç sayısı, öğrenme oranı ve daha birçok parametre bulunur. Bu parametreler, modelin eğitimi ve performansı üzerinde etkilidir. En uygun hiperparametrelerin belirlenmesi, modelin performansını artırabilir. Kodörneğinde XGBRegressor kullanılarak, 2018 ve 2019 yıllarına ait veriler kullanılarak bir hız tahmini yapılmıştır. Model, verilerin özelliklerini (saat ve gün) kullanarak, hedef değişkeni (hız) tahmin etmiştir. Model, diğer makine öğrenimi algoritmaları gibi, eğitim verileri üzerinde eğitilmiş ve ardından test verilerinde performansını ölçmek için kullanılmıştır.

XGBRegressor için en uygun parametre seçimi yapılarak mevcut model içim daha iyi sonuçların elde edilmesi hedeflenmiş, bu doğrultuda 106 segment için sırası ile Rastgele Arama Çapraz Doğrulama yöntemi kullanılmış, belirtilen hiperparametre dağılımlarından rastgele seçilen 10 farklı kombinasyonu değerlendirilmiştir. Bu, 10 farklı hiperparametre setini deneyerek en iyi yapılandırma bulmunya çalışılmıştır. Belirtilen hipermateler aşağıdaki gibidir ;

```
n_estimators: [100, 500, 1000]  
learning_rate : [0.01, 0.1, 0.2, 0.3]  
max_depth : randint(2, 10)  
min_child_weight : randint(1, 10)  
subsample : [0.5, 0.6, 0.7, 0.8, 0.9, 1.0]  
colsample_bytree : [0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
```

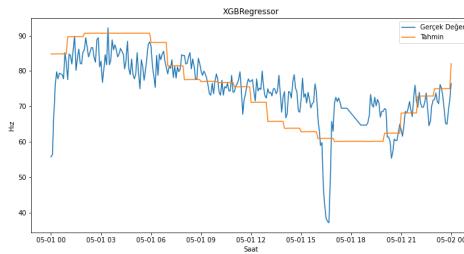
Sonuç olarak en iyi parametre değerleri aşağıdaki şekilde belirlenmiştir;

```
n_estimators: [500]  
learning_rate: [0.01]  
max_depth: [2]  
min_child_weight: [8]  
subsample: [0.5]
```

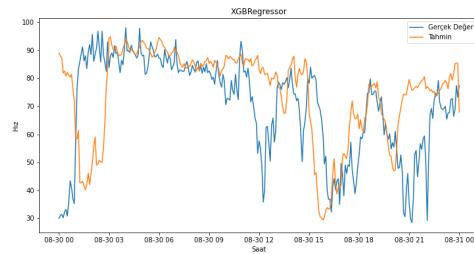
colsample_bytree: [0.7]

Kısa Bayramlar

Kısa bayram olarak sınıflandırılan 23 Nisan, 1 Mayıs, 19 Mayıs, 15 Temmuz, 30 Ağustos, 29 Ekim tarihlerindeki tatillerin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;

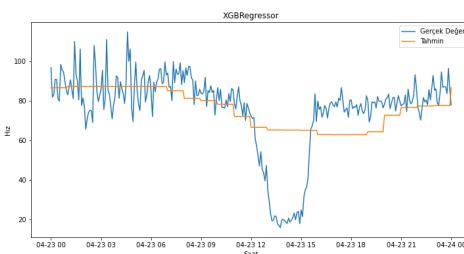


Şekil 6.13 1 Mayıs 2019 XGBoost

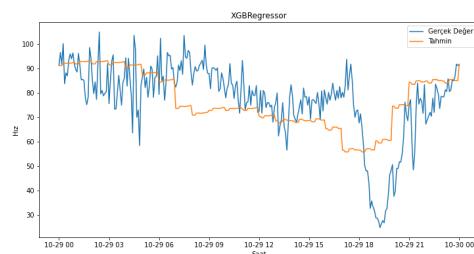


Şekil 6.14 30 Ağustos 2019 XGBoost

Avcılar 414 Nolu Segment Haftaiçine Denk Gelen Bayramlar

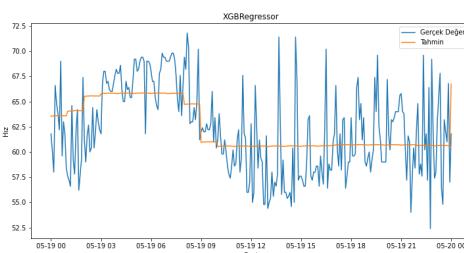


Şekil 6.15 23 Nisan 2019 XGBoost

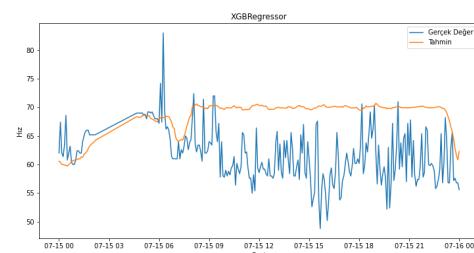


Şekil 6.16 29 Ekim 2019 XGBoost

Ataşehir 316 Nolu Segment Haftasonu İle Birleşen Bayramlar



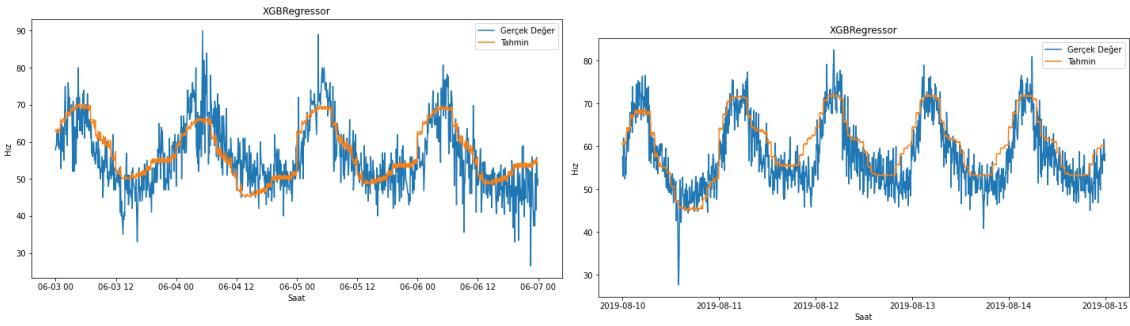
Şekil 6.17 19 Mayıs 2019 XGBoost



Şekil 6.18 15 Temmuz 2019 XGBoost

Kadıköy 2102 Nolu Segment Farklı Karakteristik İzleyen Bayramlar

Uzun Bayramlar Uzun bayram olarak sınıflandırılan ramazan bayramı ve kurban bayramı tatillerinin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;



Şekil 6.19 Ramazan Bayramı 2019
XGBoost

Şekil 6.20 Kurban Bayramı 2019 XGBoost

Kadıköy Ankara Asfaltı Uzun Bayramlar

6.5 Gradient Boosting

Gradient Boosted Trees (GBT) bir makine öğrenimi algoritmasıdır ve ağaç tabanlı bir öğrenme yöntemidir. GBT, birden fazla zayıf öğrenicinin bir araya gelerek güçlü bir öğrenici oluşturmasına dayanır. GBT, aşamalı bir şekilde, bir önceki aşamadaki hatayı en aza indirmeye çalışarak ağaçlar oluşturur. Her aşamada, ağaçlar modelin zayıf yönlerini düzeltmek için oluşturulur. Her ağaç, önceki ağaçların tahminlerinin hatalarını hedefler. Bu hataların daha sonra diğer ağaçların oluşturulması için kullanılacak yeni bir hedef değişkeni olarak kullanılır.

GBT için en uygun parametre seçimi yapılarak mevcut model içim daha iyi sonuçların elde edilmesi hedeflenmiş, bu doğrultuda 106 segment için sırası ile Rastgele Arama Çapraz Doğrulama yöntemi kullanılmış, belirtilen hiperparametre dağılımlarından rastgele seçilen 10 farklı kombinasyonu değerlendirilmiştir. Bu, 10 farklı hiperparametre setini deneyerek en iyi yapılandırma bulmuna çalışılmıştır. Belirtilen hipermateler aşağıdaki gibidir ;

```
n_estimators: [100, 500, 1000]
learning_rate: [0.01, 0.1, 0.2, 0.3]
max_depth: randint(2, 10)
min_samples_split : randint(2, 20)
min_samples_leaf : randint(1, 20)
```

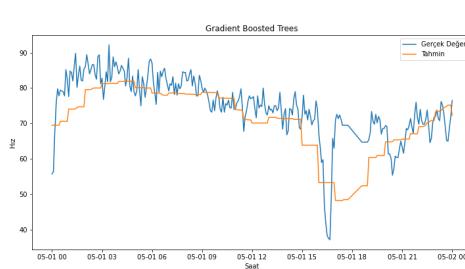
Sonuç olarak en iyi parametre değerleri aşağıdaki şekilde belirlenmiştir;

```
n_estimators: [500]
learning_rate: [0.01]
max_depth: [2]
min_samples_split : [5]
```

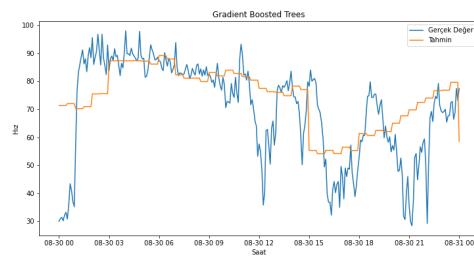
min_samples_leaf : [3]

Kısa Bayramlar

Kısa bayram olarak sınıflandırılan 23 Nisan, 1 Mayıs, 19 Mayıs, 15 Temmuz, 30 Ağustos, 29 Ekim tarihlerindeki tatillerin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;

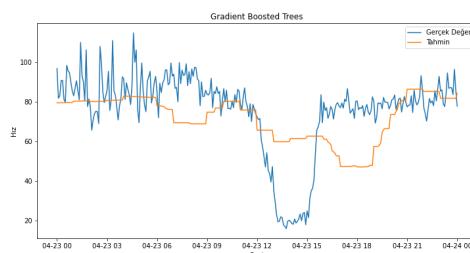


Şekil 6.21 1 Mayıs 2019 Gradient Boosted

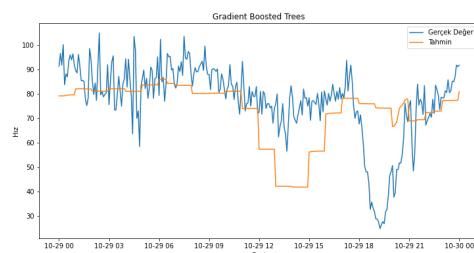


Şekil 6.22 30 Ağustos 2019 Gradient Boosted

Avcılar 414 Nolu Segment Haftaiçine Denk Gelen Bayramlar



Şekil 6.23 23 Nisan 2019 Gradient Boosted



Şekil 6.24 29 Ekim 2019 Gradient Boosted

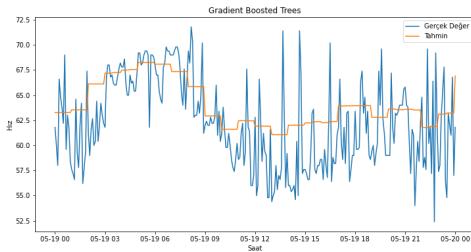
Ataşehir 316 Nolu Segment Haftasonu İle Birleşen Bayramlar

Uzun Bayramlar

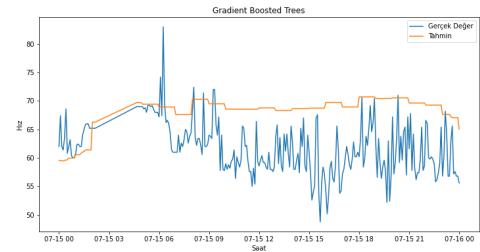
Uzun bayram olarak sınıflandırılan ramazan bayramı ve kurban bayramı tatillerinin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;

6.6 CatBoost

CatBoost, bir ağaç tabanlı bir öğrenme algoritmasıdır ve Gradient Boosting Framework (GBM) ailesinin bir üyesidir. Çalışma prensibi, Boosting yöntemi ile verileri modellemek ve tahmin etmek için bir dizi karar ağıacı (decision tree) kullanmaktadır.

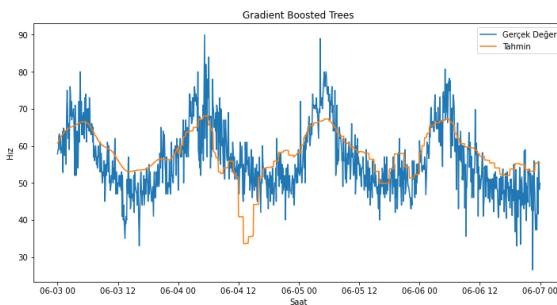


Şekil 6.25 19 Mayıs 2019 Gradient Boosted

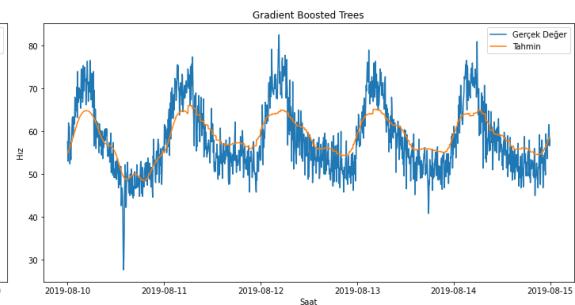


Şekil 6.26 15 Temmuz 2019 Gradient Boosted

Kadıköy 2102 Nolu Segment Farklı Karakteristik İzleyen Bayramlar



Şekil 6.27 Ramazan Bayramı 2019 Gradient Boosted



Şekil 6.28 Kurban Bayramı 2019 Gradient Boosted

Kadıköy Ankara Asfaltı 2102'nolu Segment Uzun Bayramlar

CatBoost için en uygun parametre seçimi yapılarak mevcut model içim daha iyi sonuçların elde edilmesi hedeflenmiş, bu doğrultuda 106 segment için sırası ile Rastgele Arama Çapraz Doğrulama yöntemi kullanılmış, belirtilen hiperparametre dağılımlarından rastgele seçilen 10 farklı kombinasyonu değerlendirilmiştir. Bu, 10 farklı hiperparametre setini deneyerek en iyi yapılandırma bulmunya çalışılmıştır. Belirtilen hipermateler aşağıdaki gibidir ;

```

iterations: [100, 500, 1000]
depth: [4, 6, 8, 10] learning_rate: [0.01, 0.1, 0.2]
l2_lead, eg : [1, 3, 5, 7]
border_count : [32, 64, 128]
bagging_temperature : [0.5, 0.8, 1.0]

```

Sonuç olarak en iyi parametre değerleri aşağıdaki şekilde belirlenmiştir; iterations: [500]
 depth: [4]
 learning_rate: [0.01]

```
l2_leadreg : [3]  
border_count : [128]  
bagging_temperature :: [0.5]
```

- iterations: Modelin toplam tekrar sayısını belirtir.
- learning-rate: Her bir adımda model tarafından kullanılan öğrenme oranını belirtir.
- depth: Her bir ağacın maksimum derinliğini belirler.
- l2-leaf-reg: L2 düzenleme parametresi.
- bagging-temperature: Bagging sıcaklığı parametresi.
- random-strength: Randomness kuvveti, özellikle bootstrap sırasında etkilidir.
- border-count: Belirli bir özelliğin ayrıştırılması için kullanılan en az özellik sayısı.
- thread-count: Modelin çalışması için kullanılan iş parçacığı sayısı.
- early-stopping-rounds: Eğitim işlemi sırasında overfitting'i engellemek için kullanılan erken durdurma sayısıdır.

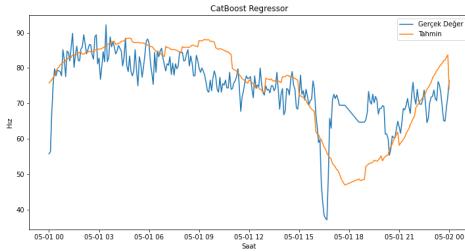
Bu parametrelerin her biri modelin farklı yönlerini kontrol eder ve modelin performansını artırmak için ayarlanabilir. Bununla birlikte, optimal parametre ayarlaması, genellikle deneme yanılma yöntemleri kullanılarak gerçekleştirilir. Projemizde (iterations=1000, learning-rate=0.1, depth=6) değişikleri yapılarak model parametreleri değiştirilmiştir

Kısa Bayramlar

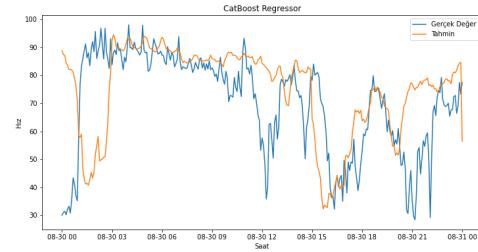
Kısa bayram olarak sınıflandırılan 23 Nisan, 1 Mayıs, 19 Mayıs, 15 Temmuz, 30 Ağustos, 29 Ekim tarihlerindeki tatillerin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;

Uzun Bayramlar

Uzun bayram olarak sınıflandırılan ramazan bayramı ve kurban bayramı tatillerinin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;

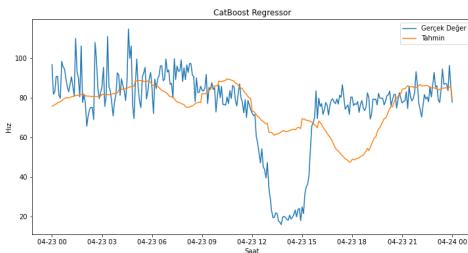


Şekil 6.29 1 Mayıs 2019 Catboost

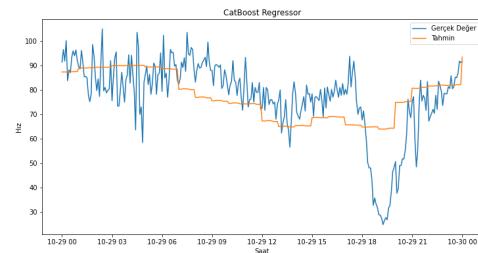


Şekil 6.30 30 Ağustos 2019 Catboost

Avcılar 414 Nolu Segment Haftaiçine Denk Gelen Bayramlar



Şekil 6.31 23 Nisan 2019 Catboost



Şekil 6.32 29 Ekim 2019 Catboost

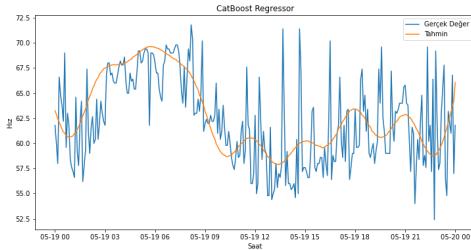
Ataşehir 316 Nolu Segment Haftasonu İle Birleşen Bayramlar

6.7 LGBMRegressor

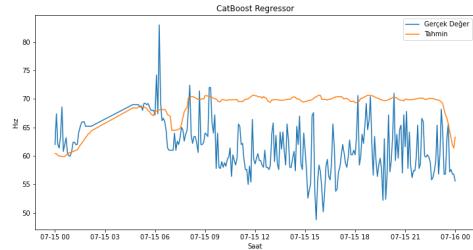
LGBMRegressor, LightGBM kütüphanesi tarafından sağlanan bir sınıflandırma ve regresyon algoritmasıdır. LightGBM, yüksek performanslı ve dağıtılmış bir makine öğrenimi kütüphanesidir ve Gradient Boosting Framework'ünü kullanarak veri kümesi üzerindeki tahminleri geliştirir. LGBMRegressor, GBM'deki ağaç temelli modelleme stratejisini kullanır ve bu modelleme stratejisi, verilerin özelliklerine ve hedefine dayalı olarak ağaçları büyütmek için özel bir eğitim işlemi kullanır. Bu, LightGBM'in diğer GBM modellerine göre daha hızlı olmasını sağlar.

LGBM için en uygun parametre seçimi yapılarak mevcut model içim daha iyi sonuçların elde edilmesi hedeflenmiş, bu doğrultuda 106 segment için sırası ile Rastgele Arama Çapraz Doğrulama yöntemi kullanılmış, belirtilen hiperparametre dağılımlarından rastgele seçilen 10 farklı kombinasyonu değerlendirilmiştir. Bu, 10 farklı hiperparametre setini deneyerek en iyi yapılandırma bulmuna çalışılmıştır. Belirtilen hipermateler aşağıdaki gibidir ;

```
n_estimators: [100, 500, 1000]
learning_rate: [0.01, 0.1, 0.2, 0.3]
max_depth: randint(2, 10)
minchildsamples : randint(1, 10)
```

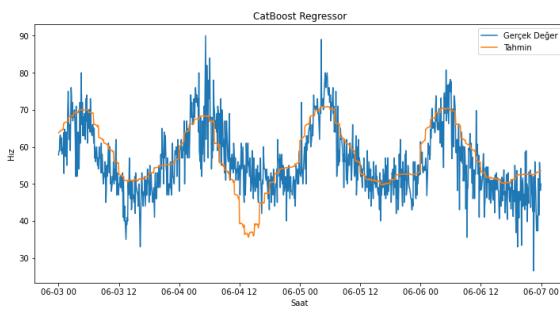


Şekil 6.33 19 Mayıs 2019 Catboost

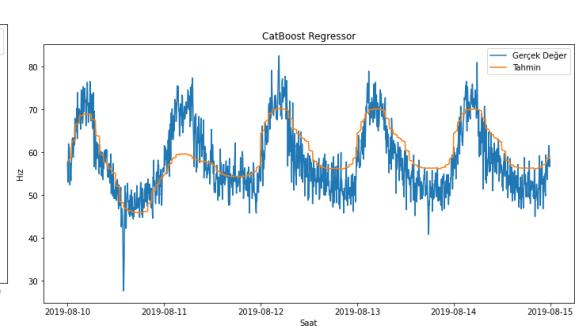


Şekil 6.34 15 Temmuz 2019 Catboost

Kadıköy 2102 Nolu Segment Farklı Karakteristik İzleyen Bayramlar



Şekil 6.35 Ramazan Bayramı 2019 Catboost



Şekil 6.36 Kurban Bayramı 2019 Catboost

Kadıköy Ankara Asfaltı 2102'nolu Segment Uzun Bayramlar

```

subsample : [0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
colsample_bytree : [0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
reg_alpha : [0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
reg_lambda : [0.5, 0.6, 0.7, 0.8, 0.9, 1.0]

```

Sonuç olarak en iyi parametre değerleri aşağıdaki şekilde belirlenmiştir;

```

n_estimators: [500]
learning_rate: [0.01]
max_depth : [6]
min_child_samples : [5]
subsample : [1]
colsample_bytree : [0.6]
reg_alpha : [0.01]
reg_lambda : [0]

```

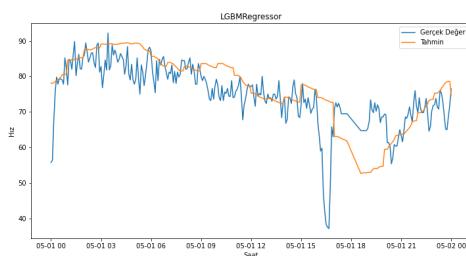
- objective: Modelin optimize edeceği amaç fonksiyonu.
- boosting-type: Gradient boosting tipi.

- n-estimators: Boosting iterasyonları sayısı.
- learning-rate: Boosting algoritmasının her iterasyonunda kullanacağı öğrenme oranı.
- max-depth: Karar ağacı derinliği sınırı. Burada, 5 olarak belirtilmiştir.
- min-child-samples: Her bir yaprak düğümünde en az kaç veri örneğinin bulunması gereği.
- subsample: Her bir iterasyonda kullanılacak alt örneklem oranı.
- colsample-bytree: Her bir iterasyonda kullanılacak özelliklerin oranı.
- n-jobs: Eğitim işleminin paralel olarak çalıştırılacağı işlemci sayısı.
- random-state: Modelin tekrarlanabilir sonuçlar vermesi için kullanılan rastgele sayı üretici için tohum değeri.

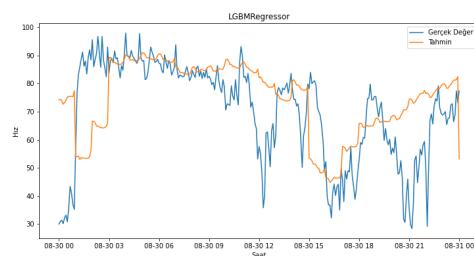
Projemizde bu metod kullanılırken parametre değerleri aşağıdaki gibi verilmiştir; 'objective': 'regression', 'boosting-type': 'gbdt', 'n-estimators': 100, 'learning-rate': 0.1, 'max-depth': 5, 'min-child-samples': 20, 'subsample': 0.8, 'colsample-bytree': 0.8, 'n-jobs': -1, 'random-state': 0

Kısa Bayramlar

Kısa bayram olarak sınıflandırılan 23 Nisan, 1 Mayıs, 19 Mayıs, 15 Temmuz, 30 Ağustos, 29 Ekim tarihlerindeki tatillerin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;



Şekil 6.37 1 Mayıs 2019 LightGBM

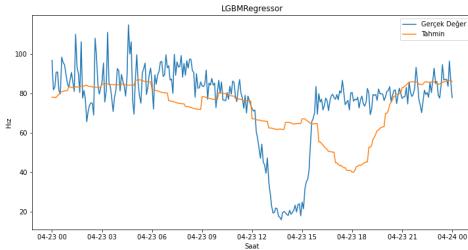


Şekil 6.38 30 Ağustos 2019 LightGBM

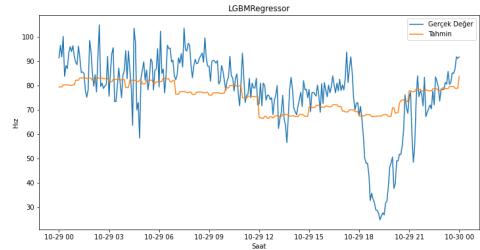
Avcılar 414 Nolu Segment Haftaiçine Denk Gelen Bayramlar

Uzun Bayramlar

Uzun bayram olarak sınıflandırılan ramazan bayramı ve kurban bayramı tatillerinin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;

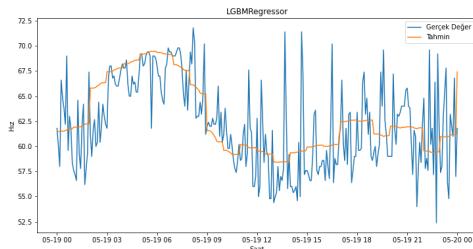


Şekil 6.39 23 Nisan 2019 LightGBM

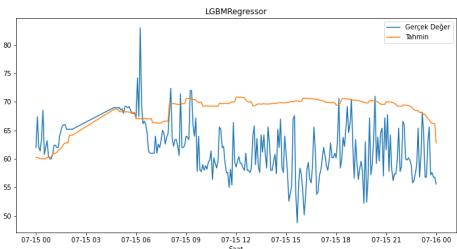


Şekil 6.40 29 Ekim 2019 LightGBM

Ataşehir 316 Nolu Segment Haftasonu İle Birleşen Bayramlar



Şekil 6.41 19 Mayıs 2019 LightGBM



Şekil 6.42 15 Temmuz 2019 LightGBM

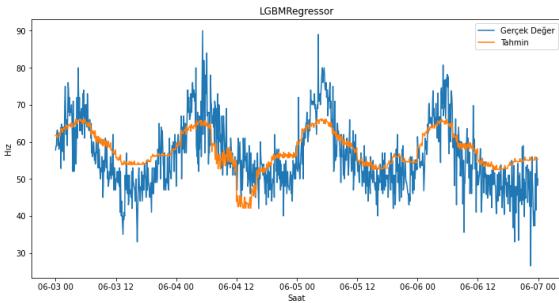
Farklı Karakteristik İzleyen Bayramlar

6.8 DNNs

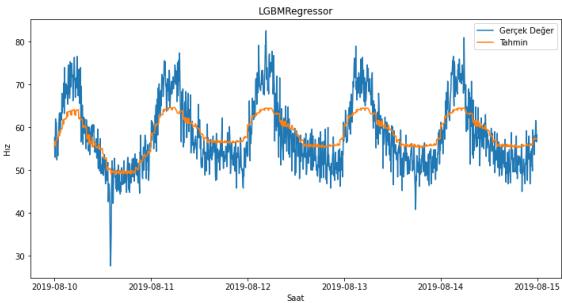
Deep Neural Networks (DNNs) veya derin sinir ağları, yapay sinir ağı mimarisinin bir türüdür ve birçok katmandan oluşur. Giriş katmanı verileri alır, çıktı katmanı sonuçları verir ve ara katmanlar arasında bilgi işlemeyi gerçekleştirir. Her bir katmanın nöronları, girdi değerleriyle ağırlıkların çarpılmasını ve bir aktivasyon fonksiyonuna sokulmasını içeren matematiksel işlemler yapar. Bu işlemler sonucunda çıktılar, bir sonraki katmana veya çıktı katmanına aktarılır. DNN'ler, yüksek ölçekli, karmaşık veri kümelerinde yapısal özelliklerini bulmak ve öğrenmek için kullanılır. Örneğin, bir görüntü tanıma modeli için, ilk katmanlar genellikle basit özellikleri (çizgiler, kenarlar, renkler) algılamak için tasarlanırken, daha sonraki katmanlar bu özelliklerin daha karmaşık kombinasyonlarını öğrenerek daha yüksek seviyeli özellikler (objeler, yüzler) algılamak için kullanılır. DNN'ler, gerçek dünya problemlerinde başarılı sonuçlar vermek için geniş bir veri kümesi ve yeterli hesaplama gücü gerektirir. Günümüzde DNN'ler, görüntü ve dil işleme, nesne tespiti, doğal dil işleme, otomatik sürüs, oyun oynama ve daha birçok alanda kullanılır.

DNN'lerin temel parametreleri şunlardır

- Topoloji: DNN'lerin birçok katmanı vardır ve her bir katmanın sayısı, boyutu ve türü belirlenmelidir. Ayrıca, her katmanın ne kadar öğrenme yapması gereği



Şekil 6.43 Ramazan Bayramı 2019
LightGBM



Şekil 6.44 Kurban Bayramı 2019
LightGBM

Kadıköy Ankara Asfaltı Uzun Bayramlar

de belirlenir.

- Aktivasyon fonksiyonu: Aktivasyon fonksiyonları, her katmanın çıktısını hesaplamak için kullanılır. ReLU, sigmoid, tanh ve softmax gibi çeşitli aktivasyon fonksiyonları mevcuttur.
- Öğrenme hızı: Bu, ağıın öğrenme hızını kontrol etmek için kullanılan bir parametredir. Öğrenme hızı ne kadar yüksek olursa, ağı daha hızlı öğrenir ancak daha yüksek bir hata oranına sahip olabilir.
- Mini-batch boyutu: Bu, ağıın eğitim verilerini kaç parçaya ayıracığını belirler. Mini-batch boyutu arttıkça, ağı daha hızlı öğrenir ancak daha az hassas bir sonuç verebilir.
- Epoch sayısı: Bir epoch, tüm verilerin ağıda bir kez eğitilmesini ifade eder. Epoch sayısı, ağıın ne kadar süre eğitileceğini belirler.
- Optimizasyon algoritması: Ağıın eğitilmesi sırasında kullanılan optimizasyon algoritması, ağıın hızını ve doğruluğunu etkiler. Stokastik gradyan inişi (SGD), ADAM ve RMSProp gibi çeşitli optimizasyon algoritmaları mevcuttur.
- Dropout: Dropout, ağıın aşırı öğrenmesini önlemek için kullanılan bir tekniktir. Bu, rastgele birimlerin devre dışı bırakıldığı bir yöntemdir ve ağıın daha iyi genelleme yapmasına yardımcı olabilir.

Projemizde kullanılan parametreler şu şekildedir:

```
model.add(Dense(128, activation="relu", input_dim = X_train.shape[1]))
model.add(Dropout(0.3))
model.add(Dense(64, activation = "relu"))
model.add(Dropout(0.3))
```

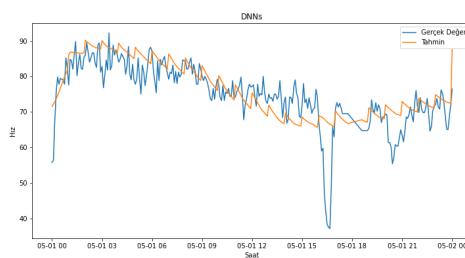
```

model.add(Dense(1, activation = "linear"))
model.compile(loss = "mean_absolute_error", optimizer = Adam(lr = 0.001))
history = model.fit(X_train, y_train, epochs = 100, batch_size = 32, verbose = 0)

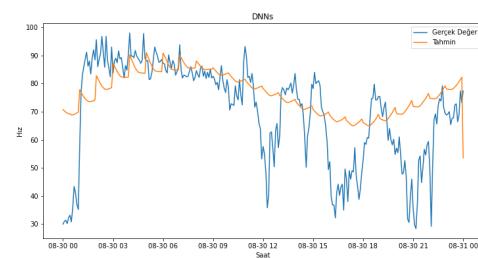
```

Kısa Bayramlar

Kısa bayram olarak sınıflandırılan 23 Nisan, 1 Mayıs, 19 Mayıs, 15 Temmuz, 30 Ağustos, 29 Ekim tarihlerindeki tatillerin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;

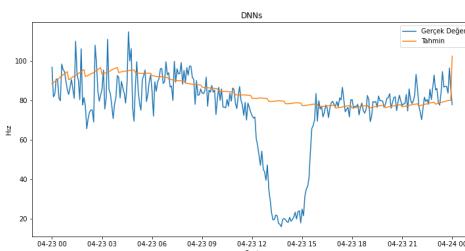


Şekil 6.45 1 Mayıs 2019 DNNs

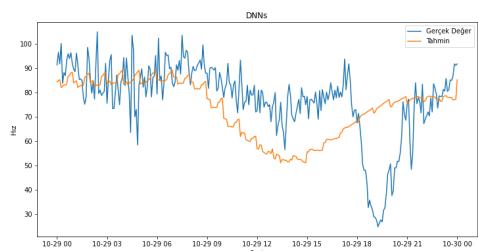


Şekil 6.46 30 Ağustos 2019 DNNs

Avcılar 414 Nolu Segment Haftaiçine Denk Gelen Bayramlar

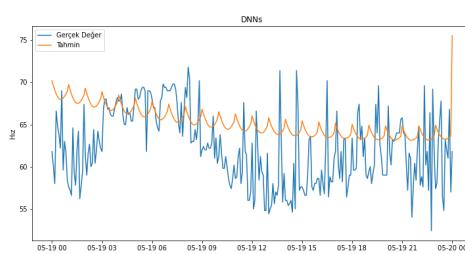


Şekil 6.47 23 Nisan 2019 DNNs

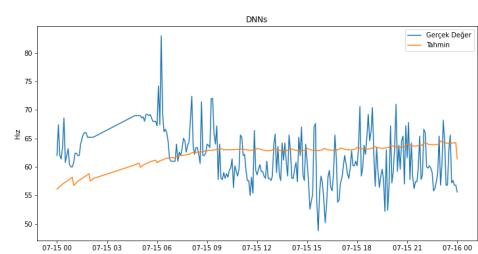


Şekil 6.48 29 Ekim 2019 DNNs

Ataşehir 316 Nolu Segment Haftasonu İle Birleşen Bayramlar



Şekil 6.49 19 Mayıs 2019 DNNs



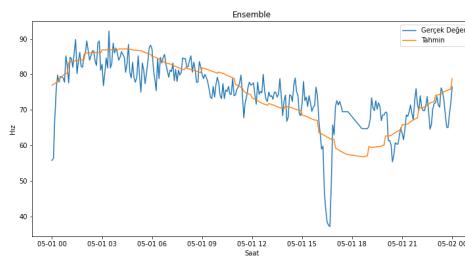
Şekil 6.50 15 Temmuz 2019 DNNs

Kadıköy Ankara Asfaltı 2102'nolu Segment Uzun Bayramlar

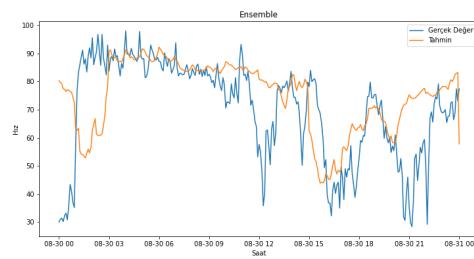
6.9 Ensemble

Kısa Bayramlar

Kısa bayram olarak sınıflandırılan 23 Nisan, 1 Mayıs, 19 Mayıs, 15 Temmuz, 30 Ağustos, 29 Ekim tarihlerindeki tatillerin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;

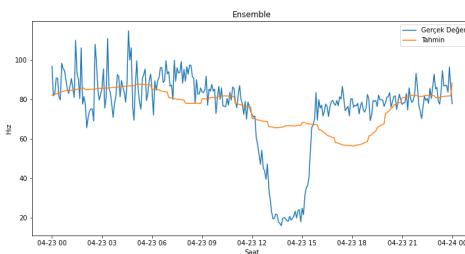


Şekil 6.51 1 Mayıs 2019 Ensemble

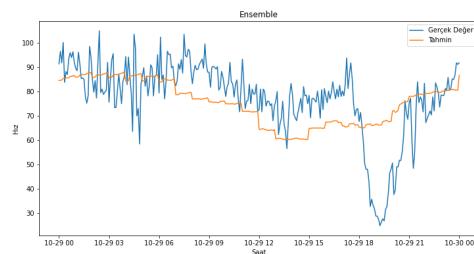


Şekil 6.52 30 Ağustos 2019 Ensemble

Avcılar 414 Nolu Segment Haftaiciine Denk Gelen Bayramlar

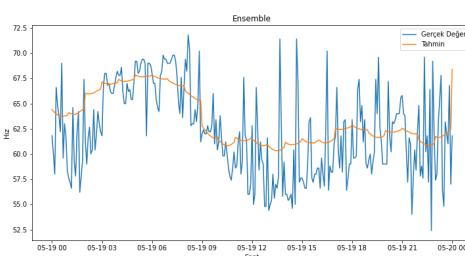


Şekil 6.53 23 Nisan 2019 Ensemble

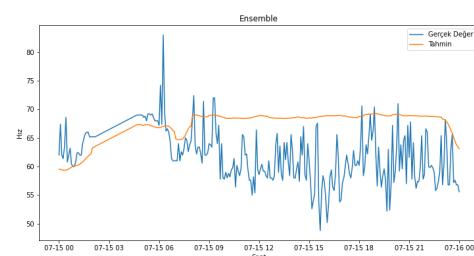


Şekil 6.54 29 Ekim 2019 Ensemble

Ataşehir 316 Nolu Segment Haftasonu İle Birleşen Bayramlar



Şekil 6.55 19 Mayıs 2019 Ensemble

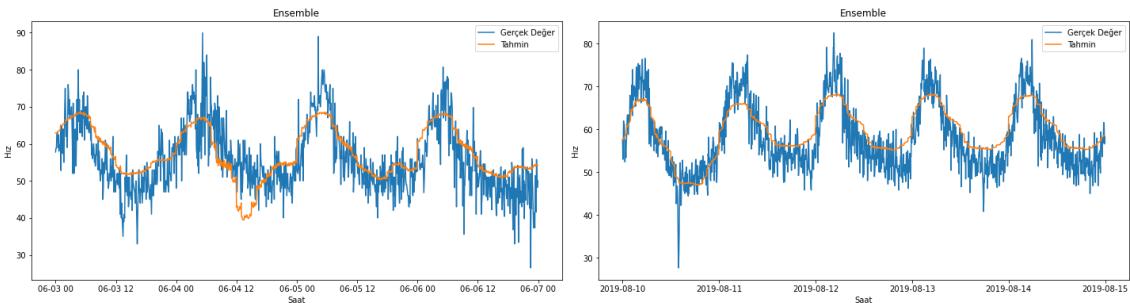


Şekil 6.56 15 Temmuz 2019 Ensemble

Farklı Karakteristik İzleyen Bayramlar

Uzun Bayramlar

Uzun bayram olarak sınıflandırılan ramazan bayramı ve kurban bayramı tatillerinin gerçek değer ve tahminleme grafikleri aşağıdaki gibidir ;



Şekil 6.57 Ramazan Bayramı 2019
Ensemble

Şekil 6.58 Kurban Bayramı 2019
Ensemble

Kadıköy Ankara Asfaltı 2102'nolu Segment Uzun Bayramlar

6.10 Hata Oranları

Hata oranları kısa ve uzun bayramlar olmak üzere ikiye ayrılmıştır. Test seti olarak kısa bayramlar için 1 günlük test seti kullanılırken uzun bayramlar için 4 ve 5 günlük test setleri kullanılmıştır. Ramazan bayramı için test seti arife ve 3 günlük bayramı içerirken kurban bayramı için arife ve 4 günlük bayramı içermektedir. Bunu yapmaktadır sebep tatil günlerinin birbirine olan benzerliği tatilden önceki veya sonraki günlerin bu benzerlige sahip olmamasıdır.

Kısa Bayramlar

Kısa bayramları 3 farklı başlık altında (Haftasonuna gelmesi, hafta içine gelerek 3 günlük bayram oluşturma, hafta içi tek günlük oluşturma) inceleme işlemlerinde her bir bayram günü ayrı ayrı değerlendirilerek set set tahminleme işlemleri yapılmıştır. Aşağıdaki tablolarda her bir bayram günü için hangi zaman aralıkları ile ne kadar hata alındığı bilgileri yer almaktadır.

Tablo 6.1 23 Nisan 2019 Hata Oranları

Set-1: Aynı Yıl Geçen Hafta

Set-2: Set1 + Geçmiş yıl milli bayramlar ve bir önceki günlerinin birleşimi

Set-3: Set1 + FFT

Set-4: Set2 + FFT

	Set-1	Set-2	Set-3	Set-4
Random Forest	15.41%	15.37%	12.24%	11.69%
GBRegressor	14.49%	13.87%	13.35%	12.31%
XGBRegressor	15.46%	15.29%	12.41%	11.69%
LGBMRegressor	14.70%	14.35%	13.27%	12.01%
Catboost	15.28%	15.10%	13.21%	11.60%
DNNs	14.24%	14.28%	13.95%	14.28%
Ensemble	14.34%	14.16%	12.19%	11.56%

Tablo 6.2 1 Mayıs 2019 Hata Oranları
 Set-1: Geçen Sene Tek Günlük Bayramlar
 Set-2: Set1 + Aynı Sene Geçen Hafta
 Set-3: Set2 + 23 Nisan 2019
 Set-4: Set3 + FFT

	Set-1	Set-2	Set-3	Set-4
Random Forest	14.00%	13.99%	10.43%	9.53%
GBRegressor	12.03%	11.83%	11.44%	10.26%
XGBRegressor	13.96%	13.88%	11.69%	10.05%
LGBMRegressor	13.07%	12.94%	11.14%	10.34%
Catboost	13.78%	13.70%	10.25%	9.63%
DNNs	11.88%	11.62%	12.76%	11.28%
Ensemble	12.68%	12.57%	10.11%	9.53%

Tablo 6.3 19 Mayıs 2019 Hata Oranları
 Set-1: Geçen Sene Tek Günlük Tatiller ve önceki bir gün
 Set-2: Aynı Yıl Bir Hafta Önce Aynı Gün
 Set-3: Set1 + FFT
 Set-4: Set2 + FFT

	Set-1	Set-2	Set-3	Set-4
Random Forest	15.12%	13.59%	15.27%	14.10%
GBRegressor	15.42%	13.27%	14.11%	13.73%
XGBRegressor	15.70%	14.07%	14.40%	13.98%
LGBMRegressor	15.13%	13.45%	13.29%	13.25%
Catboost	14.65%	13.91%	13.42%	13.73%
DNNs	20.56%	17.56%	18.19%	15.50%
Ensemble	14.72%	13.65%	14.10%	13.32%

Tablo 6.4 15 Temmuz 2019 Hata Oranları
 Set-1: Geçen Sene Tek Günlük Tatiller ve önceki bir gün
 Set-2: Set1 + FFT
 Set-3: Set1 + Aynı yıl önceki haftasonu ve önceki hafta aynı gün + FFT

	Set-1	Set-2	Set-3
Random Forest	15.20%	15.00%	13.47%
GBRegressor	13.99%	13.90%	12.90%
XGBRegressor	15.20%	15.02%	13.20%
LGBMRegressor	14.42%	14.38%	12.83%
Catboost	15.02%	14.93%	13.15%
DNNs	13.44%	13.53%	13.42%
Ensemble	14.04%	14.01%	12.76%

Tablo 6.5 30 Ağustos 2019 Hata Oranları

Set-1: Geçen Sene Tek Günlük Tatiller ve önceki bir gün + Geçen Hafta

Set-2: Set1 + FFT

Set-3: Set2 + Aynı Sene Tek Günlük Bayramlar(23 Nisan, 1 Mayıs)

	Set-1	Set-2	Set-3
Random Forest	13.43%	10.84%	10.84%
GBRegressor	10.88%	10.64%	10.64%
XGBRegressor	13.33%	9.92%	9.93%
LGBMRegressor	12.14%	11.98%	10.23%
Catboost	13.13%	12.48%	10.10%
DNNs	10.67%	10.72%	10.77%
Ensemble	11.79%	10.50%	9.87%

Tablo 6.6 29 Ekim 2019 Hata Oranları

Set-1: Geçen Sene Tek Günlük Tatiller ve önceki günleri

Set-2: Set1 + Aynı Sene Geçen Hafta

Set-3: 2 Hafta 2018 + Önceki 2 Hafta 2019

Set-4:,Set3 + FFT

	Set-1	Set-2	Set-3	Set-4
Random Forest	13.00%	13.00%	12.36%	9.89%
GBRegressor	11.36%	11.27%	12.60%	10.35%
XGBRegressor	12.98%	12.89%	13.29%	11.61%
LGBMRegressor	12.11%	11.90%	12.14%	10.55%
Catboost	12.81%	12.69%	12.27%	10.53%
DNNs	11.60%	11.28%	11.85%	11.42%
Ensemble	11.76%	11.64%	11.68%	9.75%

Uzun Bayramlar

Uzun bayramlar olan ramazan ve kurban bayramları kendine özgü davranışlar sergilemektedir. Kullanılan veri setinde 2018 ramazan bayramı haftasonuna denk gelmesi nedeni ile 3 günlük tatil olarak ilan edilmiş, 2019 senesinde ise ramazan bayramı 9 gün olması nedeni ile eğitim aşamasında 2018 ramazan bayramı günlerinin kullanılması hata sonucunu artırmaktadır. Şekil 6.7 ve Şekil 6.8 tablolarına bakıldığından ramazan bayramında oluşturulan modelin daha kötü sonuç verdiği görülebilir. Bu hata sonucunun düşürülmesi için Set-1 olarak adlandırılan eğitim setini ramazan bayramında kullanması yerine bu eğitim setine kurban bayramının tatil günlerinin de eklenmesi düşünülmüştür. Kurban bayramında ise veri seti yeterli olduğu ve birbirini takip eden senelerde aynı karakteristiği izlediği için Set-3 6.7 verisetinde Set-1 verisetine haftasonu verilerini eklemek dışında herhangi bir iyileştirmeye gidilmemiştir.

Tablo 6.7 Kurban Bayramı 2019 Hata Oranları

Set-1: Geçen Sene Kurban Bayramı 2 Hafta 13 - 27 Ağustos

Set-2: Set1 + Ramazan Bayramı

Set-3: Set1 + Aynı Sene Önceki Hafta sonu + FFT

	Set-1	Set-2	Set-3
Random Forest	11.44%	13.85%	9.14%
GBRegressor	11.44%	14.00%	8.92%
XGBRegressor	11.97%	14.92%	9.58%
LGBMRegressor	11.40%	14.45%	9.23%
Catboost	11.34%	13.75%	9.20%
DNNs	13.34%	15.38%	10.71%
Ensemble	11.28%	13.83%	8.97%

Tablo 6.8 Ramazan Bayramı 2019 Hata Oranları

Set-1: Geçen Sene Ramazan Bayramı 2 Hafta 11 - 25 Haziran

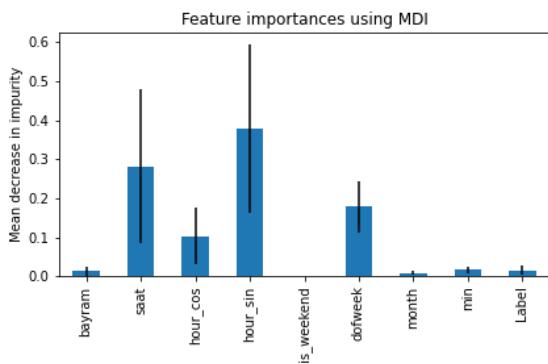
Set-2 : Set1 + Geçen Sene Haftasonu + Aynı Sene Bir Önceki Haftasonu

Set-3: Set1 + Kurban Bayramı

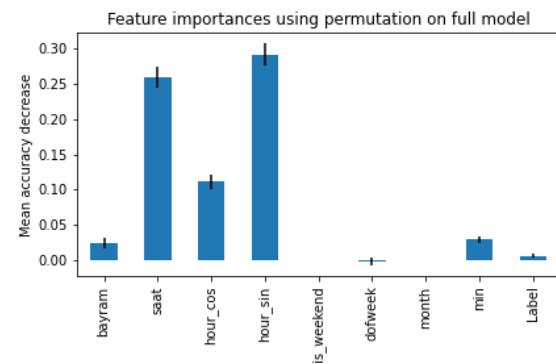
	Set-1	Set-2	Set-3
Random Forest	18.05%	15.29%	13.85%
GBRegressor	17.88%	17.01%	14.00%
XGBRegressor	19.16%	16.66%	14.92%
LGBMRegressor	18.31%	18.85%	14.45%
Catboost	17.82%	14.22%	13.75%
DNNs	19.99%	14.88%	15.38%
Ensemble	17.63%	15.61%	13.83%

6.11 Deneysel Sonuçlar

Uygulamada kısa bayramlar ve uzun bayramlar için farklı deney ortamları hazırlanmıştır. Modeli eğitirken kullanılan değişkenler kısa bayramlar için (saat, saat_cos, saat_sin, bayram, hafta_günü) olurken uzun bayramlar için (saat, saat_cos, saat_sin, bayram, hafta_günü, month, ardişik_tatil) olarak seçilmiştir. Şekil 6.59'e ve 6.60'e bakıldığında bu değişkenlerin modelin başarı oranına etkisinin grafikleştirilmiş hali hem eğitim setinde MDI kullanılarak hem de test setinde permütasyon kullanılarak hesaplanmıştır. Sadece uzun tatil günlerinde kullanılan ardişik tatil değişkeninin model performansına etkisini ise Şekil 6.61'da ve ??'da görülebilir. Aynı şekilde haftasonuna denk gelen veya haftasonuyla birleşen kısa tatil günlerinde de is_weekend değişkeni model performansında etkili olmuştur.

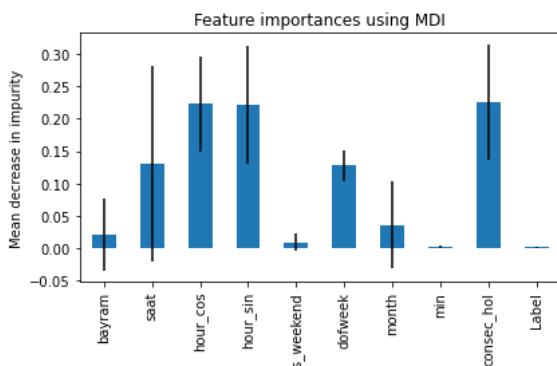


Şekil 6.59 Eğitim Setinde MDI kullanılması

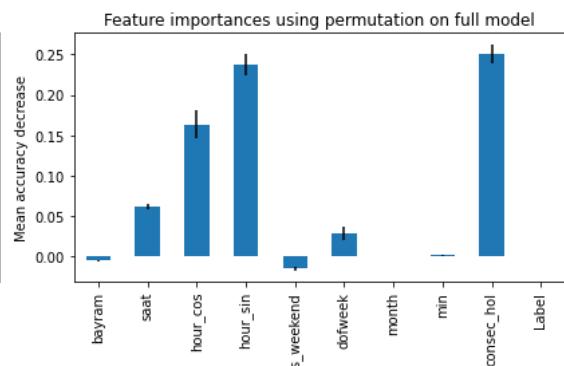


Şekil 6.60 Test Setinde Permütasyon Kullanılması

Farklı Yöntemlerle Hesaplanan Değişken Önemi



Şekil 6.61 Eğitim Setinde MDI kullanılması

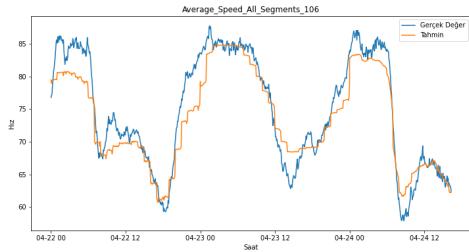


Şekil 6.62 Test Setinde Permütasyon Kullanılması

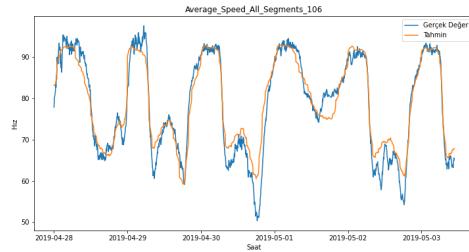
Ardışık Tatil Değişkeninin Model Performansına Etkisi

Değişken seçiminin yanısıra yapılan hız tahmininde tatil öncesi, tatil ve tatil sonrası için farklı eğitim seti yapıları oluşturulmuştur. Bu eğitim seti yapısının bir örneği Şekil 6.69'da görebilmektedir. Kısa bayramlar için bu yapı bir günlük tatillerde 1 gün öncesi

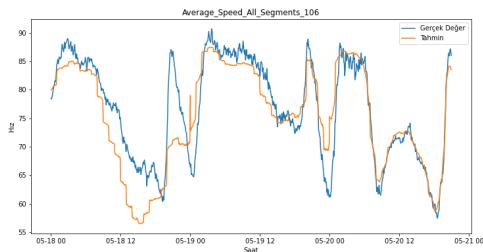
ve sonrasında kapsayan 3 günlük bir test seti oluşturulurken uzun bayramlar için bütün tatil günleriyle beraber 2 gün öncesi ve sonrasında kapsayan bir test seti oluşturulmuştur. Ayrıca tatil öncesi, tatil ve tatil sonrasında seçilen eğitim seti yapısının değişmesine ek olarak her bir model için kullanılan parametrelerde daha önce optimize edilmiş bir şekilde yapının içine dahil edilmiştir. Daha sonra bu yapı .pkl uzantılı pickle objeleri olarak kaydedilmiştir. Bu sayede her bir gün ve model için en optimum parametre ve gün seçimi sağlanmıştır.



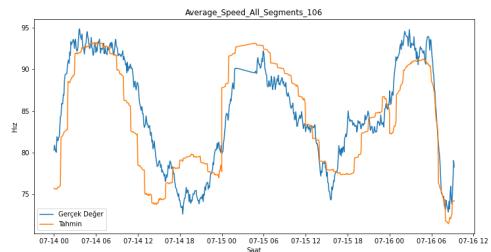
Şekil 6.63 23 Nisan 2019



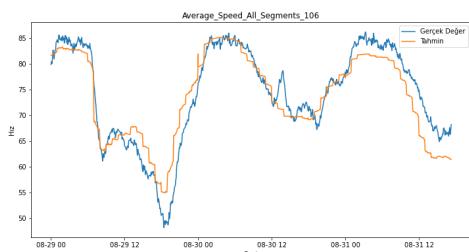
Şekil 6.64 1 Mayıs



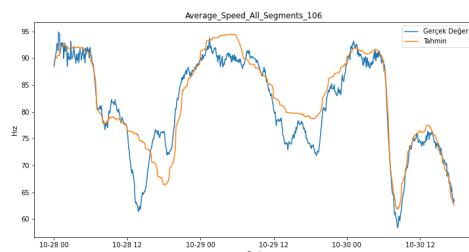
Şekil 6.65 19 Mayıs



Şekil 6.66 15 Temmuz



Şekil 6.67 30 Ağustos



Şekil 6.68 29 Ekim

Tatil Günleri İçin 106 Segment'in 'Gerçek Ve Tahmin' Hız Değer Grafikleri

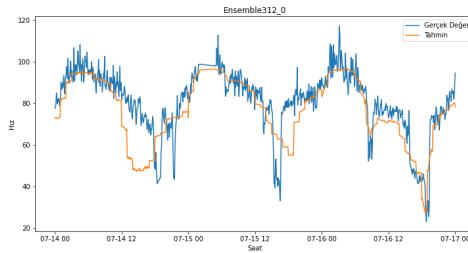
Segmentlere göre hız tahmini yapmak o segmenti kullanan araçların veya trafiği yönlendiren unsurların planlaması için iyi olsa da buna ek olarak İstanbul genelinde bir yoğunluk tahmini yapmak tatil günlerinin karakteristiklerini anlamak ve son kullanıcı için bir fikir vermesi nedeniyle önemli olmaktadır. Bu nedenle genel bir hız tahmini için yukarıdaki grafiklerde görüldüğü üzere geliştirilen model İstanbul genelinde oldukça başarılı sonuçlar vermektedir. Performans analizi kısmında hata metrikleri olarak bu sonuçlar verilmiştir.

```

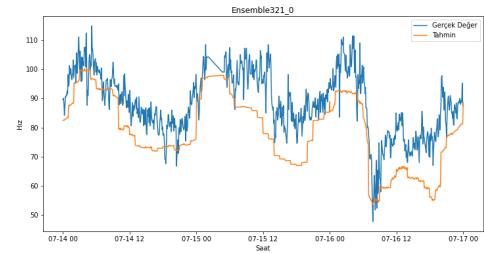
diff_day_sets = [
    pd.date_range(start="08-16-2018", end="08-16-2018 23:55", freq="5T"),
    ["05-30-2019", "05-31-2019", "07-26-2019", "06-13-2018"],
    [],
    [],
    pd.date_range(start="08-08-2019", end="08-10-2019 00:00", freq="5T")), # Tatil Öncesi
    (pd.date_range(start="08-18-2018", end="08-26-2018 23:55", freq="5T"),
    ["08-03-2019", "08-04-2019"],
    ["08-21-2018", "08-22-2018", "08-23-2018", "08-24-2018"],
    ["08-11-2019", "08-12-2019", "08-13-2019", "08-14-2019"],
    pd.date_range(start="08-10-2019", end="08-19-2019 00:00", freq="5T")), # Tatil
    (pd.date_range(start="07-22-2019", end="07-22-2019 23:55", freq="5T"), # 04-24-2019
    ["08-16-2018", "08-27-2018", "08-28-2018"],
    [],
    [],
    pd.date_range(start="08-19-2019", end="08-21-2019 00:00", freq="5T)) # Tatil Sonrası
]

```

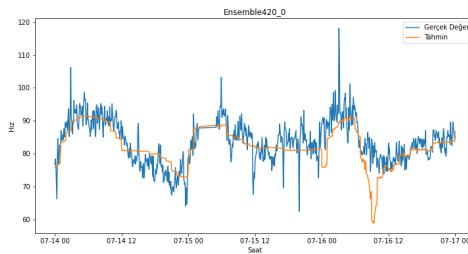
Şekil 6.69 Eğitim Seti Nesnesi



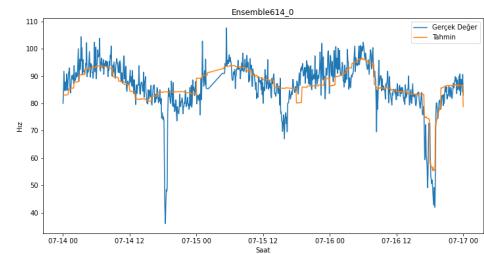
Şekil 6.70 Kadıköy (312 Numaralı Segment)



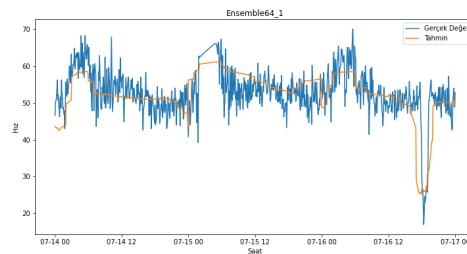
Şekil 6.71 Maltepe (321 Numaralı Segment)



Şekil 6.72 Bakırköy (420 Numaralı Segment)

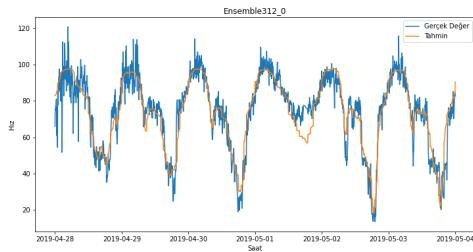


Şekil 6.73 Üsküdar (614 Numaralı Segment)

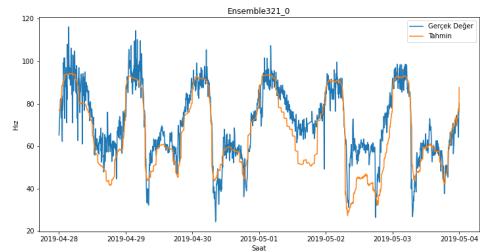


Şekil 6.74 Beşiktaş (64 Numaralı Segment)

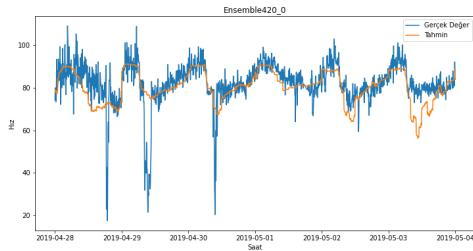
15 Temmuz 2019 Tatilinde 5 Farklı Segmentin 'Gerçek Ve Tahmin' Hız Değer Grafikleri



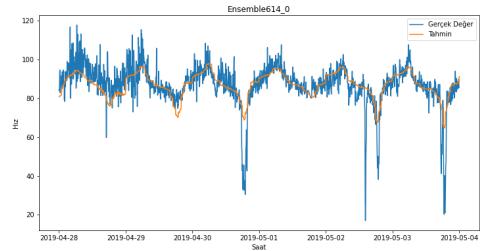
Şekil 6.75 Kadıköy (312 Numaralı Segment)



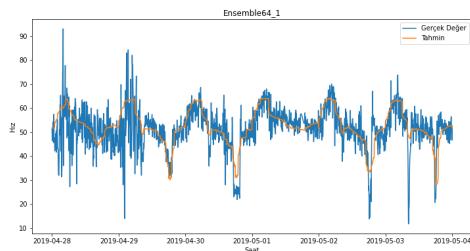
Şekil 6.76 Maltepe (321 Numaralı Segment)



Şekil 6.77 Bakırköy (420 Numaralı Segment)

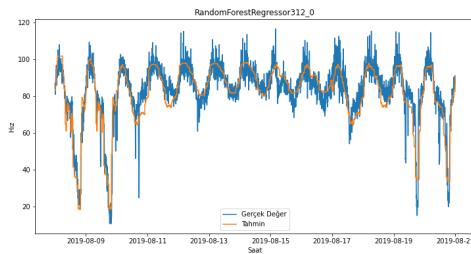


Şekil 6.78 Üsküdar (614 Numaralı Segment)

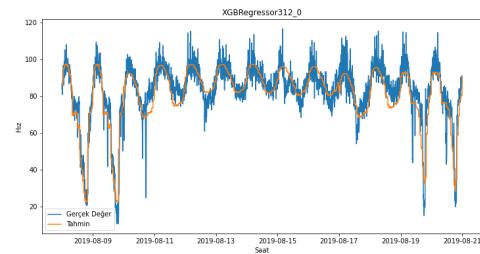


Şekil 6.79 Beşiktaş (64 Numaralı Segment)

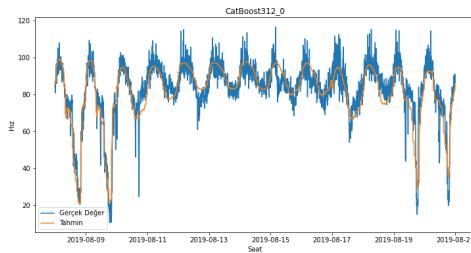
1 Mayıs 2019 Tatilinde 5 Farklı Segmentin 'Gerçek Ve Tahmin' Hız Değer Grafikleri



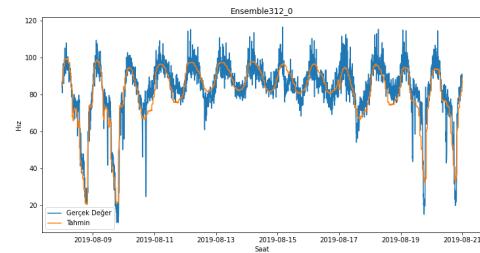
Şekil 6.80 Random Forest



Şekil 6.81 XGB Regressor

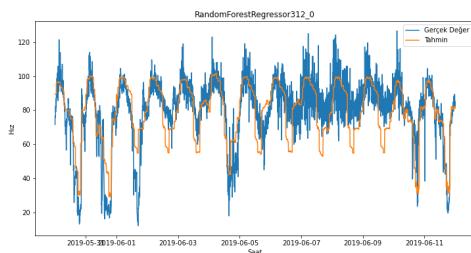


Şekil 6.82 CatBoost

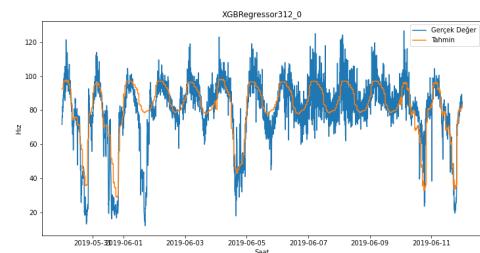


Şekil 6.83 Ensemble

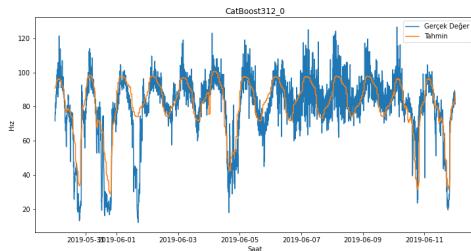
Kurban Bayramı(2019) Tatilinde Kadıköyde Farklı Modellerin 'Gerçek Ve Tahmin' Hız Değer Grafikleri



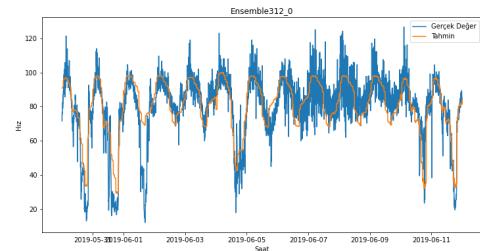
Şekil 6.84 Random Forest



Şekil 6.85 XGB Regressor



Şekil 6.86 CatBoost



Şekil 6.87 Ensemble

Ramazan Bayramı(2019) Tatilinde Kadıköyde Farklı Modellerin 'Gerçek Ve Tahmin' Hız Değer Grafikleri

6.12 Performans Analizi

Denenen farklı modeller arasında Gradient Boosting, LGBM modellerinden istenilen verim alınamadığı ve derin öğrenme modelleri için yeterli tatil verisi olmadığı için Random Forest, XGBRegressor ve Catboost modellerini FFT ön işlemesiyle birleştirilip FFT-Ensemble modeli kullanılmaya karar verilmiştir. Tablo 6.12 görüldüğü gibi farklı modeller farklı günler için iyi sonuçlar vermektedir. Bunun nedenin parametre optimizasyonu olabileceğini düşünülebilir fakat bunu en aza indirmek için her bir tatil gününe öncesi ve sonrası farklı olmak üzere parametre optimizasyonu yapılmıştır. Ortalama olarak ensemble modelimiz daha iyi sonuç verdiği görülebilir. Buna ek olarak tablo 6.9 bakıldığından genel segmentlerin karakteristiğini yansitan 106 segmentin hız tahminleri ve gerçek hız değerleri karşılaştırıldığında İstanbul genelinde ortalama %3.67 hata vermektedir. Uzun tatil günlerinde ramazan bayramının kurban bayramından daha fazla hata vermesinin nedeni 2018 yılında ramazan bayramının haftasonuna denk gelerek 3 günlük bir tatil olması aksine 2019 yılında ise haftaiçine denk gelerek 9 günlük bir tatil oluşturulmasıdır. Verideki bu problem 2019 ramazan bayramını tahmin etmede büyük sorun teşkil etmektedir. Bunu çözmek için nispeten daha uzun tatil olan 2018 kurban bayramı verisinide eğitim setine dahil ederek bu hata oranının minimuma indirilmesi denenmiştir. Tablo 6.10 örnek olarak seçilen segmentlerin hata oranları model özelinde görülmektedir. Bu segmentler seçilirken segmentlerin İstanbulun farklı bölgelerini yansıtmamasına dikkat edilmiştir. Üsküdar, Beşiktaş, Bakırköy, Kadıköy ve Maltepe seçilen segmentlerin bulunduğu illerdir. Görüldüğü üzere bu segmentlerin ikisi Avrupa yakasında diğer üçü ise Anadolu yakasında bulunmaktadır. Kadıköy ve Beşiktaş gibi nüfusun yanı sıra turistik olarak da yoğun olan bölgelerde de modelimiz başarılı sonuçlar vermektedir. Modelin eksik kaldığı nokta ise milli bayramlarda anlık olarak kutlamalardan dolayı oluşan trafik yoğunlıklarını tahmin edememesidir. Ancak bu yoğunluk sadece milli bayramlar nedenli değil aynı zamanda konser, maç, trafik kazası vb. farklı dış etkenler kaynaklı olabilir. Bu projede genel olarak tatil karakteristikleri incelenmesi nedeniyle bu etkenler gözardı edilmiştir. Bunun etkileri milli bayramlarda Kadıköy ve Beşiktaş gibi merkez ilçelerin MAPE oranlarında görülebilmektedir.

Tablo 6.9 İstanbul Geneli Segmentlerin MAPE Ortalaması

	23 Nisan	1 Mayıs	19 Mayıs	15 Temmuz	30 Ağustos	29 Ekim	Ramazan	Kurban	Ortalama
Genel MAPE	3.04%	3.47%	4.32%	3.55%	3.48%	3.41%	4.18%	3.79%	3.65%

Tablo 6.10 Örnek Segmentlere Göre MAPE Hata Oranı

Segment No	23 Nisan	1 Mayıs	19 Mayıs	15 Temmuz	30 Ağustos	29 Ekim	Ramazan	Kurban	Ortalama
614 Üsküdar	5.84%	7.18%	5.66%	5.23%	12.80%	4.95%	7.08%	4.69%	6.68%
420 Bakırköy	6.29%	8.55%	6.19%	4.44%	13.63%	9.41%	9.34%	8.28%	8.27%
312 Kadıköy	13.68%	9.40%	18.10%	11.89%	15.77%	11.32%	14.51%	7.81%	12.81%
64 Beşiktaş	7.87%	11.46%	13.36%	7.92%	11.38%	10.04%	11.68%	7.55%	10.16%
321 Maltepe	9.83%	11.40%	8.15%	11.89%	16.60%	11.41%	11.19%	11.47%	11.49%

Tablo 6.11 Farklı Modellerin 106 Segment Genelinde Ortalama MAPE Hata Oranı

Modeller	23 Nisan	1 Mayıs	19 Mayıs	15 Temmuz	30 Ağustos	29 Ekim	Ramazan	Kurban	Ortalama
RandomForestRegressor	12.26%	14.67%	12.97%	11.28%	13.61%	12.10%	13.66%	11.20%	12.72%
XGBRegressor	11.50%	16.17%	13.15%	11.36%	13.88%	12.32%	13.51%	10.70%	12.82%
CatBoost	14.07%	14.86%	12.99%	11.26%	14.04%	12.17%	13.54%	10.89%	12.98%
Ensemble	12.25%	14.97%	12.97%	11.09%	13.59%	11.99%	13.36%	10.82%	12.63%

7 Sonuç

Bu projede İstanbul sınırları içerisinde trafik hızının kısa ve uzun vadeli tahminini gerçekleştiren model geliştirilmiştir. Mevcut verilerde tatil günleri için zamana ve hızla bağlı grafikler ile görselleştirme sağlanmış, grafikler günlük ve haftalık şekilde ayrı ayrı incelenerek benzer bayramlar tespit edilmeye çalışılmış, benzerlik gösteren tatil günlerinin ortak özellikleri belirlenmiş ve gruplama işlemi gerçekleştirilmiştir. Tatil günleri olan milli bayram ve dini bayram günlerinin trafik benzerliği olarak bizim de milli-dini olarak ayıabileceğimiz veriler incelenerek belirlenmiştir. Milli bayramların cuma veya pazartesi gününe denk gelerek 3 günlük tatil oluşturması, haftanın diğer herhangi gününe denk gelmesine göre farklı karakteristik göstermesi milli bayram tatillerini de kendi içinde iki farklı gruba ayırmamız gerektiğini göstermiştir. Dini bayram tatillerinde ise tatil günlerinin hafta içine denk gelmesi ve 9 günlük tatil oluşturması diğer günlere denk gelmesine göre farklı bir karakter oluşturmuş ve kendi içerisinde ikiye ayrılmıştır. Gruplara ayırma süreci sonrasında benzerliğin yalnızca geçmiş milli veya dini bayramlar ile olabileceğiinin yanı sıra bazı segmentler için aynı yıl içerisinde yakın tarihteki günler ile de yüksek olduğu tespit edilmiştir. Tespit edilen bu segmentler incelendiğinde genel olarak aynı yılda gerçekleşen dış faktörlerin bu benzerliğe neden olduğu görülmüştür. Sadece geçen sene verisini kullanarak tahmin yapmanın yüksek hatalı sonuçlar vermesi segment bazlı ortaya çıkan yol çalışması, kaza, hava durumu, o yıla özel etkinlikler sıralanabilecek bu dış faktörlerdir. Bu benzerlikler esas alınarak her bir tatil günü için ayrı eğitim günleri bulunmuştur. Verilerin görselleştirilmesi ve yorumlanması sonucu elde edilen benzer günlerin tespiti sonrasında tahminleme için uygun modellerin tespiti yapılmış ve sonuç olarak Random Forest, CatBoost, XGBRegressor modellerine projede yer verilmiştir. Elenen modellerde göz önüne alınan faktör GBRegressor ve türevi modellerin benzer sistemlerle çalışması, Derin Öğrenme Modellerinin ise yeterli veriyile beslenememesi sonucu beklenilen sonucu verememesidir. Model seçiminde dikkat edilen nokta ise farklı modellerin kullanımının hata oranını azaltacağı yönündeki hipotezin son aşamada seçilen modeller özelinde başarılı olması, boosting yönteminin yorumlanabilirliği ve bagging yöntemlerinin çalışma sisteminin ortaya

Tablo 7.1 Önerilen Model ve Halihazırdağı diğer modellerin MAPE oranı açısından karşılaştırılması

	Tatil Günleri	Tatil Günleri	Tatil ve Öncesi Sonrası	Tatil ve Öncesi Sonrası
	Hybrid SVR-HA	FFT-Ensemble	Temel Model(HA)	FFT-Ensemble
Kısa Tatiller	12.34%	11.16%	15.78%	12.81%
Uzun Tatiller	11.05%	11.40%	14.73%	12.09%
Ortalama	11.60%	11.22%	15.52%	12.63%

çıkan hataları azaltmaktaki başarısıdır. Hız tahmini seçilen bu modellerin tahmin sonuçlarının ortalaması olarak esas alınmıştır. Seçilen model parametrelerinin tahmin edilen tatil gününe göre değişmesi gerektiği yapılan deneyler sonucunda tespit edilmiş ve bu doğrultuda her bir tatil günü için ayrı parametreler belirlenmiştir. Belirlenen bu parametrelerden kısa tatiller için sonucu en çok etkileyen parametre saat ve türevlerini kullanan parametreler olurken uzun tatiller için ardisık tatil ve haftasonunu belirten parametreler olmuştur. Bu parametrelere ek olarak günleri özniteliklerine göre gruplayan kümeleme algoritmaları denenerek yeni bir etiket özniteliği oluşturmak denenmiş bazı segmentlerde de başarı sağlanmıştır. Ancak bahsi geçen öznitelik özelinde daha detaylı bir araştırma gerektiği düşünülmektedir. Proje gerçekleştirilmesi sürecinde eksik verilerin doğru tahminlemeye olumsuz etki etmesi nedeni ile ortalama değer ile eksik verilerin doldurulması işlemi gerçekleştirilmiştir. Projenin sonraki adımlarında veri tamlığı sağlanarak ve yıl çeşitliliği artırılarak gerçege daha yakın değerlerde tahmin edilebileceği planlanmıştır. Verilen düzenlenmesi ve eksik kısımların doldurulmasından sonra yukarıda bahsedilen şekilde oluşturulan eğitim setlerine FFT uygulanmış ve verideki kirlilik kaldırılmıştır. Bu işlem sayesinde hata oranı %5.45 oranında azaltılmıştır. Eksik değerleri doldurma, eğitim seti oluşturulması, Ön işlem(FFT), öznitelik seçimi, parametre seçimi(RGSCV), ensemble adımları sonucunda ortaya çıkan model Tablo 7.1'de görülebileceği üzere tatil günlerinde Hibrit SVR-HA [1] modeline göre ortalamada %3.27'lik bir iyileştirme sağlanırken tatil günlerinin kısa tatiller için bir gün öncesi ve sonrası (3 veya 5 günlük) uzun tatiller için ise iki gün öncesi ve sonrası (13 günlük) içeren test setleri kullanılarak tahmin sonucu üretilen modelde temel modele göre %18.62'lik bir iyileştirme sağlanıyor.

Literatürde bu konuyu inceleyen modellere göre yapılan iyileştirmelere rağmen trafik tahmin probleminin hava durumu, kaza tahmini, yol bakım çalışması, segment özellikleri gibi dış faktörlerin etkisi de dahil edilerek tahmin performansı artırılabilir. Hava durumu özelinde literatürde birçok çalışma bulunmaktadır[6], [8], [9]. Bu çalışmalar incelendiğinde hava durumunun hava durumu tipi, ortalama rüzgar hızı, en düşük sıcaklık, en yüksek sıcaklık ve haftanın günü verisinin trafik tahminine etki ettiği görülmektedir. Özellikle yağmurlu havalarda ıslanan zeminin trafiğe etkisi artmaktadır ve yağmurlu havaların trafiğe etkisi ile ilgili yapılan çalışmalar ilgili

tarihin 3 hafta öncesine kadar olan verileri esas almaktadır. Fakat, yağmurlu gün sayısının sürekli olmaması ve sınırlandırıcı etki olması sistemin doğruluğunu olumsuz olarak etkileyebilmektedir. Bu nedenle hava durumu bu projede kullanılmamış yalnızca trafiğe olan etkisi yorumlanmıştır. Buna ek olarak gerçekleşen kazalar sonucu normalin dışında davranış gösteren trafik yoğunluğu da tahminleme sürecini olumsuz etkilemektedir. Kazaların rastgeleliği nedeniyle hız veya akış verisini kullanan modeller buralardaki anlık hız değişimlerini yakalamakta başarısız olmaktadır. Bu projede önerilen modelde de bu durum görülmektedir. Bu problemi ortadan kaldırmak için tarihsel kaza ve hava durumu verisini kullanarak olasılıksal kaza tahmini yapan farklı bir model geliştirilip çıkan sonuç yeni bir öznitelik olarak ana modele eklenebilir. Bu projede bu durum ele alınmamıştır. Gelecekteki çalışmalarda, sistem performansını artırmak için önceki paragrafta bahsedilen eksik verileri minimal hale getirilmesinin ve yeni bir kaza öznitelığının sisteme dahil edilmesinin yanı sıra hava durumu verisini kullanılması ve aynı lokasyon için daha geniş zamanlı veri setlerinin sisteme dahil edilmesi planlanmaktadır.

Referanslar

- [1] I. A. E. Al., “Traffic characteristics of short and long public holidays: A hybrid holiday-oriented speed prediction approach via feature engineering,” *IEEE Sensors Journal*, pp. 25 016–25 025, 2023.
- [2] X. Luo, D. Li, and S. Zhang, “Traffic flow prediction during the holidays based on dft and svr,” *Journal of Sensors*, vol. 2019, 2019.
- [3] B. Lartey, A. Homaifar, A. Girma, A. Karimoddini, and D. Opoku, “Xgboost: A tree-based approach for traffic volume prediction,” in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2021, pp. 1280–1286.
- [4] N. C. Petersen, F. Rodrigues, and F. C. Pereira, “Multi-output bus travel time prediction with convolutional lstm neural network,” *Expert Systems with Applications*, vol. 120, pp. 426–435, 2019.
- [5] Ş. Yaprak and A. Akbulut, “Trafik kaza ve denetim istatistikleri,” *Polis Akademisi Yayınları*, vol. 75, pp. 8–13, 2019.
- [6] H. Al-Selwi, A. Aziz, F. Abas, N. Hamzah, and A. Mahmud, “The impact of weather data on traffic flow prediction models,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, p. 1223, Dec. 2022. doi: 10.11591/ijai.v11.i4. pp1223-1231.
- [7] UYM. “Kurbagalidere yol çalışması.” (2019), [Online]. Available: <https://uym.ibb.gov.tr/kurumsal/haberler-ve-duyurular/kurba%C4%9Fal%C4%B1dere-%C4%B1slah-projesi-kapsam%C4%B1nda-e-5-te-%C3%A7al%C4%B1%C5%9Fma-yap%C4%B1lacak> (visited on 10/07/2019).
- [8] H. Chang, Y. Lee, B. Yoon, and S. Baek, “Dynamic near-term traffic flow prediction: System-oriented approach based on past experiences,” *IET intelligent transport systems*, vol. 6, no. 3, pp. 292–305, 2012.
- [9] P. Kar and S. Feng, “Intelligent traffic prediction by combining weather and road traffic condition information: A deep learning-based approach,” *International Journal of Intelligent Transportation Systems Research*, pp. 1–17, 2023.

Özgeçmiş

BİRİNCİ ÜYE

İsim-Soyisim: Fatih ECEVİT
Doğum Tarihi ve Yeri: 09.08.2000, Yozgat
E-mail: fatih.ecevit@std.yildiz.edu.tr
Telefon: 0553 684 41 65
Staj Tecrübeleri: YazEvi Yazılım

İKİNCİ ÜYE

İsim-Soyisim: Yusuf Mert ÇELİKARSLAN
Doğum Tarihi ve Yeri: 05.09.2000, Trabzon
E-mail: mert.celikarslan@std.yildiz.edut.tr
Telefon: 0546 760 82 52
Staj Tecrübeleri: Logo Yazılım

Proje Sistem Bilgileri

Sistem ve Yazılım: Windows İşletim Sistemi, Python
Gerekli RAM: 16GB
Gerekli Disk: 20GB