İSTANBUL TEKNİK ÜNİVERSİTESİ
Bilgisayar ve Bilişim Fakültesi

UNSUPERVISED LEARNING CLUSTRING (KUMELEME) ALGORITMALARI KULLANARAK ANOMALY DETECTION PYTHON UYGULAMASI

STAJ

AHMET ZAFER SAĞLIK
150160519

16.06/13.07.2020

İstanbul Teknik Üniversitesi

Bilgisayar ve Bilişim Fakültesi

STAJ RAPORU

Akademik Yıl: 2020/2021

Staj yapılan dönem: ☐Yaz ✓ Bahar ☐Güz

Öğrenci ile ilgili bilgiler

Adı ve Soyadı:	Ahmet Zafer SAĞLIK
Öğrenci Numarası:	150160519
Bölüm:	Bilgisayar Mühendisliği
Program:	Bilgisayar Mühendisliği (%30 İngilizce)
E-posta Adresi:	saglik16@itu.edu.tr
(Cep) Tel No:	0 5536842824

ÇAP öğrencisi misiniz? ☐ Evet (ÇAP yapağınız Fakülte/Bölüm:
✓ Hayır

Mezuniyet durumunda mısınız?	✓ Evet ☐ Hayır
Yaz okulunda ders alıyor musunuz?	☐Evet (Ders sayısı: __) ✓ Hayır

Öğrencinin çalıştığı kurum ile ilgili bilgiler

İsmi: Vakıfbank Ebis Genel Müdürlüğü
Birimi: Uygulama Geliştirme
Web Adresi: https:www.vakifbank.com.tr/
Kısa Adresi: Saray Mahallesi, Dr. Adnan Büyükdeniz Cd No:20, 34768 Ümraniye/ İstanbul

Yetkili kişi ile ilgili bilgiler

.

Bölümü:	Uygulama Geliştirme
Unvanı:	Teknik Müdür
Adı ve Soyadı:	Şefik BAYCAN
(Kurumsal) E-posta:	Sefik.BAYCAN@vakifbank.com.tr
(Kurumsal) Tel. No.:	+90-216-7241000

Yapılan iş ile ilgili bilgiler

Staj yeri	✓Türkiye ☐Yurtdışı
Staj başlangıç tarihi	16.06.2020
Staj bilş tarihi	13.07.2020
Stajda çalışılan net **gün** sayısı	20
Staj süresince sigortanız var mıydı?	☐ Evet, İTÜ tarafından sigortalandım. ✓Evet, kurum taraundan sigortalandım. ☐Hayır, yurtdışı stajı yaptım. ☐Hayır.

İÇİNDEKİLER

1.KURULUŞ HAKKINDA BİLGİLER

Vakıfbank 1954 yılında kurulan ve Türkiye de hem çalışanı desteklerken hemde ekonomiye katkıda bulunan lider bir bankadır. Kurumsal, ticari, KOBİ ve tarım yanı sıra bireysel ve kurumsal hizmetlerde bulunurken, yanlızca bir alanda değil bir çok alanda hizmet veren önde gelen bir bankadır.

İç ve dış ticaret sektöründe öncü olan Vaıfbank temel bankacılığın yanında bir katılım ve yatırım bankasıdır. Ayrıca finansal yardımcılarıyla faktoring ve Arge faliyetleriyle sektörün ileri görüşlü kurumları arasındadır.

Ülke çapından 900 den fazla şubesi olan Vakıfbank, bir çok teknolojik alt yapı hizmetini yanında kullanmaktadır.KKTC'de Tasfiye Halinde World Vakıf UBB. Ltd. ve Kıbrıs Vakıflar Bank. Ltd. olmak üzere yurt dışında üç bankada da iştiraki bulunmaktadır. VakıfBank'ın diğer iştirakleri arasında; Vakıf Faktoring A.Ş., Vakıf Finansal Kiralama A.Ş. , ABD New York ve Kuzey Irak Erbil şubelerinin yanı sıra bir de Bahreyn'de kıyı bankacılığı şubesi vardır. Ayrıca, Avusturya'da VakıfBank International AG (Viyana Şubesi ve Almanya'da Köln Şubesi), , Vakıf Gayrimenkul Yatırım Ortaklığı A.Ş., Vakıf Menkul Kıymet Yat. Ort. A.Ş., Vakıf Yatırım Menkul Değerler A.Ş. Vakıf Pazarlama San. ve Ticaret A.Ş., Taksim Otelcilik A.Ş., Vakıf Enerji ve Madencilik A.Ş., Vakıf Gayrimenkul Değerleme A.Ş. bulunmaktadır.[1]

2. GİRİŞ

13.01.2020 – 07.02.2020 tarihleri arasında 20 iş günü olarak çalıştığım stajımı Vakıfbank Ebis Genel Müdürlüğü'n de tamamladım ve bu raporu yapmış olduğum
staj hakkında tarafınızın bilgi edinmesi amacıyla hazırlamış bulunmaktayım. Stajımı online olarak tamamladım.Zaten şirkette hali hazırda part time olarak çalıştığım Vakıfbank Ebis Genel Müdürlüğü'nde gerçekleştirdim.Staj boyunca ekip liderim ve teknik müdürüm Şefik BAYCAN'ın direktifleri doğrultusunda Banka datasına direk erişimin sağlandı. Bu sayede sql query leriyle hazır tablodan direk istediğim veri setini indirebiliyordum. Ekip arkadaşarımda sürec boyunca her türlü sorumu yanıtlayıp yardımcı oldular.

Bu staj bana mezuniyetten sonra iş hayatı hakkında öngörümün oluşmasını sağladı. Proje süreçlerinin neler olduğu, yoğun iş hayatını, donanımlı ekip olmak gerektiğini, disiplinili iş hayatını staj boyunca gözlemledim. Bu doğrultuda kariyerim için elimden geldiğince kodlamada ve proje analizinde pratik yaptım, gelecek şartlarına kendimi hazırlamaya çalıştım. Okul sonrası hayatımda kurumsal firmada çalışma konusunda motive oldum.

3. STAJ PROJESİNİN TANIMI VE ANALİZİ

Stajda ilk olarak Projeyi yapabilmek için gerekli olan araçları belirleyerek başlangıç yaptım ve bu araçların kullanımı hakkında gerekli araştırmaları tamamladım. Proje için yazılım geliştirme dili olarak python ve geliştirme ortamı olarak Google Colap notebook online python editorunu kullandım. Banka bilgisayarlarından gerekli izinleri almak zor olduğu ve Jupiter Notebook tarzı bir ide'ye ihtiyacım olduğundan Colab çok işime yaradı.

Amacım olan Unsupervised Anomaly Detection için öncelikle Ekip Müdürüm olan Şefik BAYCAN ile algoritma tasarımı yaptık.Çeşitli algoritma örnekleriyle anomaly detectionın temellerini öğrenip Sql ile data setimi hazırladım ve çalışmaya başladım.[2]

3.1 Projeyi Geliştirmek İçin Programların kurulması

Projenin geliştirileceği kütüphaneler olan ; numpy, pandas, matplotlib, sklearn, hmmlearn ve scipy Google Colab notebook'a kurdum ve gerekli konfigürasyonları yaptım. Kodumu rahat bir şekilde tüm outputları görebileğimiz şekilde yazdım.Dataset olarak bankamızın aylık ve haftalık Eft transfer verisini içeriren datayı Sql den çektim ve bu seti Colab a entegre ettim.

3.2 Uygulamanın Amacı, Kapsamı ve İşleyişi

3.2.1 Uygulamanın Amacı ve Kapsamı

Uygulamanın amacı, hali hazırda izleme biriminde çalıştığımızdan bankanını tüm datası bize gelmektedir. Bu data üzerinden sql ile statik thresholdlar konulmuş ve bu thresholdlar vasıtasıyla alarm mekanizması oluştutulmuştur. Ancak yaklaşık 1000 kadar izleme birimi olduğundan dolayı bu yöntem hem zaman hem kod hemde banka personeli açısından çok pahalı (costly) olmaktadır.

Bu sorunu çözmek için bir makine öğrenme mekanizması uygulanmalıydı. Amacım varolan data trendini tespit edip datamı bu trend ile train edip yeni gelen datanın outlier olup olmadığına bakmak olmalıydı. Ancak datamda labellar olmadığından ve geçmişteki her alarm (statik thresholdu geçme) gerçek alarm olmadığından dolayı (true-false değerler) labellama işlemi de çok masraflı olucaktı.Bu nedenle yöntem olarak Unsupervised Anomaly Detection algoritmalarını kullandım. Bu yapılar Clustiring (Kümeleme) tekniğini kullanmaktaydı.

3.2.2 İşleyişi

Çalışmama öncelikle Oracle PLSQL Developer uygulmasından çeşitli Sql sorgulamaları ve optimum constraintlerle Dataseti mi çekerek başladım. İndirdiğim data set haftalık ve aylık zaman dilimlerindeydi. 3 tane sutuna sabitti,bunlar metricid, tarih ve measure sutunlarıydır. Metricid kullanılan izleyicinin numarası idi.Tarih sutunu Python Pandas Date fonksiyonuna benzer bir yapıdaydır.Veriler her bir dakikaya ait ve en güncel veri son bir dakikaya aittir. Figür 1 de kullandığım data setten bir örnek gösterilmiştir.

TARIH	METRİC	MEASURE
06/08/2020 10:39:00	68	63
06/08/2020 10:40:00	68	59
06/08/2020 10:55:00	68	58
06/08/2020 10:56:00	68	56
06/08/2020 11:11:00	68	53
06/09/2020 11:45:00	68	46
06/09/2020 15:09:00	68	35
06/09/2020 16:43:00	68	44
06/09/2020 17:00:00	68	18
06/09/2020 16:10:00	68	45
06/09/2020 16:28:00	68	47
06/09/2020 16:44:00	68	42
06/09/2020 16:29:00	68	37
06/09/2020 16:45:00	68	48

Figür 1

Data seti gerekli şekilde Sql den çektikten sonra pythona entegre etmek için Pandas'ın "read_csv" fonksiyonunu kullandım. Bu sayede Measure Ve Tarıh kolonlarına Numpy dizisi şeklinde erişebildim. Kodum aşağıdaki Figür 2 de görüldüğü gibidir.

```
# Import the necessary packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.mixture import GaussianMixture
from hmmlearn import hmm
import numpy as np

plt.style.use('ggplot')
data = pd.read_csv("result.csv")
data['MEASURE']  # as a Series
array = data['MEASURE'].values  # as a numpy array
measure=np.asarray(array)
print(measure.shape)
data['TARIH']  # as a Series
array2 = data['TARIH'].values  # as a numpy array
tarih=np.asarray(array2)
print(tarih.shape)
```

Figür 2

Tarih ve Measure sutunları arasında bir Map fonksiyonu kullanmam gerekliydi. Bu nedenle bende dataframe oluşturmayı tercih ettim. Bunun içinde yine Pandasın Dataframe fonksiyonun kullandım.Kodum Figür 3 te görüldüğü gibidir.

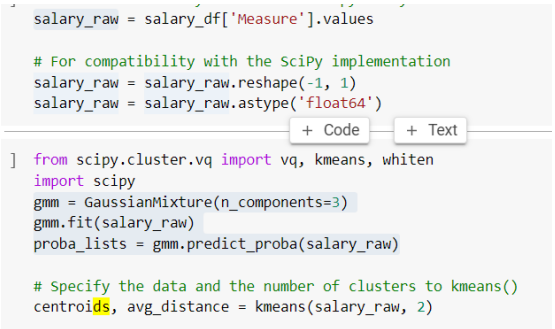
```
# Create pandas DataFrame
salary_df = pd.DataFrame(
    {'Tarih': tarih,
     'Measure': measure
    })

# Print a subsection of the DataFrame
print(salary_df.head())
```

Figür 3

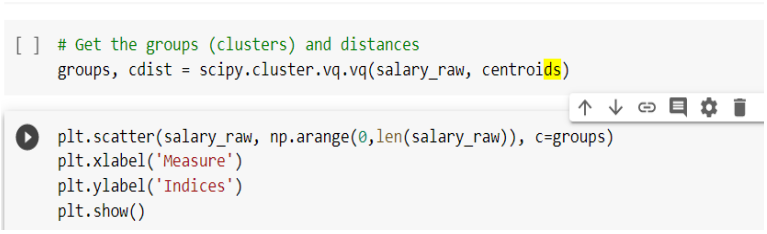
Datamı hazırladıktan sonra optimum Cluster (Kümeleme) yöntemini bulmak için testlere başladım. Bunun için ilk denediğim yöntem K-means Clustering yöntemiydi.

Clustiring yöntemlerinde Benzer türdeki nesneleri gruplayarak başlarız. Matematiksel olarak, bu benzerlik Öklid mesafesi, Manhattan mesafesi ve benzeri gibi mesafe ölçüm fonksiyonları ile ölçülür. Bende kullandığım yöntemlerde Öklid mesafesini kullandım. Ardından kmeans modülünü scipy.cluster.vq dosyasından içe aktardım. (SciPy, Scientific Python anlamına gelir ve bilimsel deneyler yapmak için çeşitli kullanışlı araçlar sağlar.) Uyguladığım kod Figür 4 te görüldüğü gibidir.Cluster yapısında amacım Outlier (Anormal) leri bulmak olduğundan yanlızca 2 küme kullandım.



Figür 4

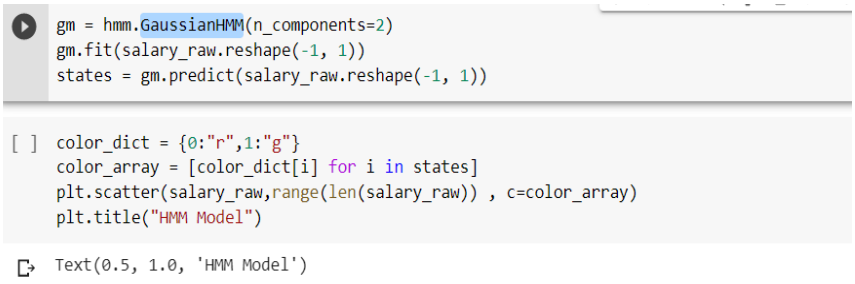
Yukarıdaki kod yığınında, measure verilerini(=salary_df) kmeans () ile besledik. Ayrıca veri noktalarını gruplandırmak istediğimiz küme sayısını da belirledik. Centroidler kmeans() tarafından üretilen centroidlerdir ve Avg_distance, beslenen veri noktaları ile kmeans () tarafından üretilen centroidler arasındaki ortalama Öklid mesafesidir.[3] Ardından Figür 5 te belirttiğim kodla datayı ve grupları görselleştirdim. (Figürler Sonuçlar ve Karşılaştırma bölümünde gösterilecektir.)



Figür 5

İkinci yöntem olarak Hidden Markov Clustring algoritmasını kullandım. Hidden Markov Modeli (HMM), modellenen sistemin gözlemlenmemiş (yani gizli) durumlara sahip bir Markov süreci olduğu varsayılan istatistiksel bir Markov modelidir. Hidden Markov modelleri özellikle konuşma, el yazısı, jest tanıma, konuşma parçası etiketleme, müzikal skor takibi, kısmi deşarj ve biyoinformatik gibi takviye öğrenme ve zamansal örüntü tanıma uygulamalarıyla bilinir.

Bu yöntem için kullandığım temel kod Figür 6 da görüldüğü gibidir.[4]



Figür 6

Data seti bir kere hazırladığımdan ve kütüphaneleri import ettiğimden dolayı küçük kod dizileriyle direk algoritmaları uygulayabiliyordum.Ardından Figür 6'da belirttiğim kodla datayı ve grupları görselleştirdim.

(Figürler Sonuçlar ve Karşılaştırma bölümünde gösterilecektir.)

Son kullandığım yöntem **Hierarchical Agglomerative Clustering** yöntemiydi.Hiyerarşik kümeleme algoritmaları benzer nesneleri küme adı verilen gruplara ayırır. İki tür hiyerarşik kümeleme algoritması vardır:[5]

Topak - Aşağıdan yukarıya yaklaşım. Birçok küçük kümeyle başlayın ve daha büyük kümeler oluşturmak için bunları birleştirilir.

Bölücü - Yukarıdan aşağıya yaklaşım. Küçük kümelere ayırmak yerine tek bir kümeyle başlanır.

Method olarak ben single distance ı kullandım.Bu methodda iki küme arasındaki mesafe, her kümedeki iki nokta arasındaki en kısa mesafe şeklinde hesaplanmaktadır.Algoritmanın temel mantığı birbirini en yakın iki kümeyi alıp onları tekbir küme yapmaktır.Ardından Figüre 7de belirttiğim kodla datayı ve grupları görselleştirdim.
(Figürler Sonuçlar ve Karşılaştırma bölümünde gösterilecektir.)

```
[ ] model = AgglomerativeClustering(n_clusters=2, affinity='euclidean', linkage='single')
    model.fit(salary_raw)
    labels = model.labels_
```

```
[ ] plt.scatter(salary_raw, np.arange(0,len(salary_raw)), c=labels)
    plt.xlabel('Measure')
    plt.ylabel('Indices')
    plt.show()
```

Figür 7

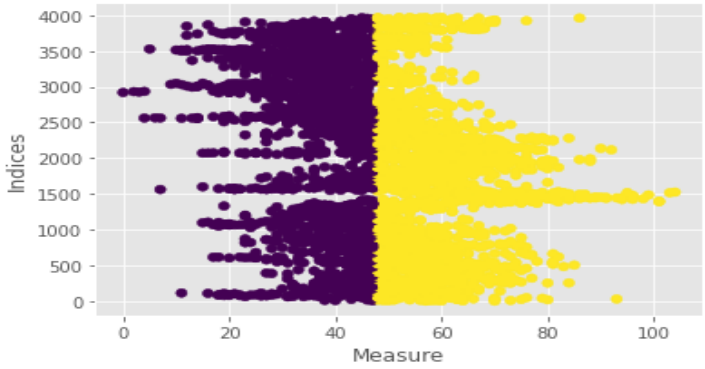
Su ana kadar gösterdiğim 3 kümeleme algoritması(**K-means,Hidden Markov Model** ve **Hierarchical Agglomerative Clustering** en iyi sonucları veren yöntemlerdir.Bunun dışında test sırasında kullandığım algoritmalar rapora dahil edilmemiştir.

3.2.3 Sonuçlar ve Karşılaştırma

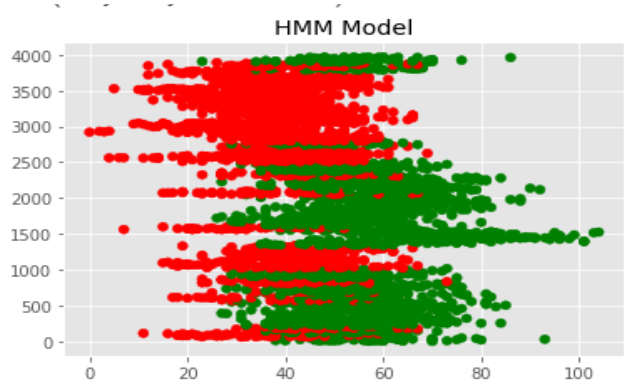
Test seti olarak iki ayrı veriyi kullandım.Birincisi bankanın aylık datası ikinci veri ise haftalık datasıydı.Sonuclar aşağıdaki matplot çizimlerinde gösterildiği gibidir.

Haftalık Data

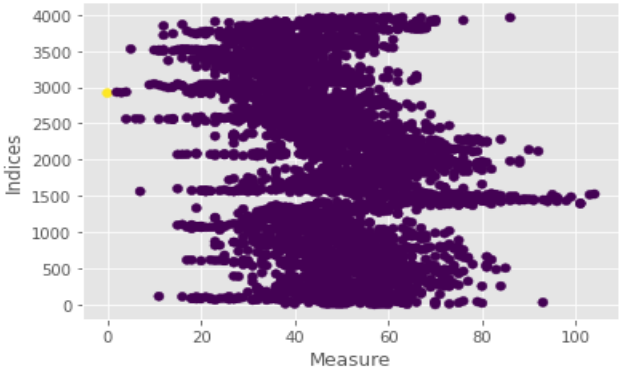
K-means



Hidden Markov Model

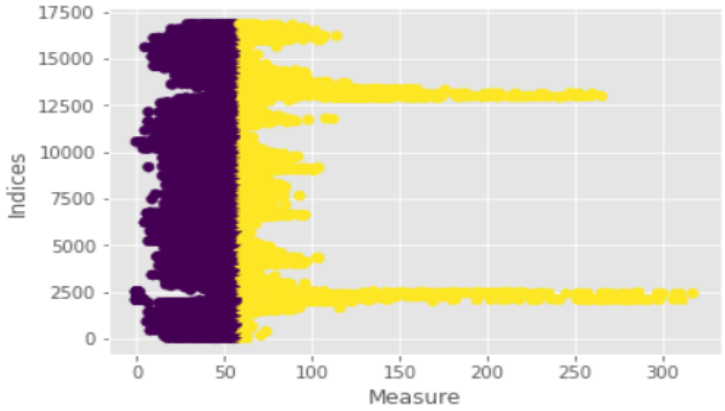


Hierarchical Agglomerative Clustering

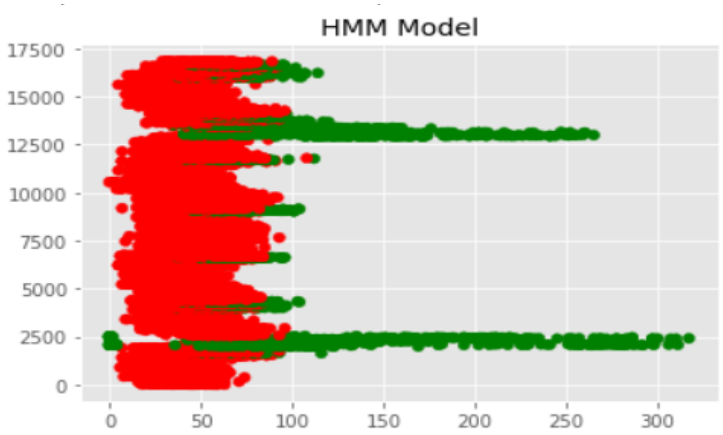


Aylık Data

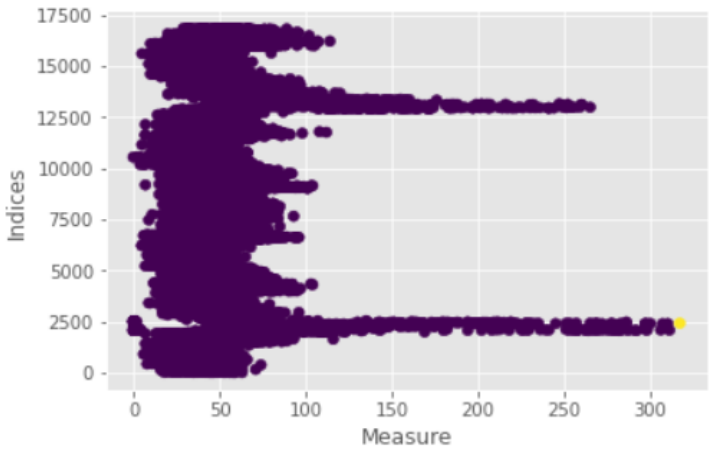
K-means



Hidden Markov Model



Hierarchical Agglomerative Clustering



Haftalık ve aylık sonuçlarda gözlemlenen renkler kümeleri temsil etmektedir. Azınlıkta olan küme anomalliğin olduğu bölgedir. Bizim buradaki amacımızı trendi tam olarak tespit etmek ve anormalliği minimal düzeyde doğru olarak tespit etmektir. Bu amaç ile sonuçlara baktığımızda **Hierarchical Agglomerative Clustering** methodunun haftalık ve aylık setlerde anormalliği son derece düzgün bulduğunu gözlemlemekteyiz. Buradaki asıl sebep hiyerarşik kümeleme yaparken kullandığımız **Simple leakage** methodudur. Haftalık veride sadece bir tane sıfır geldiğinden anormallik sıfır üzerine odaklanmış.Ancak aylık verideki birkaç hafta üst üste gelen sıfır nedeniyle bu anormallik ortadan kalkmıştır.[6]

Yıllık veri kullandığımızda bu methotla tam doğru sonuca ulaşabileceğimizi düşünüyorum. **Hierarchical Agglomerative Clustering** time series verilerde başarısını gözlemlemiş oldum. Staj sonrasındaki dönemde de bu analizi şirket databasine uygulayarak şirketi büyük bir yükten kurtarmayı hedefliyorum.

4.SONUÇ

Staj sürecinde bir yazılım geliştirmenin yanında iş hayatına ve kurum yapısına dair birçok bilgi edindim. Yazılım mühendisliğinin canlı uygulamalarına şahit olup içerisinde yaşananlar ve ihtiyaçları konusunda bilgi sahibi oldum. Kod yazarken, geliştirme araçlarını ve servislerini kullanırken yardıma ihtiyacım olduğunda tecrübeli mühendislere danışabiliyor olmak bu süreçte işimi önemli miktarda kolaylaştırdı. Yazılım mühendisliğinin temel prensipleri hakkında teorik ve uygulamalı bilgi sahbi oldum. İleride de Data Scientist olmak ve Machine Learning algoritmalarıyla uğraşmak istediğimden bu staj benim için çok verimli oldu.Staj sürecinde yapılan diğer literatür araştırmalarıyla pratiğin yanında teorik bilgiler elde ettim.

Staj yaptığım şirkette yeni insanlarla tanışma fırsatım oldu ve bu insanların ileriki hayatımda benim bir yerlere gelmeme ve sektör hakkında derin tecrübelere sahip olmama katkı sağlayacağını düşünüyorum. Data izlemetajyer olmamın benim için en uygun tercih olduğunu anladım.

5.REFERANSLAR

[1] https://www.vakifbank.com.tr

[2] https://medium.com/learningdatascience/anomaly-detection-techniques-in-python-50f650c75aaf

[3] https://blog.floydhub.com/introduction-to-anomaly-detection-in-python/

[4] https://medium.com/@kangeugine/hidden-markov-model-7681c22f5b9

[5] https://towardsdatascience.com/machine-learning-algorithms-part-12-hierarchical-agglomerative-clustering-example-in-python-1e18e0075019

[6] https://nlp.stanford.edu/IR-book/completelink.html