



Python Kullanarak Ön Değerlendirme Platformunda İşveren ve Aday Davranışlarının Veri Analizi

Yazılım Mühendisliği Ana Bilim Dalı

Yüksek Lisans / Bitirme Projesi

Eda Yıldız

Y210240063

Proje Danışmanı: Prof. Dr. Ayşegül Alaybeyoğlu

Haziran 2023

İzmir Kâtip Çelebi Üniversitesi Fen Bilimleri Enstitüsü öğrencisi **Eda Yıldız** tarafından hazırlanan **Python Kullanarak Ön Değerlendirme Platformunda İşveren ve Aday Davranışlarının Veri Analizi** başlıklı bu çalışma tarafımızca okunmuş olup, yapılan savunma sınavı sonucunda kapsam ve nitelik açısından başarılı bulunarak jürimiz tarafından YÜKSEK LİSANS / BİTİRME PROJESİ olarak kabul edilmiştir.

ONAYLAYANLAR:

Proje Danışmanı:

Prof. Dr. Ayşegül Alaybeyoğlu
İzmir Kâtip Çelebi Üniversitesi

Yazarlık Beyanı

Ben, **Eda Yıldız**, başlığı **Python Kullanarak Ön Değerlendirme Platformunda İşveren ve Aday Davranışlarının Veri Analizi** olan bu projenin ve projenin içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu tezin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Tezin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Kayda değer yardım aldığım bütün kaynaklara teşekkür ettim.
- Tezde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

Tarih: 16.06.2023

Python Kullanarak Ön Değerlendirme Platformunda İşveren ve Aday Davranışlarının Veri Analizi

ÖZ

Bu proje, bir ön değerlendirme platformunda işveren ve aday davranışlarını analiz etmeyi amaçlamaktadır. Bu analiz, adayların işverenlerle olan etkileşimlerini ve platformdaki performansını inceleyen kapsamlı bir veri analizi üzerine kuruludur. Proje, işverenlerin ve adayların davranışlarını anlama ve bu bilgiyi kullanarak aday başarısını tahmin etme potansiyelini araştırır. Bu analiz, işverenlerin katılımının ve adayların performansının nasıl optimize edilebileceğine dair anlayışlar sağlamayı amaçlar. Veriler, Mixpanel'dan ve veri tabanından toplanır ve Python programlama dili ve özellikle Pandas, NumPy ve Matplotlib gibi Python kütüphaneleri kullanılarak analiz edilir. Ayrıca, veri analizi sonuçları, makine öğrenimi tekniklerinin aday başarısını tahmin etmede nasıl kullanılabileceğine dair fikirler sağlar. Bu proje, işe alım sürecini daha etkili ve verimli hale getirmek için veri analizi ve makine öğrenimini nasıl kullanabileceğimizi anlama çabalarını ilerletmeyi hedefler.

Anahtar Sözcükler: Veri Analizi, Python, Makine Öğrenimi, İşe Alım Süreci, İşveren ve Aday Davranışları

Data Analysis of Employer and Candidate Behavior on a Pre-Assessment Platform Using Python

Abstract

This project endeavors to analyze the behavior of employers and candidates on a pre-evaluation platform, employing a comprehensive scrutiny of data. The analysis revolves around candidates' interactions with employers and their performance on the platform. The primary objective of this project is to gain insights into employer and candidate behavior, utilizing this understanding to predict candidate success. The analysis aims to provide valuable insights for optimizing employer participation and enhancing candidate performance. Data is gathered from Mixpanel and a database, and it is subsequently analyzed using the Python programming language, specifically utilizing libraries such as Pandas, NumPy, and Matplotlib. Moreover, the outcomes of the data analysis provide valuable insights into how machine learning techniques can be employed to forecast candidate success. Ultimately, this project strives to contribute to our understanding of leveraging data analysis and machine learning to improve the effectiveness and efficiency of the recruitment process.

Keywords: Data Analysis, Python, Machine Learning, Recruitment Process, Employer and Candidate Behavior

Teşekkür

Projenin gerekli verileri Coensio şirketi tarafından sağlanmıştır. Çalışmanın çıkış noktası olmalarıyla birlikte gerekli verileri sağladıklarından ve çalışmaya olan katkılarından dolayı Coensio'ya ve ekibine teşekkürlerimi iletiyorum.

İçindekiler

Yazarlık Beyanı.....	2
Öz.....	3
Abstract.....	4
Teşekkür.....	5
İçindekiler.....	6
Şekiller Listesi.....	7
1. Giriş.....	8
2. Literatür Araştırması.....	9
3. Veri Hazırlama.....	10
3.1 Veri Toplama.....	10
3.2 Veri Ön İşleme.....	11
4. Veri Analizi.....	17
5. Sonuç.....	21
Kaynaklar.....	23
Ekler.....	24
Ek A.....	25
Özgeçmiş.....	26

Şekiller Listesi

Şekil 3.2.1 Candidate View Company Based veri setinin son örnek çıktısı..... 15

Şekil 3.2.2 Candidate Application Company Based veri setinin son örnek çıktısı... 15

Şekil 4.1 View vs Application Scatter Plot..... 18

Şekil 4.2 View vs Application regresyon sonucu 20

1. Giriş

Günümüzde işe alım süreçlerinin daha etkin ve verimli hale getirilmesi, işverenler ve iş arayan adaylar için kritik bir öneme sahiptir. Bu süreçlerin daha bilgilendirici ve aydınlatıcı olabilmesi için genellikle çeşitli ön değerlendirme platformları kullanılır. Bu tür platformlar, işverenlerin ve adayların etkileşimlerini ve performanslarını kaydetmek ve analiz etmek için bir fırsat sağlar. Bu analizler, işverenlerin adayları daha iyi anlamalarına, adayların performanslarını optimize etmelerine ve genel olarak işe alım süreçlerinin etkinliğini artırmalarına yardımcı olabilir (Nikolaou, 2020).

Bu projede, bir ön değerlendirme platformunda işveren ve aday davranışlarını Python programlama dili kullanarak analiz etmeyi amaçlıyoruz. Veri analizinde kullanılan veriler, platformda gerçekleşen etkileşimlerin kaydını tutan Mixpanel ve bir veritabanından toplanmaktadır. Analiz sürecinde, Python'un güçlü kütüphanelerinden yararlanılmaktadır: Pandas veri manipülasyonu ve analizi için, NumPy sayısal hesaplamalar için ve Matplotlib de verinin görselleştirilmesi için kullanılacaktır (McKinney, 2012; Harris et al., 2020; Hunter, 2007).

Veritabanında ve Mixpanel'da kaydedilen bilgiler, kullanıcı kimliği, eylem türü, gerçekleştirilen eylem zamanı, eylem sırasında kaydedilen veri değeri ve eylemin gerçekleştirildiği cihaz türü gibi bilgileri içerir. Bu veriler, kullanıcıların (işverenler ve adaylar) platformdaki etkileşimlerini ve davranışlarını daha iyi anlamak ve analiz etmek için kritik öneme sahip olabilir.

Veri analizi süreci, aşağıdaki sorulara yanıt arayarak aday ve işveren davranışlarına geniş bir bakış açısı sunar: Adayların sınav performansı nasıl? Hangi müşteriler en sık kullanıcı hesabı oluşturuyorlar? Hangi reklam kanalları en yüksek dönüşüm oranlarına sahip? Müşteriler hangi özellikleri en çok kullanıyorlar? Hangi sınav soruları en zor veya en kolay? İşverenler hangi özellikleri en çok kullanıyorlar? Hangi işverenler en sık iş ilanı yayınlıyorlar? İşverenler hangi adayları daha fazla incelemeye alıyorlar? Hangi işverenler en fazla aday ile iletişime geçiyorlar? İşverenlerin hangi ilanlarına en fazla başvuru yapılıyor? (Chen et al., 2012)

Bunlar ve daha fazlası, proje boyunca yanıtlanması hedeflenen bazı kritik sorulardır. Veri analizi sonuçları, aday başarısını tahmin etmek için makine öğrenme tekniklerinin uygulanabileceği bir yol gösterebilir (Breiman, 2001). Bu, işveren katılımını ve aday performansını platformda nasıl optimize edebileceğimize dair daha derin bilgiler sağlayacaktır.

2. Literatür Araştırması

Bu bölümde, veri analizi, ön değerlendirme platformları, makine öğrenmesi ve reklam kanallarının dönüşüm oranları gibi projemizin merkezinde yer alan kavramları ve terimleri kapsayan mevcut literatürü ayrıntılı bir şekilde inceliyoruz. Bu temel terimler ve kavramlar, projemizin temel amacı olan aday ve işveren davranışlarını analiz etme ve işe alım süreçlerini optimize etme hedefleriyle doğrudan ilişkilidir.

Veri Analizi: Veri analizi, çok sayıda verinin incelenmesi ve yorumlanması sürecidir. İşe alım süreçlerinde veri analizi, adayların yeteneklerini ve becerilerini daha objektif ve ayrıntılı bir şekilde anlama imkanı sağlar. Python programlama dili, bu süreci gerçekleştirmek için yaygın olarak kullanılır. Özellikle, Pandas, NumPy ve Matplotlib gibi Python kütüphaneleri, veriyi etkin ve hızlı bir şekilde işlemek ve görselleştirmek için kullanılır (McKinney, 2012; Harris et al., 2020; Hunter, 2007). Bu kütüphaneler, karmaşık veri yapılarını ve işlemlerini kolaylıkla yönetmek için esneklik ve çeşitlilik sunar. Bu, analistlerin daha derinlemesine analizler yapmasını ve daha bilgilendirici sonuçlar çıkarmasını sağlar.

Ön Değerlendirme Platformları: Ön değerlendirme platformları, işe alım süreçlerinin daha verimli ve etkin hale getirilmesinde önemli bir rol oynar. İşverenlerle adaylar arasındaki etkileşimleri izleyerek, işverenlere adayların beceri ve yeteneklerini daha doğru bir şekilde değerlendirme imkanı sağlarlar (Nikolaou, 2020). Bu platformlar, işe alım sürecinde adil ve şeffaf bir ortam sağlar ve aynı zamanda işverenlerin en uygun adayları bulmasını kolaylaştırır. Bu, işe alım süreçlerinin daha verimli hale gelmesine yardımcı olur.

Makine Öğrenmesi: Makine öğrenmesi, yapay zekanın bir alt dalıdır ve bilgisayarların belirli bir görevi daha iyi bir şekilde yerine getirmek için öğrenme yeteneğini ifade eder. İşe alım süreçlerinde, makine öğrenmesi modelleri aday başarısını tahmin etmek için kullanılır. Bu modeller, verileri analiz eder ve adayların başarısını etkileyen faktörleri belirlemeye yardımcı

olur (Breiman, 2001; Hastie et al., 2009). Bu, işverenlerin en uygun adayları belirlemesine yardımcı olur.

Reklam Kanallarının Dönüşüm Oranları: Dönüşüm oranları, bir reklam kampanyasının etkinliğini ölçmek için kullanılır. İşe alım süreçlerinde, dönüşüm oranları, belirli bir reklam kampanyasının kaç adayı platforma çektiğini gösterir. Bu, işverenlerin hangi reklam kampanyalarının en etkili olduğunu belirlemesine ve aday çekme stratejilerini geliştirmesine yardımcı olur (Lewis and Rao, 2015).

Bu literatür araştırması, bu projenin kavramsal çerçevesinin anlaşılmasına yardımcı olur ve aynı zamanda veri analizinin işe alım süreçlerini nasıl optimize edebileceğini ve makine öğrenmesinin aday başarısını nasıl tahmin edebileceğini daha iyi anlamamıza yardımcı olur. Bu bilgi, bu projenin daha geniş amaçlarına ulaşılmasında kritik öneme sahiptir.

3. Veri Hazırlama

Veri toplama ve veri ön işleme süreçleri genellikle "Veri Hazırlama" süreci altında ele alınır. Bu süreç, genellikle bir veri bilimi veya makine öğrenmesi projesinin önemli bir bölümünü oluşturur. Veri toplama, bu sürecin ilk aşamasını temsil ederken, veri ön işleme, toplanan verinin işlenebilir ve analiz için uygun hale getirildiği ikinci aşamadır.

1. 3.1 Veri Toplama

Veri toplama, herhangi bir analitik projenin temel taşlarından biridir. Bu süreç, araştırma hedeflerine uygun olarak tasarlanmış yöntemler kullanarak kaliteli, güvenilir ve işlenebilir verilerin toplanmasını içerir. Veri toplama sürecinin etkinliği, projenin genel başarısında kritik bir rol oynar, çünkü analizlerin geçerliliği ve güvenilirliği doğrudan toplanan verinin kalitesine bağlıdır.

Veri toplama yöntemleri genellikle iki genel kategoriye ayrılır: Birincil ve ikincil veri toplama. Birincil veri toplama, doğrudan kaynaktan orijinal verilerin toplanmasını içerir. Bu süreç genellikle anketler, deneyler veya gözlemler yoluyla gerçekleştirilir. İkincil veri toplama ise, daha önce toplanmış verilerin analiz edilmesini içerir. Bu genellikle literatür incelemeleri, veri

tabanlarından veri çekme veya daha önce yapılan arařtırmaları kullanma řeklinde gerekleřtirilir.

Bu projede, hem birincil hem de ikincil veri toplama teknikleri kullanılmıřtır. İkincil veri toplama, mevcut veritabanından ve Mixpanel'dan veri çekme yoluyla gerekleřtirilmiřtir. Veritabanı ve Mixpanel, kullanıcıların platformdaki etkinliklerini kaydeder, böylece kullanıcı davranıřları ve etkileřimleri hakkında kapsamlı bilgiler saęlar. Birincil veri toplama ise, Python programlama dili kullanılarak gerekleřtirilen veri manipölasyonu ve analizler yoluyla gerekleřtirilmiřtir.

Veri toplama sürecinde, dikkat edilmesi gereken bazı kritik noktalar vardır. İlk olarak, toplanan verinin projenin hedeflerine ve analiz gereksinimlerine uygun olması gerekmektedir. Bu proje için, belirlenen arařtırma sorularına yanıt verebilecek bilgileri içeren verilerin toplanması önemlidir. İkinci olarak, verinin doęruluęu ve tutarlılıęı, analizin güvenilirlięini ve geçerlilięini doęrudan etkiledięi için hayati önem tařır. Bu yüzden veri toplama sürecinde herhangi bir hata veya tutarsızlıęı tespit etmek ve düzeltmek önemlidir.

Son olarak, veri toplama sürecinde, verilerin analiz edilmesini kolaylařtırmak için çeřitli adımlar atılmıřtır. Veri manipölasyonu ve temizleme teknikleri kullanılarak, veri setinin analiz için hazırlanması ve optimize edilmesi saęlanmıřtır. Bu süreç, verinin uygun bir formatta olduęundan ve analiz için kullanılabilir olduęundan emin olmak için gerekleřtirilmiřtir. Bu adımlar, verinin daha etkili bir řekilde analiz edilmesini ve projenin genel hedeflerine ulařılmasını saęlamaktadır.

2. 3.2 Veri Ön İřleme

Veri Ön İřleme, ham verinin makine öęrenmesi veya veri analizi gibi sonraki iřlemler için hazır hale getirilmesi sürecidir. Bu süreç, genellikle verinin anlaşılır, etkin ve güvenilir hale getirilmesi için veri temizleme, veri dönüřtürme ve veri indirgeme adımlarını içerir. Veri ön iřleme adımları arasında eksik verilerin doldurulması, aykırı deęerlerin iřlenmesi, gereksiz özelliklerin kaldırılması ve verinin uygun formata dönüřtürülmesi bulunabilir.

Veri ön iřleme ařaması, veri analizi veya makine öęrenmesi projelerinde kritik öneme sahiptir. Temiz, düzenli ve tutarlı veri setleri, analiz sürecini kolaylařtırır ve sonuçların doęruluęunu

artırır. Ayrıca, veri ön işleme, verinin işlenmesi ve analiz edilmesi sürecinde daha hızlı ve daha verimli hale getirir.

Bu projede, veri ön işlememiz, Python kütüphaneleri Pandas ve NumPy'nin işlevselliğini kullanarak gerçekleştirildi. İlk olarak, eksik veriler belirlendi ve bu veriler uygun bir şekilde dolduruldu veya kaldırıldı. Ardından, aykırı değerler belirlendi ve bu değerler, genel veri setinin bütünlüğünü bozmayacak şekilde işlendi. Gereksiz özellikler, analiz için önemli olmayan veya yanıltıcı sonuçlara yol açabilecek bilgiler belirlendi ve veri setinden çıkarıldı. Son olarak, veri uygun bir formata dönüştürüldü, böylece makine öğrenmesi algoritmaları tarafından daha kolay işlenebilir hale geldi.

Veri ön işleme aşamasında dikkat edilmesi gereken en önemli noktalar, verinin bütünlüğünün korunması ve analiz sürecinin kalitesinin artırılmasıdır. Veriyi işlerken, her adımda, verinin doğruluğunu ve tutarlılığını etkileyebilecek potansiyel sorunları dikkatlice değerlendirmek önemlidir. Ayrıca, verinin işlenmesi ve analiz edilmesi sürecini kolaylaştıracak araç ve tekniklerin seçilmesi de önemlidir. Bu projede, veri ön işleme sürecimiz, verinin tutarlılığını koruma ve analiz sürecini optimize etme hedeflerine dayanmaktadır.

Veri Ön işleme ortamı için Google Colaboratory notebook kullanılmıştır¹. Aşağıdaki kütüphaneler eklenerek Python dili ile ön işleme yapılmıştır.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
```

Bu projede incelenmek üzere 2 ana veri setine karar verilmiştir: “Candidate View Company Based” ve “Candidate Application Company Based”.

Candidate View Company Based veri seti; Date, Event, Company Name, Oct 1 2022, 12:00AM - May 23 2023, 11:43PM sütunlarından oluşmaktadır².

Date: Değerlerin kaydedildiği tarihi belirtmektedir.

¹ Notebook dosyasının linkine EK A.1'dan ulaşılabilir.

² Candidate View Company Based veri seti linkine EK A.2'den ulaşılabilir.

Event: Hangi etkinliğin verileri olduğunu belirtiyor Dataset event'i; Her bir firmanın ilgili zaman aralığında unique adaya bakma sayısını vermektedir.

Company Name: Firma isimlerini belirtiyor. Veri ön işleme sürecinde KVKK gereği firma isimleri başta bulanık veya çizili olacaktır. Datasetler birleştirildikten sonra her bir firma ismi Company1, Company2 şeklinde sıralanıp analiz edilecektir.

Oct 1 2022, 12:00AM - May 23 2023, 11:43PM: Verinin çekildiği zaman aralığını vermektedir. Altında event'in değerleri yer almaktadır.

Bu veri setinde sırayla aşağıdaki veri ön işleme işlemleri yapılmıştır:

1. Sütun isimlerinde karmaşık ve uzun olanlar yeniden oluşturulmuştur.
2. Kullanılacak veriler seçilmiş ve ihtiyaç olmayan sütunlar silinmiştir.

Yukarıdaki maddeler için örnek kodlara aşağıdan ulaşabilirsiniz.

```
# Oct 1 2022, 12:00AM - May 23 2023, 11:43PM sütununun adını
Countolarak güncelle
candidate_view = candidate_view.rename(columns={'Oct 1 2022, 12:00AM -
May 23 2023, 11:43PM': 'Count'})

# Company Name sütununun adını Company olarak güncelle
candidate_view = candidate_view.rename(columns={'Company Name':
'Company'})

# Event sütununu sil
candidate_view = candidate_view.drop('Event', axis=1)
```

3. Karmaşık veriler daha okunur hale getirilmiştir.

```
# Benzersiz tarih değerlerini al
dates = candidate_view['Date'].unique()

# Yeni bir DataFrame oluştur
```

```

new_df = pd.DataFrame(columns=['Company'] + list(dates))

# Her bir şirket için verileri doldur
for company in candidate_view['Company'].unique():
    company_data = candidate_view[candidate_view['Company'] == company]
    count_values = company_data['Count'].tolist()
    new_row = [company] + count_values
    new_df = new_df.append(pd.Series(new_row, index=new_df.columns),
ignore_index=True)

# DataFrame'i görüntüle
print(new_df)

```

4. Tarihsel ayrıma ihtiyacımız olmadığı için Total Count alıp yeni bir sütun yazdırılmıştır ve 10 değerinin altındaki satırlar silinmiştir. Çünkü analiz etmek için yeterli veriyi oluşturmamaktadır.

```

# Tarih sütunlarının isimlerini belirtin
tarih_sutunlari = ['2022-10', '2023-01', '2023-04']

# Toplamı hesaplamak için tarih sütunlarını toplayın
new_df[View] = new_df[tarih_sutunlari].sum(axis=1)

new_df = new_df[new_df[View] >= 10]

```

5. Tarihsel sütunlar silinmiş ve yalnızca Company Name ve Total sütunları bırakılmıştır. Veri setinin düzenlenmiş son hali Şekil 3.2'de görüldüğü gibidir.

```

new_df.drop(['2022-10', '2023-01', '2023-04'], axis=1, inplace=True)
new_df = new_df[['Company', View]]
last1 = new_df.copy()
last1.head()

```

	Company	View
1		50.0
4		422.0
5		24.0
6		1832.0
7		657.0

Şekil 3.2.1: Candidate View Company Based veri setinin son örnek çıktısı

Candidate Application Company Based veri seti; Company, Application sütunlarından oluşmaktadır³.

Company: Firma isimlerini belirtmektedir.

Application: İlgili firmaya Oct 1 2022, 12:00AM - May 23 2023, 11:43PM zaman aralığında gelen başvuru sayılarını belirtmektedir. Bu veri setinde sırayla aşağıdaki veri ön işleme işlemleri yapılmıştır:

1. Veri seti diğeriyle birleştirilmeye uygun gözüktüğü için herhangi bir işleme yapılmamıştır. Örnek çıktı Şekil 3.2.2'deki gibidir:

	Company	Application
0		212
1		5131
2		305
3		11
4		1

³ Candidate Application Company Based veri seti linkine EK A.3'den ulaşılabilir.

Şekil 3.2.2: Candidate Application Company Based veri setinin son örnek çıktısı

Veri setlerinin ayrı ayrı ön işleme bu adımda tamamlanmıştır. Bundan sonraki işlemler veri setlerini birleştirme ve işleme üzerinedir.

1. Veri setleri Company'ye göre birleştirilmiştir ve diğer satırlar silinmiştir. Çünkü 2 verisi olmayan firmalar için ilişkilerini sağlıklı şekilde inceleyemeyiz.

```
merged_df = pd.merge(last1, last2, on='Company', how='inner')
```

2. Firma isimleri Company1, Company2 şeklinde sırayla düzenlenmiştir.

```
# Öncelikle, veri setinde bulunan eşsiz şirket adlarını alın.
unique_companies = merged_df['Company'].unique()

# Şimdi, her birine 'Company1', 'Company2', 'Company3'... gibi bir isim
atayalım.
new_company_names = {company: f'Company{i+1}' for i, company in
enumerate(unique_companies)}

# 'Company' sütununu yeniden adlandırma işlemi
merged_df['Company'] = merged_df['Company'].map(new_company_names)
```

3. Analiz edilecek verinin son çıktısı Şekil 3.2.3'deki gibidir. 74 satırdan ve 3 sütundan oluşmaktadır.

	Company	View	Application
0	Company1	50.0	201
1	Company2	422.0	5131
2	Company3	24.0	5280
3	Company4	1832.0	8372
4	Company5	657.0	8078
..
69	Company70	379.0	4033
70	Company71	18.0	106
71	Company72	297.0	1451
72	Company73	131.0	7388
73	Company74	23.0	48

Şekil 3.2.3: merged_df veri setinin son örnek çıktısı

4. Veri Analizi

Veri analizi, toplanan verilerin anlamlı sonuçlar çıkarabileceğimiz bir formata getirilmesi ve incelenmesi sürecidir. Veri analizi genellikle veri temizleme, dönüşüm, modelleme ve sonuçların yorumlanması aşamalarını içerir.

Veri analizi, karar verme sürecinde ve özellikle akademik araştırmaların sonuçları üzerinde büyük bir etkisi olan karmaşık bilgileri anlaşılır ve uygulanabilir bilgilere dönüştürme yeteneğini sağlar. Dolayısıyla, belirli bir araştırma veya inceleme çerçevesinde önemli bir rol oynar.

Bu çalışmada, veri analizi, toplanan veri setinin incelenmesi ve anlamlı bilgilerin çıkarılması için kullanıldı. Öncelikle, verilerin genel özelliklerini anlamak ve veri setinin temel özelliklerini belirlemek için açıklayıcı veri analizi (EDA) kullanıldı. Bu analiz, veri setimizin genel dağılımını, merkezi eğilimlerini ve dağılımını incelemek için kullanıldı.

Ardından, 'View' ve 'Application' değişkenleri arasındaki ilişkiyi incelemek için regresyon analizi yapıldı. Bu analiz, bir 'View' artışının, 'Application' üzerinde ne kadarlık bir etkisi olduğunu belirlemek için kullanıldı.

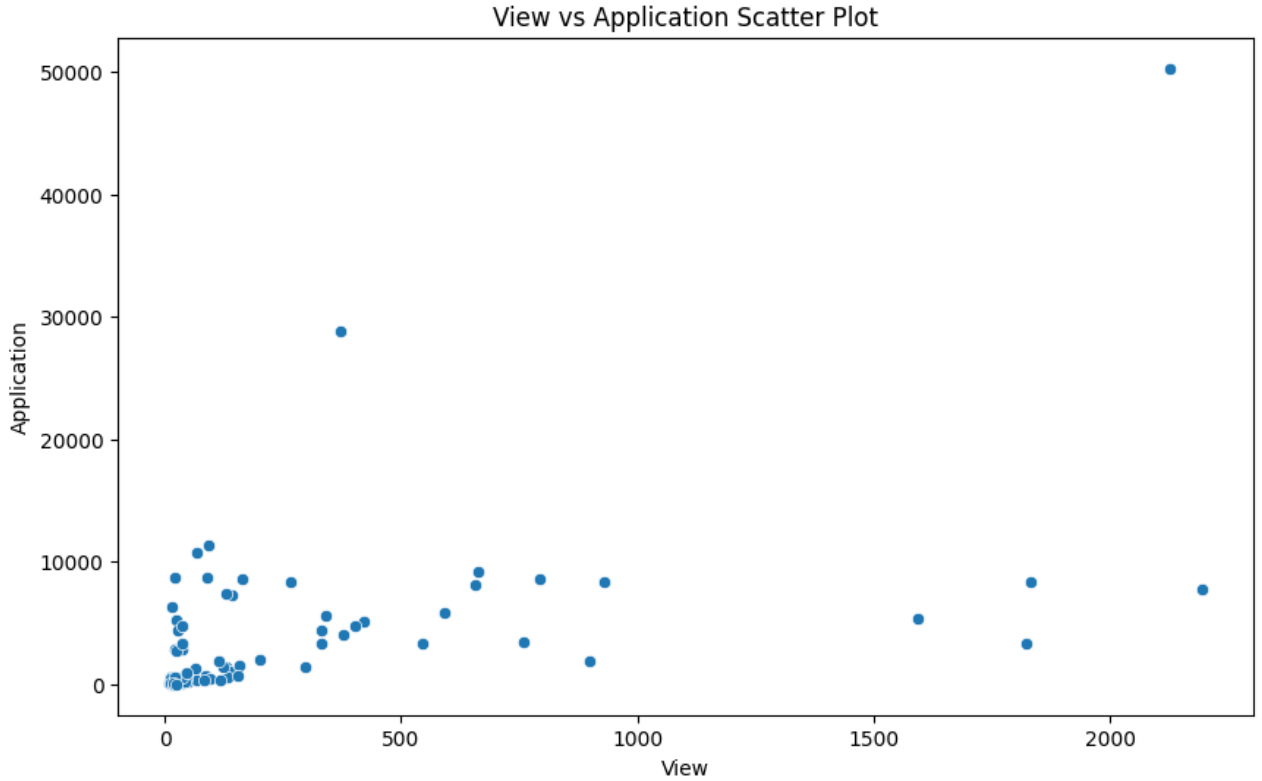
İlk olarak, 'View' ve 'Application' sütunları arasındaki ilişkiyi anlamak için bir scatter plot (dağılım grafiği) oluşturabiliriz.

```
# Scatter plot oluşturma
plt.figure(figsize=(10,6))
sns.scatterplot(data=merged_df, x='View', y='Application')

# X ve Y eksenlerini adlandırma
plt.xlabel('View')
plt.ylabel('Application')
plt.title('View vs Application Scatter Plot')

plt.show()
```

Bu scatter plot, 'View' ve 'Application' arasındaki ilişkiyi Şekil 4.1'deki gibi görselleştirir. Eğer noktalar bir çizgi etrafında toplanıyorsa, bu iki değişken arasında güçlü bir ilişki olduğunu gösterir.



Şekil 4.1: View vs Application Scatter Plot

Bu tabloya göre bazı uç değerler olduğunu görüyoruz. Ancak noktaların belirli bir çizgi etrafında toplandığını söyleyebiliriz. Bu yüzden aralarında anlamlı bir ilişki olduğu yönünde incelemeye devam edebiliriz.

Daha sonra, bu iki değişken arasındaki korelasyonu hesaplayabiliriz. Korelasyon, iki değişken arasındaki ilişkinin gücünü ve yönünü ölçer.

```
# Korelasyon hesaplama
correlation = merged_df['View'].corr(merged_df['Application'])

print(f"View and Application correlation: {correlation}")
```

Bu kod, 'View' ve 'Application' arasındaki korelasyonu hesaplar. Korelasyon katsayısı -1 ile +1 arasında değerler alır. +1, mükemmel pozitif korelasyonu; 0, korelasyonun olmadığını; -1 ise mükemmel negatif korelasyonu temsil eder.

Veri setine göre View ve Application arasında çıkan korelasyon: 0.521213559882364
0.5 korelasyona göre pozitif yönde orta kuvvetli bir ilişki olduğu hakkında yorumlayabiliriz.

Son olarak, 'View' ve 'Application' arasındaki ilişkiyi daha da iyi anlamak için bir regresyon analizi yapabiliriz.

```
# Regresyon analizi
import statsmodels.api as sm

# Bağımlı ve bağımsız değişkenleri belirle
X = merged_df['View']
y = merged_df['Application']

# Regresyon modeli oluşturma
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()

# Regresyon sonuçlarını yazdırma
print(model.summary())
```

Bu kod, 'View' (bağımsız değişken) ve 'Application' (bağımlı değişken) arasındaki regresyon modelini oluşturur ve sonuçları yazdırır. Regresyon sonuçlarından, modelin uygunluğunu, katsayıları ve diğer istatistiksel bilgileri alabiliriz. Bu, 'View' ve 'Application' arasındaki ilişkiyi daha detaylı bir şekilde anlamamızı sağlar.

Verilerimiz için çıkan regresyon sonucunu 4.2'deki gibi inceleyebiliriz.

```

=====
                        OLS Regression Results
=====
Dep. Variable:          Application    R-squared:                0.272
Model:                 OLS           Adj. R-squared:           0.262
Method:               Least Squares  F-statistic:              26.86
Date:                 Wed, 24 May 2023  Prob (F-statistic):       1.92e-06
Time:                 19:14:00        Log-Likelihood:          -747.79
No. Observations:    74             AIC:                     1500.
Df Residuals:        72             BIC:                     1504.
Df Model:             1
Covariance Type:     nonrobust
=====
                        coef      std err          t      P>|t|     [0.025     0.975]
-----
const          1987.2010    812.518        2.446    0.017    367.476    3606.926
View            7.3347         1.415         5.182    0.000     4.513     10.156
=====
Omnibus:                 76.382    Durbin-Watson:           2.022
Prob(Omnibus):           0.000    Jarque-Bera (JB):       714.684
Skew:                    3.054    Prob(JB):                6.43e-156
Kurtosis:                16.945    Cond. No.                668.
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```

Şekil 4.2: View vs Application regresyon sonucu

R-squared ve Adj. R-squared: Bu değerler, modelin verideki varyansın ne kadarını açıkladığını gösterir. R-squared ve Adjusted R-squared değerleri sırasıyla %27.2 ve %26.2'dir. Yani, model, 'Application' varyansının sadece %27.2'sini açıklıyor, bu genellikle düşük kabul edilir. Ancak modelin kalitesini yalnızca bu istatistiklere dayanarak değerlendirmemeliyiz.

F-statistic ve Prob (F-statistic): F istatistiği, modelin anlamlılığını test eder. F istatistiği 26.86'dır ve p-değeri (Prob (F-statistic)) 1.92e-06'dır. Bu, modelin anlamlı olduğunu, yani 'View' değişkeninin 'Application' üzerinde önemli bir etkisi olduğunu gösterir.

Katsayılar: 'View' değişkeninin katsayısı 7.3347'dir. Bu, 'View' değerinde bir birimlik artışın, 'Application' değerinde 7.3347 birimlik bir artışa neden olduğunu gösterir.

Regresyon analizi sonucunda, 'View' deęişkeninin katsayısının 7.3347 olduęu bulundu. Bu, 'View' deęerinde bir birimlik artışın, 'Application' deęerinde 7.3347 birimlik bir artışa neden olduęunu gösterir. R-squared deęeri, modelin 'Application' varyansının %27.2'sini açıkladıęını gösterir.

Bu veri analizi, 'View' ve 'Application' deęişkenleri arasında anlamlı bir ilişki olduęunu göstermiştir. Ancak, modelin sadece 'Application' deęişkenindeki varyansın %27.2'sini açıklayabildięi belirlendi. Bu, dięer deęişkenlerin de 'Application' üzerinde önemli bir etkisi olabileceęini gösterir. Bu nedenle, bu veri setinin daha fazla analizi, belki de dięer deęişkenlerin dahil edilmesiyle, 'Application' üzerindeki etkileri daha iyi anlamamızı sağlayabilir.

Bu veri analizi, 'View' ve 'Application' deęişkenleri arasında anlamlı bir ilişki olduęunu göstermiştir. Ancak, modelin sadece 'Application' deęişkenindeki varyansın %27.2'sini açıklayabildięi belirlendi. Bu, dięer deęişkenlerin de 'Application' üzerinde önemli bir etkisi olabileceęini gösterir. Bu nedenle, bu veri setinin daha fazla analizi, belki de dięer deęişkenlerin dahil edilmesiyle, 'Application' üzerindeki etkileri daha iyi anlamamızı sağlayabilir.

5. Sonuç

Bu araştırma, firmaların ön deęerlendirme platformunu kullanarak adayları inceleme süreçleri (burada 'View' olarak adlandırılan) ile gelen başvuru sayısı ('Application' olarak adlandırılan) arasındaki ilişkiyi incelemek için yapıldı. İőe alım süreçlerinin etkinlięini ve etkinlięini anlamak adına, iki deęişken arasındaki ilişkiyi belirlemek için veri analizi tekniklerini kullandık.

Scatterplot grafięi analizi, 'View' ve 'Application' arasında belirgin bir ilişki olduęunu ortaya koydu, ancak bazı uç deęerlerin varlıęını da tespit ettik. Korelasyon analizi, iki deęişken arasında orta derecede bir pozitif ilişki olduęunu doęruladı ve korelasyon katsayısı 0.521 olarak hesaplandı.

Yaptıęımız regresyon analizi sonucunda, 'View' ve 'Application' arasında anlamlı bir ilişki olduęu sonucuna vardık. Ancak R-squared deęeri, modelin 'Application' varyansının yalnızca %27.2'sini açıkladıęını belirtti. Bu, genellikle düşük kabul edilen bir deęerdir, ancak modelin anlamlılıęını deęerlendirmede tek faktör olmamalıdır. F istatistięi (26.86) ve p-deęeri (1.92e-06) modelin genel anlamlılıęını doęruladı, bu da 'View' deęişkeninin 'Application' üzerinde

istatistiksel olarak anlamlı bir etkisi olduğunu gösterdi. 'View' değişkeninin katsayısı 7.3347 olarak belirlendi, bu da 'View' değişkeninde bir birimlik artışın 'Application' değişkeninde 7.3347 birimlik bir artışa sebep olacağını gösterdi.

Araştırmamızın sonuçları, firmaların ön değerlendirme platformunu kullanarak adaylara bakma süreçlerinin, gelen başvuru sayısı ile anlamlı bir ilişki içinde olduğunu göstermektedir. Ancak, 'View' değişkeninin 'Application' değişkenindeki varyansın sadece %27.2'sini açıkladığı tespit edildi. Bu, başvurular üzerinde önemli bir etkisi olabilecek diğer faktörlerin varlığını göstermektedir.

İşe alım süreçlerini daha da optimize etmek adına, bir sonraki adımımız, 'View' dışındaki potansiyel etkenleri belirlemek ve bu etkenlerin başvurular üzerindeki etkilerini ölçmek olmalıdır. Ayrıca, veri setindeki uç değerlerin sonuçları nasıl etkilediği üzerinde daha ayrıntılı bir analiz yapmak da önemlidir.

Firmaların adayların ön değerlendirme süreçlerini optimize etmelerine yardımcı olacak verileri sağlama kapasitemiz, işe alım süreçlerini daha da etkili hale getirmek için kritik önem taşır. 'View' değişkenindeki bir birimlik artışın, başvuru sayısında 7.3347 birimlik bir artışa yol açacağını belirlememiz, firmaların adayları ön değerlendirmeye ne kadar çok zaman ayırdıklarının, başvuru sayılarına önemli bir etkisi olduğunu göstermektedir.

Ancak, başvurular üzerinde anlamlı bir etkisi olabilecek diğer etkenlerin varlığını da göz önünde bulundurmalıyız. Örneğin, iş ilanının niteliği, firmaların sektörel itibarı, maaş ve diğer yan haklar, işe alım sürecinin uzunluğu ve karmaşıklığı gibi faktörler, adayların bir iş ilanına başvurup başvurmayacağını etkileyebilir. Bu faktörlerin etkisini ölçmek ve anlamak, firmaların başvuru sayılarını artırmak ve kaliteli adayları çekmek için stratejilerini daha iyi şekillendirmelerine yardımcı olabilir.

Bu araştırma, işe alım süreçlerini optimize etmek için kullanılacak önemli veriler sağlamıştır ve bu alanda daha fazla çalışma yapılması, işe alım süreçlerini daha da geliştirmemize yardımcı olacaktır. Araştırmamızın sonuçları, işe alım süreçlerinde adayları değerlendirme ve başvuruları artırma stratejilerinin etkinliği hakkında önemli bir perspektif sağlamaktadır.

Kaynaklar

Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.

Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business intelligence and analytics: from big data to big impact. *MIS quarterly*, 36(4), 1165-1188.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95.

Lewis, R., & Rao, J. M. (2015). The unfavorable economics of measuring the returns to advertising. *The Quarterly Journal of Economics*, 130(4), 1941-1973.

McKinney, W. (2012). *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc."

Nikolaou, I. (2020). Selection methods for big data in recruitment. *Current Opinion in Psychology*, 40, 92-97.

Nikolaou, I. (2020). Artificial intelligence in recruitment and selection. *The International Journal of Management Reviews*, 22(4), 471-490.

Ekler

Ek A

Aşağıda proje içerisinde kullanılmış ham veri setleri ve bu setlerinin işlendiği notebook ortamının linki yer almaktadır.

EK A.1

Notebook Drive Linki:

<https://colab.research.google.com/drive/1TwD9T4sarXgE-dCELGDXffxeYesv-65h?usp=sharing>

EK A.2

Candidate View Company Based Veri Seti Drive Linki:

<https://drive.google.com/file/d/1KsftoAScMjCfGhBPTEA4xSipQvDEXrb-/view?usp=sharing>

EK A.3

Candidate Application Company Based Veri Seti Drive Linki:

https://docs.google.com/spreadsheets/d/1XcqCQKGGLAjyc52ZWh_QAa81t3lC0Yn7/edit?usp=sharing&ouid=114508582385803059753&rtpof=true&sd=true

Özgeçmiş

Adı Soyadı: Eda Yıldız

Eğitim:

2017 - 2021 Maltepe Üniversitesi, Sosyoloji
2022 - Devam ediyor. İzmir Kâtip Çelebi Üniversitesi, Yazılım Mühendisliği

İş Deneyimi:

2020 – 2021 Moodsta / Customer Success Specialist
2021 – 2022 DIGITA / Customer Success Specialist
2022 – Devam ediyor. Coensio / Customer Success Specialist

Yayınlar:

3. Meral Özbek'in “Arabesk Kültür” Makale İncelemesi ve Tartışması
(<https://edayildiz37.academia.edu/>)
4. HABERMAS’IN FİKİR VE KAVRAM İNCELEMESİ
(<https://edayildiz37.academia.edu/>)
5. LEFEBVRE VE HALL KARŞILAŞTIRMALI İNCELEME: KURAM VE KAVRAM
(<https://edayildiz37.academia.edu/>)