Büyük Veri Analizi

Ders 4

Ali Mertcan KOSE Ph.D.
amertcankose@ticaret.edu.tr

İstanbul Ticaret Üniversitesi

Giriş

Bir veri bilimi projesinin en önemli aşamalarından biri ve ilk adımı, bir iş sorununu anlamak ve tanımlamaktır. Ancak bu, mevcut sorunun bir ifade veya yazılı bir rapor olarak tekrarlanmasından ibaret olamaz. Bir iş sorununu ayrıntılı olarak incelemek ve kapsamını tanımlamak için, mevcut iş metriklerini kullanarak ilgili kalıpları açıklayabilir veya geçmiş verileri nicelleştirip analiz ederek yeni metrikler üretebiliriz. Bu tür tanımlanmış metrikler, mevcut sorunu ölçen ve iş paydaşlarına sorunun etkisini ileten Temel Performans Göstergeleri'dir (KPI'lar). Bu bölüm, bir iş sorununu anlamak ve tanımlamak, ilgili temel metrikleri belirlemek ve bu belirlenen ve oluşturulan KPI'ları tanımlayıcı analizler için Pandas ve benzeri kütüphaneler aracılığıyla kullanmakla ilgilidir. Bölüm ayrıca, yapılandırılmış bir yaklaşım ve metodoloji aracılığıyla bir veri bilimi projesinin nasıl planlanacağını ele alıyor ve bir sorunun grafik ve görselleştirme teknikleri kullanılarak nasıl temsil edileceğiyle son buluyor.

Bir İş Problemini Tanımlamak

Veri biliminde bir iş problemi, bir işletmenin karşılaştığı uzun veya kısa vadeli bir zorluktur ve bu zorluk, iş hedeflerine ulaşılmasını engelleyebilir ve verimli bir veri odaklı karar sistemiyle önlenebilecek büyüme ve sürdürülebilirlik için bir kısıt görevi görebilir. Bazı tipik veri bilimi iş problemleri şunlardır: gelecek haftaki tüketici ürünlerine olan talebi tahmin etmek, üçüncü taraf lojistik (3PL) için lojistik operasyonlarını optimize etmek ve sigorta taleplerindeki hileli işlemleri tespit etmek.

Bir İş Problemini Tanımlamak

Veri bilimi ve makine öğrenimi, verileri önceden oluşturulmuş algoritmalara aktararak bu iş problemlerini çözebilecek sihirli teknolojiler değildir. Uçtan uca analitik projeler oluşturmak için gereken yaklaşım ve tasarım açısından karmaşıktırlar. Bir işletme bu tür çözümlere ihtiyaç duyduğunda, nihai hedefin net bir şekilde anlaşılmaması durumunda bir gereksinim boşluğu oluşturan bir durumla karşılaşabilirsiniz. Bunun için güçlü bir temel, iş problemini nicel olarak tanımlamak ve ardından gereksinimler doğrultusunda kapsam belirleme ve çözümler geliştirmekle başlar. Aşağıda, günümüz sektörünün karşılaştığı ve veri bilimi ve analitiği yoluyla çözülen yaygın iş sorunları hakkında sezgisel bir fikir verecek yaygın veri bilimi kullanım örneklerine birkaç örnek verilmiştir:

Bir İş Problemini Tanımlamak

- Yanlış talep/gelir/satış tahminleri
- Zayıf müşteri dönüşümü, kaybı ve elde tutma
- Kredi sektöründe ve sigortacılıkta dolandırıcılık ve fiyatlandırma
- Etkisiz müşteri ve satıcı/dağıtıcı puanlaması
- Çapraz satış/yukarı satış için etkisiz öneri sistemleri
- Öngörülemeyen makine arızaları ve bakımı
- Metin verileri aracılığıyla müşteri duygusu/duygu analizi
- Yapılandırılmamış veri analitiği gerektiren tekrarlayan görevlerin otomatikleştirilmemesi

Bir İş Problemini Tanımlamak

Hepimizin bildiği gibi, son birkaç yılda sektörler, teknoloji ve inovasyon ortamının etkisiyle muazzam bir değişim geçirdi. Gelişen teknolojinin hızıyla birlikte, başarılı işletmeler de buna uyum sağlıyor ve bu da son derece gelişen ve karmaşık iş zorluklarına ve sorunlarına yol açıyor. Böylesine dinamik bir ortamda yeni iş sorunlarını anlamak kolay bir süreç değil. Ancak, duruma göre iş sorunları ve bunlara yaklaşımlar değişebilir. Ancak, bu yaklaşım büyük ölçüde genelleştirilebilir.

Bir İş Problemini Tanımlamak

Aşağıdaki ipuçları, bir iş sorununu tanımlamak ve sonuçlandırmak için adım adım geniş bir yaklaşımdır ve aşağıdaki bölümde her adımın ayrıntılı bir açıklaması verilmiştir:

1. Sorun tanımlama
2. Gereksinim toplama
3. Veri hattı ve iş akışı
4. Ölçülebilir metrikleri belirleme
5. Dokümantasyon ve sunum

Sorun Tanımlama

Yatırım fonları alanında müşteri ediniminde güçlü bir etkiye sahip olan, yani doğru müşterileri hedefleyip onları sisteme dahil eden bir Varlık Yönetim Şirketi'nin (AMC), veri bilimi tabanlı çözümlerle premium müşterilerinin ortalama müşteri gelirini ve cüzdan payını artırmak için daha yüksek müşteri sadakati sağlamaya çalıştığı bir örnekle başlayalım. Buradaki iş sorunu, mevcut müşterilerden elde edilen geliri ve cüzdan paylarını nasıl artıracağımızdır.

Sorun ifadesi ise "Müşteri sadakati analitiğiyle premium müşterilerin ortalama müşteri gelirini ve cüzdan payını nasıl artırabiliriz?" şeklindedir. Sorunu belirtildiği gibi özetlemek, bir iş sorununu tanımlamanın ilk adımı olacaktır.

Gereksinim Toplama

Sorun tespit edildikten sonra, müşterinizle, konunun uzmanı (SME) veya soruna hakim biri de dahil olmak üzere, nokta nokta bir görüşme yapın. Sorunu onların bakış açısından anlamaya çalışın ve konuyla ilgili çeşitli bakış açılarından sorular sorun, gereksinimleri anlayın ve mevcut geçmiş verilerden sorunu nasıl tanımlayabileceğinizi belirleyin. Bazen, müşterilerin sorunu tam olarak anlayamadıklarını fark edeceksiniz. Bu gibi durumlarda, her ikiniz için de tatmin edici bir sorun tanımı oluşturmak için müşterinizle birlikte çalışmalısınız.

Veri Boru Hattı ve İş Akışı

Konuyu ayrıntılı olarak kavradıktan sonraki aşama, sorunu ölçmek için ölçülebilir metrikleri belirlemek ve bunlar üzerinde anlaşmaya varmaktır; yani, daha ileri analizler için kullanılacak metrikler konusunda müşterilerle anlaşmaya varmaktır. Uzun vadede, bu sizi birçok sorundan kurtaracaktır. Bu metrikler, işletmenin performansını izlemek için kullanılan mevcut sistemle ilişkilendirilebilir veya geçmiş verilerden yeni metrikler türetilebilir.

Veri Boru Hattı ve İş Akışı

Sorunu takip etmek için metrikleri incelediğinizde, sorunu tanımlama ve niceleme verileri birden fazla veri kaynağından, veritabanlarından, eski sistemlerden, gerçek zamanlı verilerden vb. gelebilir. Bu süreçte yer alan veri bilimcisinin, gerekli verileri çıkarmak, toplamak ve daha ileri analiz için analitik araçlara aktarmak üzere müşterinin veri yönetimi ekipleriyle yakın bir şekilde çalışması gerekir. Bunun için, veri toplamak için güçlü bir veri hattına ihtiyaç vardır. Elde edilen veriler, önemli niteliklerini ve zaman içinde nasıl değiştiklerini belirlemek ve KPI'ları oluşturmak için daha ayrıntılı olarak analiz edilir. Bu, müşteri etkileşiminde önemli bir aşamadır ve ekipleriyle birlikte çalışmak, işi kolaylaştırmak için büyük önem taşır.

Ölçülebilir Metrikleri Belirleme

Gerekli veriler veri hatları aracılığıyla toplandıktan sonra, geçmiş verileri analiz etmek ve iş sorununa dair içgörüler üretmek için tanımlayıcı modeller geliştirebiliriz.

Tanımlayıcı modeller/analizler, zaman trendi analizi ve veri analizinin yoğunluk dağılımı gibi yöntemlerle geçmişte neler olduğunu bilmekle ilgilidir. Bunun için, hangi veri özelliklerinin mevcut sorunla ilişkili olduğuna dair içgörüler elde etmek amacıyla geçmiş verilerden çeşitli özelliklerin incelenmesi gerekir.

Ölçülebilir Metrikleri Belirleme

Önceki örnekte açıklandığı gibi, bir Müşteri Memnuniyeti Yönetim Şirketi (AMC) müşteri elde tutma ile ilgili belirli bir iş sorununa çözüm aramaktadır. Elde tutma sorununu anlamak için KPI'ları nasıl oluşturabileceğimizi belirlemeye çalışacağız.

Bunun için, geçmiş veriler, önceki yatırımların müşteri işlem modellerini analiz etmek ve bunlardan KPI'lar türetmek için kullanılır. Bir veri bilimcisinin, bu KPI'ları, sorunun değişkenliğini veya bu durumda müşteri elde tutmayı açıklamadaki alaka ve verimliliklerine göre geliştirmesi gerekir.

Dokümantasyon ve Sunum

Son adım, belirlenen KPI'ları, önemli eğilimlerini ve bunların işletmeyi uzun vadede nasıl etkileyebileceğini belgelemektir. Önceki müşteri elde tutma örneğinde, tüm bu metrikler -ilişki süresi, ortalama işlem sıklığı, müşteri kaybı oranı- KPI görevi görebilir ve sorunu nicel olarak açıklamak için kullanılabilir. Müşteri kaybı oranındaki eğilimi gözlemlersek ve bu örnekte, son birkaç ayda artan bir eğilim gösterdiğini varsayalım ve bunu grafiksel olarak gösterirsek, müşteri, müşteriyi müşteri kaybından önce tespit etmek ve daha güçlü bir müşteri elde tutma hedeflemek için öngörücü müşteri kaybı analitiğinin önemini kolayca anlayabilir.

Bir müşteri sadakati sisteminin potansiyelinin müşteriye sunulması gerekir; bunun için KPI'ların dokümantasyonu ve grafiksel gösterimi yapılmalıdır. Önceki durumda, belirlenen KPI'lar, modellerindeki değişikliklerle birlikte dokümante edilmeli ve müşteriye sunulmalıdır.

Bir İş Problemini Ölçülebilir Metriklere ve Keşifsel Veri Analizine (EDA) Dönüştürme

Belirli bir iş sorunuyla karşılaştığımızda, o iş sorununu tanımlayan KPI'ları belirlememiz ve ilgili verileri incelememiz gerekir. Sorunla ilgili KPI'lar oluşturmanın ötesinde, eğilimleri incelemek ve Keşifsel Veri Analizi (EDA) yöntemleriyle sorunu nicelleştirmek bir sonraki adım olacaktır. KPI'ları keşfetme yaklaşımı şu şekildedir:

- Veri toplama
- Veri üretiminin analizi
- KPI görselleştirme
- Özellik önemi

Veri Toplama

Sorunu analiz etmek için gereken veriler, iş sorununu tanımlamanın bir parçasıdır. Ancak, verilerden öznitelik seçimi iş sorununa göre değişecektir. Aşağıdaki örnekleri inceleyin:

- Eğer bir öneri motoru veya müşteri kaybı analizi söz konusuysa, diğer verilerin yanı sıra geçmiş satın alımlara ve Müşterinizi Tanıyın (KYC) verilerine bakmamız gerekir.
- Eğer talep tahminiyle ilgiliyse, günlük satış verilerine bakmamız gerekir. Gerekli verilerin sorundan soruna değişebileceği sonucuna varılmalıdır.

Veri Üretiminin Analizi

Mevcut veri kaynaklarından bir sonraki adım, tanımlanan problemle ilgili metrikleri belirlemektir. Verilerin ön işlenmesinin yanı sıra , bazen bu metrikleri üretmek için verileri işlememiz gerekebilir veya bunlar doğrudan verilen veride bulunabilir.

Örneğin, sensör veya bilgisayar tarafından oluşturulan kayıt verilerinin kullanıldığı, öngörücü bakım problemi (hizmet içi ekipman veya makinelerin arızalanmadan önce durumunu tahmin etmek için öngörücü analitiğin kullanıldığı bir problem) gibi denetlenen bir analize baktığımızı varsayalım. Kayıt verileri yapılandırılmamış olsa da, hangi kayıt dosyalarının makinelerin arızalarını açıkladığını ve hangilerinin açıklamadığını belirleyebiliriz. Yapılandırılmamış veriler sütun veya satır içermez.

KPI Görselleştirme

KPI'lardaki eğilimleri ve kalıpları anlamak için bunları etkileşimli görselleştirme teknikleriyle temsil etmemiz gerekir. Kutu grafiği, zaman-eğilim grafikleri, yoğunluk grafikleri, dağılım grafikleri, pasta grafikleri ve ısı haritaları gibi farklı yöntemler kullanabiliriz. Bu konuda daha fazla bilgiyi bu bölümdeki XX. Alıştırma, Hedef Değişken için Özellik Önemini Oluşturma ve EDA Gerçekleştirme'de öğreneceğiz.

Özellik Önemi

Hedef değişken belirlendikten sonra, verilerdeki diğer niteliklerin ve bunların hedef değişkenin değişkenliğini açıklamadaki önemlerinin incelenmesi gerekir. Bunun için, hedef değişkenle diğer değişkenlerin (açıklayıcı veya bağımsız değişkenler) ilişkilerini kurmak amacıyla ilişki, varyans ve korelasyon yöntemlerini kullanırız.

Çalışmadaki değişkenlerin türüne bağlı olarak kullanılabilecek Pearson Korelasyonu, Ki-Kare testleri ve Gini değişken önemine dayalı algoritmalar, karar ağaçları ve Boruta gibi birden fazla özellik-önem yöntemi ve algoritması vardır.

Aşağıdaki alıştırmada, veri toplama ve analizini (birden fazla veri kaynağının birleştirilerek veya bir araya getirilerek analiz için tek bir veri kümesi oluşturulmasıyla elde edilen veriler) KPI görselleştirmesiyle ele alacağız ve ardından sonraki alıştırmada, özellik öneminin ne olduğunu ele alacağız.

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

1. bank.csv verilerini aşağıdaki çevrimiçi kaynaklardan indirin:
   - https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
   - https://archive.ics.uci.edu/ml/machine-learning-databases/00222/
2. Egzersiz için bir klasör oluşturun (packt_exercises) ve indirilen verileri oraya kaydedin.

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

3. Jupyter not defterini başlatın ve gerekli tüm kütüphaneleri gösterildiği gibi içe aktarın. Şimdi os.chdir() fonksiyonunu kullanarak çalışma dizinini ayarlayın:

```
import numpy as np
import pandas as pd
import seaborn as sns
import time
import re
import os
import matplotlib.pyplot as plt
sns.set(style="ticks")
os.chdir("/Users/svk/Desktop/packt_exercises")
```

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

4. CSV'yi okumak ve veri setini incelemek için aşağıdaki kodu kullanın:

```
df = pd.read_csv('bank.csv', sep=';')
df.head(5)
print(df.shape)
df.head(5)
df.info()
df.describe()
```

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

5. Önceki komutu çalıştırdıktan sonra aşağıdakine benzer bir çıktı alacaksınız:

Hedef değişkeni (abone olunan veya olunmayan - y) incelerken, dağılımına bakmak önemlidir. Bu veri kümesindeki hedef değişkenin türü kategoriktir veya birden fazla sınıftan oluşur. Bu durumda, ikili (Evet/Hayır) olur. Dağılım tek bir sınıfa doğru eğildiğinde, sorun değişkendeki dengesizlik olarak bilinir. Hedef değişkenin oranını bir çubuk grafiği kullanarak inceleyebiliriz. Bu bize her sınıftan kaç tane olduğu (bu durumda, hayır ve evetten kaç tane olduğu) hakkında bir fikir verir. Hayır oranı evet oranından çok daha yüksektir ve bu da verilerdeki dengesizliği açıklar.

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

6. Verilen veriler için çubuk grafiğini çizmek üzere aşağıdaki komutları çalıştıralım:

```
count_number_susbc = df["y"].value_counts()
sns.barplot(count_number_susbc.index, count_number_susbc.va
df['y'].value_counts()
```

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

7. Şimdi, her bir değişkeni ele alıp dağılım eğilimlerine bakacağız.

Örnek olarak sunulan aşağıdaki histogram, veri kümesindeki 'yaş' sütununa (öznitelik) aittir. Histogramlar/yoğunluk grafikleri, çubuk grafiklere benzer şekilde sayısal/kayan noktalı değişkenleri incelemek için harika bir yoldur. Kategorik veri değişkenleri için kullanılabilirler. Burada, örnek olarak bir histogram kullanarak yaş ve denge olmak üzere iki sayısal değişkeni ve çubuk grafikler kullanarak eğitim ve ay olmak üzere iki kategorik değişkeni göstereceğiz:

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

```
# yaş için histogram (matplotlib kullanılarak)

plt.hist(df['age'], color = 'grey', edgecolor =
'black',
bins = int(180/5))

# Yaş için histogram (Seaborn kullanılarak)

sns.distplot(df['age'], hist=True, kde=False,
bins=int(180/5), color = 'blue',
hist_kws='edgecolor':'black')
```

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

8. Veri kümesindeki denge niteliğinin histogramını çizmek için aşağıdaki komutu kullanın:

```
# denge için histogram (matplotlib kullanılarak)

plt.hist(df['balance'], color = 'grey', edgecolor =
'black',
bins = int(180/5))

# histogram for balance (using seaborn)

sns.distplot(df['balance'], hist=True, kde=False,
bins=int(180/5), color = 'blue',
hist_kws='edgecolor':'black')
```

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

9. Şimdi, aşağıdaki kodu kullanarak veri setindeki eğitim niteliği için bir çubuk grafiği çizin:

```
# 'eğitim' değişkeni için barplot

count_number_susbc = df["eğitim"].value_counts()
sns.barplot(count_number_susbc.index,
count_number_susbc.values)
df['eğitim'].value_counts()
```

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

10. Veri kümesinin ay niteliği için bir çubuk grafiği çizmek üzere aşağıdaki komutu kullanın:

```
# barplot for the variable 'month'

count_number_susbc = df["month"].value_counts()
sns.barplot(count_number_susbc.index, count_number_susbc.va
df['education'].value_counts()
```

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

11. Bir sonraki görev, hedef değişkenin her sınıfı için bir dağılım oluşturmak ve dağılımları karşılaştırmaktır. Hedef değişken için yaş niteliğinin histogramını çizin (evet/hayır):

```
{# her abonelik türü için ayrı bir liste oluştur\

x1 = list(df[df['y'] == 'yes']['age'])\

x2 = list(df[df['y'] == 'no']['age'])\

# assign colors for each subscription type\

colors = ['#E69F00', '#56B4E9']\

names = ['yes', 'no']\
```

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

```
# histogramı çiz\
plt.hist([x1, x2], bins = int(180/15), density=True,\
color = colors, label=names)\
# plot formatting\
plt.legend()\
plt.xlabel('IV')\
plt.ylabel('prob distr (IV) for yes and no')\
plt.title('Histogram for Yes and No Events w.r.t. IV')}
```

Verilen Verilerden İşletme Problemi için Hedef Değişkeni ve İlgili KPI'ları Belirleyin

12. Şimdi, aşağıdaki komutu kullanarak hedef değişken için ay bazında gruplandırılmış bir çubuk grafiği çizin:

```
df.groupby(["ay", "y"]).size().unstack().plot(tür='bar',
yığılmış=True, figsize=(20,10))
```

Bu alıştırmada, KPI'ları ve hedef değişkeni (veri toplama ve analiz verisi (analiz için tek bir veri kümesi elde etmek üzere birden fazla veri kaynağının birleştirilmesi veya birleştirilmesiyle oluşturulan veriler)) belirlemeyi inceledik. KPI'lar ve hedef değişken (KPI görselleştirmesi) belirlendi. Şimdi, bir sonraki alıştırmada, hedef değişkenin varyansını açıklama açısından hangi değişkenlerin önemli olduğunu belirleyeceğiz (özellik önemi).

Hedef Değişkenin Özellik Önemini Oluşturun ve EDA'yı Gerçekleştirin

Önceki alıştırmada, niteliklerin eğilimlerini inceleyerek dağılımlarını belirledik ve bunları gerçekleştirmek için çeşitli grafik ve görselleştirme yöntemlerini nasıl kullanabileceğimizi inceledik. İster öngörücü ister sınıflandırma problemi olsun, bir modelleme problemini ele almadan önce (örneğin, önceki pazarlama kampanyası verilerinden, dönüşüm olasılığı en yüksek olan gelecekteki müşterileri nasıl tahmin edeceğimiz), verileri önceden işlememiz ve abonelik kampanyası çıktı modellerini etkileyecek önemli özellikleri seçmemiz gerekir. Bunu yapmak için, niteliklerin sonuçla (hedef değişken) ilişkisini, yani hedef değişkenin her değişken tarafından ne kadar değişkenlik gösterdiğini görmemiz gerekir.

Hedef Değişkenin Özellik Önemini Oluşturun ve EDA'yı Gerçekleştirin

Değişkenler arasındaki ilişkiler birden fazla yöntem kullanılarak kurulabilir; ancak bir yöntem/algoritma seçerken veri türünü göz önünde bulundurmalıyız. Örneğin, sayısal değişkenleri (sıralı tam sayılar, ondalıklı sayılar vb.) inceliyorsak, korelasyon analizini; birden fazla sınıfa sahip kategorik değişkenleri inceliyorsak, Ki-Kare yöntemlerini kullanabiliriz. Ancak, her ikisini de birlikte işleyebilen ve değişkenlerin önemini karşılaştırmak için ölçülebilir sonuçlar sağlayan birçok algoritma vardır. Bu alıştırmada, özelliklerin önemini belirlemek için çeşitli yöntemlerin nasıl kullanılabileceğine bakacağız:

Hedef Değişkenin Özellik Önemini Oluşturun ve EDA'yı Gerçekleştirin

1. bank.csv dosyasını indirin ve aşağıdaki komutu kullanarak verileri okuyun:

```
import numpy as np
import pandas as pd
import seaborn as sns
import time
import re
import os
import matplotlib.pyplot as plt
sns.set(style="ticks")
```

Hedef Değişkenin Özellik Önemini Oluşturun ve EDA'yı Gerçekleştirin

```
os.chdir("/Users/svk/Desktop/packt_exercises")

# read the downloaded input data (marketing data)\

df = pd.read_csv('bank.csv', sep=';')\
```

Hedef Değişkenin Özellik Önemini Oluşturun ve EDA'yı Gerçekleştirin

2. Değişkenler arasındaki korelasyonu belirlemek için aşağıdaki komutu kullanarak bir korelasyon matrisi geliştirin:

```
df['y'].replace(['yes','no'],[1,0],inplace=True)
df['default'].replace(['yes','no'],[1,0],inplace=True)
df['housing'].replace(['yes','no'],[1,0],inplace=True)
df['loan'].replace(['yes','no'],[1,0],inplace=True)
corr_df = df.corr()

sns.heatmap(corr_df, xticklabels=corr_df.columns.values,
yticklabels=corr_
df.columns.values, annot = True, annot_kws='size':12)
heat_map=plt.gcf(); heat_map.set_size_inches(10,5)
plt.xticks(fontsize=10); plt.yticks(fontsize=10);
plt.show()
```

Hedef Değişkenin Özellik Önemini Oluşturun ve EDA'yı Gerçekleştirin

3. Boruta'ya (rastgele bir orman etrafında bir sarmalayıcı algoritma) dayalı bir özellik önem çıktısı oluşturun:

```
# import DecisionTreeClassifier from sklearn and

# BorutaPy from boruta

import numpy as np

from sklearn.ensemble import RandomForestClassifier

from boruta import BorutaPy

# transform all categorical data types to integers (hot−encoding)}

for col_name in df.columns:
if(df[col_name].dtype == 'object'):
df[col_name]= df[col_name].astype('category')
df[col_name] = df[col_name].cat.codes
```

Hedef Değişkenin Özellik Önemini Oluşturun ve EDA'yı Gerçekleştirin

```
# generate separate dataframes for IVs and DV (target variable)

X = df.drop(['y'], axis=1).values
Y = df['y'].values

# build RandomForestClassifier, Boruta models and #
related parameter

rfc = RandomForestClassifier(n_estimators=200, n_jobs=4,
class_

weight='balanced', max_depth=6)
boruta_selector = BorutaPy(rfc, n_estimators='auto',
verbose=2)
n_train = len(X)

# fit Boruta algorithm

boruta_selector.fit(X, Y)
```

Hedef Değişkenin Özellik Önemini Oluşturun ve EDA'yı Gerçekleştirin

4. Özelliklerin sıralamasını aşağıdaki gibi kontrol edin:

```
feature_df = pd.DataFrame(df.drop(['y'],
axis=1).columns.tolist(),
columns=['features'])
feature_df['rank']=boruta_selector.ranking_
feature_df = feature_df.sort_values('rank',
ascending=True).reset_
index(drop=True)
sns.barplot(x='rank',y='features',data=feature_df)
feature_df
```