

Klasik Makine Ogrenmesi Yontemleri ile Mesleklerin Siniflandirilmesi(kariyer.net)

Havvanur Dervişoğlu,drvshavva@gmail.com

August 29, 2019

1 Çalışmanın Tanımı ve Kullanılan Araçlar

Bu calismadaki adimlar:

- Web scraping ile veri setinin olusturulmasi.
- Olusturulan veri setine preprocessing uygulanmasi.
- On islem asamasindan gecmis veri setindeki is tanimi ozelligine feature extraction uygulanmasi. Calisma da tf-idf ve count vectorizer oznitelik cikarimi yontemleri farkli parametrelerle denenmis olup siniflandirma basirilari kaydedilmistir.
- Oznitelik cikarimi asamasindan sonra klasik makine ogrenmesi yontemleri ile siniflandirma basarilarinin sonuclari alinmistir. Klasik makine ogrenmesi yontemlerinden; logistic regression, random forest, decision tree classifier ve naive bayes siniflandiricilari kullanilmistir.

1.1 Web Scraping icin Kullanilan Araclar

- Calisma da web scraping icin PyCharm calisma ortami kullanilmistir.
- Programa dili olarak python kullanilmistir.
- Kariyer.net'ten veri cekmek icin selenium ve scrapy kutuphaneleri kullanilmistir.

1.2 Siniflandirma icin Kullanilan Araclar

Veri setinde uygulanan preprocessing islemi icn google colab kullanilmistir. Mesleklerin siniflandirilasi icin Pyspark kullanilmistir ve Zeppelin gelistirme ortami kullanilmistir.

2 Web Scraping

Kariyer.net web sitesinde veri çekmek için scrapy ve selenium kutuphaneleri kullanılmıştır. Kodda sınıf ilan aramaları sınıf bazında yapılmaktadır. Figure 1’de, oluşturulacak sınıfın adı sınıf değişkenine atanmıştır, sonuçların yazılacağı dosya adı ve tipi belirtilmiştir.

```
#bu programda kariyer.net sitesinden belirlenen kategori ismi belirtilerek arama yapılması ve sonuçların bir csv dosyasına
#kaydedilmesi hedeflenmiştir
class IlanlarSpider(scrapy.Spider):
    name = 'ilanlar'

    allowed_domains = ['www.kariyer.net']
    start_urls = ['https://www.kariyer.net/is-ilanlari']

    # olusturalacak kategori ismi
    sinif = 'banka-sigorta'
    #sonuçların kaydedilmesi için önce gerekli dizini belirtiyoruz
    custom_settings = {'FEED_FORMAT': 'csv',
                       'FEED_URI': '/home/safir/kariyer/'+sinif+'_'+%(time)s+'.csv'}

    def parse(self, response):
```

Figure 1: Web scraping-1(kariyer.net)

Figure 2’de, selenium kutuphanesinin kullanımı için gerekli olan web driverin olduğu dizin belirtilmiştir ve sonrasında açılan web sitesinde aramanın yapıldığı yere gidilip sınıf adı aratılmıştır.

```
def parse(self, response):
    #bilgisayarınızda chromedriver'in olduğu dizini burada belirtiyoruz
    webdriver_path = '/home/safir/Desktop/chromedriver'
    self_driver = webdriver.Chrome(webdriver_path)
    self_driver.get('https://www.kariyer.net/is-ilanlari')

    #sayfada arama yapılan yere kategori ismini yazıyoruz
    search_item = self_driver.find_element_by_xpath('// *[@id = "txtSearchKeyword"]')
    search_item.clear()
    search_item.send_keys(IlanlarSpider.sinif)
    sleep(1.8)
    #kategori ismini yazdıktan sonra butona tikliyoruz
    next_page = self_driver.find_element_by_xpath('//*[@id="btnSearchKeyword"]')
    self_driver.execute_script("arguments[0].click();", next_page)
    sleep(4.5)

    items = KariyerItem()
    #next_page = 'https://www.kariyer.net/is-ilanlari'
    ilan_links = []
    page = 0
    #23 sayfadan ilanlara ait linkleri alıyoruz
```

Figure 2: Web scraping-2(kariyer.net)

Figure 3’de, arama sonucunda cikan ilanlari sayfa sayfa ilerleyerek toplamaktadır.

```
#23 sayfadan ilanlara ait linkleri aliyoruz
while page <= 23:
    scrapy_selector = Selector(text=self_driver.page_source)
    links = scrapy_selector.xpath('//a[@class="link_position"]/@href').extract()
    for i in range(0, len(links)):
        ilan_links.append(links[i])

#sonraki sayfaya geciyoruz
next_page = self_driver.find_element_by_xpath('//a[@id="lnkNextPage"]')
self_driver.execute_script("arguments[0].click();", next_page)
page += 1
sleep(4.0)
```

Figure 3: Web scraping-3(kariyer.net)

Figure 4’de, toplanan ilan linklerine gidilerek ilana dair istenilen ozellikler alınmaktadır.

```
#topladigimiz ilan linklerine gidip belirtilen ozellikleri aliyoruz
for i in range(0, len(ilan_links)):
    url = 'https://www.kariyer.net' + ilan_links[i]
    self_driver.get(url)
    scrapy_selector = Selector(text=self_driver.page_source)
    genel = scrapy_selector.xpath("//*[h3[contains(., 'GENEL NİTELİKLER VE İŞ TANIMI')]/following-sibling::node()//text()']").extract()
    if genel:
        items['ilan_baslik'] = scrapy_selector.xpath('//a[@id="jobTitle"]/text()').extract()
        items['sirket_adi'] = scrapy_selector.xpath('//a[@id="jobCompany"]/text()').extract()
        items['genel_nit_is_tanimi'] = genel
        #items['is_tanimi'] = scrapy_selector.xpath("//*[div//h3[starts-with(., 'İŞ TANIMI')]/following-sibling::node()/descendant-or-self::text()").extract()
        items['tecrube'] = scrapy_selector.xpath('//div[@class="sub-box aday-kriterleri"]/div[2]/div[1]/div[2]/p/text()').extract()
        items['egitim'] = scrapy_selector.xpath('//div[@class="sub-box aday-kriterleri"]/div[2]/div[3]/div[2]/p/text()').extract()
        # items['yabanci_dil'] = scrapy_selector.xpath('//div[@class="sub-box aday-kriterleri"]/div[2]/div[4]/div[2]/p/text()').extract()
        items['sektor'] = scrapy_selector.xpath('//div[@class="sub-box pozisyon-bilgileri"]/div[2]/div[1]/div[2]/p/text()').extract()
        items['departman'] = scrapy_selector.xpath('//div[@class="sub-box pozisyon-bilgileri"]/div[2]/div[2]/div[2]/p/text()').extract()
        items['calisma_sekli'] = scrapy_selector.xpath('//div[@class="sub-box pozisyon-bilgileri"]/div[2]/div[3]/div[2]/p/text()').extract()
        items['sehir'] = scrapy_selector.xpath('//div[@class="sub-box pozisyon-bilgileri"]/div[2]/div[6]/div[2]/p/text()').extract()
        items['label'] = IlanlarSpider.sinif
        sleep(5)

    yield items
```

Figure 4: Web scraping-4(kariyer.net)

3 Veri Seti

Calismada iki farkli veri seti kullanilmistir:

1. Kariyer.net web sitesinden genel 5 meslek grubundan olusan turkce veri seti olusturulmustur
2. Kariyer.net web sitesinden birbirlerileri ile koreslasyonlari yuksek olan 10 meslekten olusan veri seti olusturulmustur.

Amac:

- Turkce veri setinde meslekler arasi korelasyon arttiginda siniflandirma basarisinin nasil degistigini gozlemlemek.

3.1 Kariyer.net Veri Seti-1

Bu veri setinde ilanlara ait kullanılan ozellikler; is tanim, label, ilan basligi, dil, egitim seviyesi, sirket, sektor, sehir seklindedir. Siniflardaki ornek sayilari asagida gosterildigi gibidir:

1. banka-sigorta: 954 ilan
2. egitim-ogretim: 822 ilan
3. bilisim-telekom: 768 ilan
4. saglik : 814 ilan
5. yapi-mimar-insaat: 953 ilan

Veri seti olustururken siniflardaki ornek sayilarinin dengeli olmasina dikkat edilmistir.

3.2 Kariyer.net Veri Seti-2

Bu veri setinde ilanlara ait kullanılan ozellikler; is tanim, label, ilan basligi, dil, egitim seviyesi, sirket, sektor, sehir seklindedir. Siniflardaki ornek sayilari asagida gosterildigi gibidir:

1. acente-sigorta-danisman: 81 ilan
2. avukat: 88 ilan
3. hasta-hizmetleri: 93 ilan
4. is-analizi-raporlama : 68 ilan
5. is-guvenlik-saglik: 67 ilan
6. proje-yonetimi: 105 ilan
7. sap: 75 ilan
8. sistem-yonetimi: 76 ilan
9. tibbi-tanitim : 66 ilan
10. web-tasarim: 74 ilan

Veri seti olustururken siniflardaki ornek sayilarinin dengeli olmasina dikkat edilmistir.

4 Preprocessing

On islem asamasi siniflandirma yapmadan once verilerden anlamlı olan bilgileri cikarmak için önemli bir adımdır; öncelikle Figure 5’de de görüldüğü gibi meslek siniflandırılmasında kullanılmayacak olan özellikleri attık.

```
data.drop(['calisma_sekli', 'egitim', 'sehir', 'sektor', 'sirket_adi', 'tecrube', 'departman'], axis=1, inplace=True)  
data.head()
```

	genel_nit_is_tanimi	ilan_baslik	label
0	Üniversitelerin Bilgisayar Mühendisliği, Bilgi...	Front-End Developer	web_tasarim
1	"İYİ OL" hedefini dünyanın dört bir tarafında ...	Web Yazılım Uzmanı	web_tasarim
2	Üniversitelerin Endüstri Mühendisliği, İşletme...	WEB ANALİST	web_tasarim
3	"İYİ OL" hedefini dünyanın dört bir tarafında ...	Junior Web Yazılım Uzmanı	web_tasarim
4	Tanımı:,Html, css ve js ile taslakları tüm ger...	Web Geliştirme Uzmanı	web_tasarim

Figure 5: Preprocess-1(kariyer.net)

Figure 6’da, veri setinde uygulanan işlemler görülmektedir:

1. Harfleri küçük harfe çevirme
2. Türkçe karakter harfleri İngilizce karaktere çevirme
3. Noktalama işaretleri vb işaretleri atma
4. Rakamları silme
5. Cümleyi kelimelerine ayırıp stop words olup olmadığını kontrol etme ve eğer stop word ise almama
6. Kelime uzunluğu 3 den küçük olanları alma(amac kesme işareti ile ayrılmış kelimelerden kurtulma nin nin gibi)
7. Kelimeler arası bir boşluk olacak şekilde birleştirme.

```
def clean(text):
    processed_tweet = []
    text=text.lower()

    text=text.replace('ş','s')
    text=text.replace('ı','i')
    text=text.replace('ö','o')
    text=text.replace('ü','u')
    text=text.replace('ğ','g')
    text=text.replace('ç','c')

    pattern = r"[{}]" .format('&#*.-"“”’?!.,.:;></_-')
    text = re.sub(pattern, " ", text)

    text = re.sub(r'[0-9]+', ' ', text)
    text = text.strip()
    #burada eger basta i varsa onu kelimededen ayriyor istanbul => i, istanbul
    #soralim stopwords kullanimini ve usteki
    tokens = WPT.tokenize(text)
    filtered_tokens = [token for token in tokens if token not in stop_word_list ]
    filtered_tokens = [token for token in tokens if len(token)>3 ]
    text = ' '.join(filtered_tokens)
    return text
```

Figure 6: Preprocess-2(kariyer.net)

Figure 7’de, veri setinin son gorunumunun csv dosyasina kaydedilmesi gorulmektedir.

```
clean_df = pd.DataFrame(clean_is_tanim,columns=['genel_nit_is_tanimi'])
clean_df['ilan_baslik'] = clean_ilan_baslik
clean_df['label'] = data.label
clean_df.to_csv('cleaned_web_tasarim.csv',encoding='utf-8')
clean_df.head()
```

	genel_nit_is_tanimi	ilan_baslik	label
0	universitelerin bilgisayar muhendisligi bilgis...	front developer	web_tasarim
1	hedefini dunyanin dort tarafinda hayata gecirm...	yazilim uzmani	web_tasarim
2	universitelerin endustri muhendisligi sletme m...	anali	web_tasarim
3	hedefini dunyanin dort tarafinda hayata gecirm...	junior yazilim uzmani	web_tasarim
4	tanimi html taslamlari gereksinimleri gercekle...	gelistirme uzmani	web_tasarim

Figure 7: Preprocess-3(kariyer.net)

Figure 8’de, olusan veri setinde labellara id verilmesi gosterilmistir. Her label icin bir id atamasi yapilmistir.

```
%pyspark
#BURADA LABEL SUTUNUDA YER ALAN SINIFLARA INDEX NUMARASI VERILİYOR
indexer = StringIndexer(inputCol = "label", outputCol = "labelIdx")
df = indexer.fit(df).transform(df)
df.show()
```

Figure 8: Preprocess-4(kariyer.net)

5 Feature Extraction

Veri setlerinde iki farklı özellik çıkarımı yöntemi kullanılmıştır. Özellik çıkarımı uygulanmadan önce kariyer.net sitesinden elde edilen veri集中的 ilan tanımı ve başlığı özellikleri tek bir sütunda birleştirilmiştir ve bu sütuna özellik çıkarımı işlemi uygulanmıştır (Figure 9).

```
%pyspark
#SINIFLANDIRMA İÇİN KULLANILACAK OLAN SUTUNLARI TEK BİR SUTUNDA BİRLESTIRYORUZ
df = df.withColumn('tum_ozellikler', concat(col('genel_nit_is_tanimi'),lit('_'), col('ilan_baslik')))
df.show()
```

genel_nit_is_tanimi	ilan_baslik	label	labelIdx	tum_ozellikler
universitelerin b...	front developer	web_tasarim	6.0	universitelerin b...
hedefini dunyanin...	yazilin uzman	web_tasarim	6.0	hedefini dunyanin...
universitelerin e...	anal	web_tasarim	6.0	universitelerin e...
hedefini dunyanin...	junior yazilin uz...	web_tasarim	6.0	hedefini dunyanin...
tanimi html tasla...	gelistirme uzman	web_tasarim	6.0	tanimi html tasla...
danismanligini ya...	onyuz gelistirme ...	web_tasarim	6.0	danismanligini ya...
front developer g...	onyuz front yazil...	web_tasarim	6.0	front developer g...
universitelerin i...	senior developer a...	web_tasarim	6.0	universitelerin i...
universitelerin i...	senior developer	web_tasarim	6.0	universitelerin i...
front development...	frontend developer	web_tasarim	6.0	front development...
bilsoft yazilin y...	yazilin uzman	web_tasarim	6.0	bilsoft yazilin y...
consulting verini...	designer	web_tasarim	6.0	consulting verini...
sisli osmanbey bu...	tasarin stajyeri	web_tasarim	6.0	sisli osmanbey bu...
sirketinizde cali...	tasarin uzman	web_tasarim	6.0	sirketinizde cali...
tercihleri html is	frontend yazilin	web_tasarim	6.0	tercihleri html is

Figure 9: Özelliklerin birleştirilmesi(kariyer.net)

Figure 10’da, özellik işlemi uygulanılacak olan sütundaki cümlelerin kelimelerine ayrılması işlemi gösterilmiştir. Bu işlem özellik çıkarımı yöntemlerinin o sütunda kullanılması içindir.

```
%pyspark
#ÖZELLİKLERİN BİRLESTIRILMESİYLE ELDE EDİLEN SUTUNDAKİ KELİMLERİ AYIRIYORUZ VE YENİ BİR SUTUNA YERLESTIRIYORUZ
tokenizer = Tokenizer(inputCol="tum_ozellikler", outputCol="words")
wordsData = tokenizer.transform(df)
wordsData.show()
```

genel_nit_is_tanimi	ilan_baslik	label	labelIdx	tum_ozellikler	words
universitelerin b...	front developer	web_tasarim	6.0	universitelerin b...	[universitelerin, ...]
hedefini dunyanin...	yazilin uzman	web_tasarim	6.0	hedefini dunyanin...	[hedefini, dunyan...
universitelerin e...	anal	web_tasarim	6.0	universitelerin e...	[universitelerin, ...]
hedefini dunyanin...	junior yazilin uz...	web_tasarim	6.0	hedefini dunyanin...	[hedefini, dunyan...
tanimi html tasla...	gelistirme uzman	web_tasarim	6.0	tanimi html tasla...	[tanimi, html, ta...
danismanligini ya...	onyuz gelistirme ...	web_tasarim	6.0	danismanligini ya...	[danismanligini, ...]
front developer g...	onyuz front yazil...	web_tasarim	6.0	front developer g...	[front, developer...
universitelerin i...	senior developer a...	web_tasarim	6.0	universitelerin i...	[universitelerin, ...]
universitelerin i...	senior developer	web_tasarim	6.0	universitelerin i...	[universitelerin, ...]
front development...	frontend developer	web_tasarim	6.0	front development...	[front, developme...
bilsoft yazilin y...	yazilin uzman	web_tasarim	6.0	bilsoft yazilin y...	[bilsoft, yazilin...
consulting verini...	designer	web_tasarim	6.0	consulting verini...	[consulting, veri...
sisli osmanbey bu...	tasarin stajyeri	web_tasarim	6.0	sisli osmanbey bu...	[sisli, osmanbey, ...]
sirketinizde cali...	tasarin uzman	web_tasarim	6.0	sirketinizde cali...	[sirketinizde, ca...
tercihleri html is	frontend yazilin	web_tasarim	6.0	tercihleri html is	[tercihleri, html

Figure 10: Cümlelerin kelimelerine ayrılması

Figure 11’de,tf-idf vectorizer oznitelik cikarimi yonteminin kullanim ornegi bulunmaktadir.

```
%pyspark
#OZNITELIK CIKARIMI OLARAK TF-IDF YONTEMI KULLANILDI
hashingTF = HashingTF(inputCol="words", outputCol="rawFeatures", numFeatures=1000)
featurizedData = hashingTF.transform(wordsData)

idf = IDF(inputCol="rawFeatures", outputCol="features")
idfModel = idf.fit(featurizedData)
result = idfModel.transform(featurizedData)
result.show()
```

genel_nit_is_tanimi	ilan_baslik	label labelIdx	tum_ozellikler	words	rawFeatures	features
universitelerin b...	front developer web_tasarim	6.0 universitelerin b...	[universitelerin,...	(1000,[20,26,41,7...	(1000,[20,26,41,7...	
hedefini dunyanin...	yazilim uzman web_tasarim	6.0 hedefini dunyanin...	[hedefini, dunyan...	(1000,[4,14,26,40...	(1000,[4,14,26,40...	
universitelerin e...	anal web_tasarim	6.0 universitelerin e...	[universitelerin,...	(1000,[11,19,21,2...	(1000,[11,19,21,2...	
hedefini dunyanin...	junior yazilim uz... web_tasarim	6.0 hedefini dunyanin...	[hedefini, dunyan...	(1000,[4,14,26,40...	(1000,[4,14,26,40...	
tanimi html tasla...	gelistirme uzman web_tasarim	6.0 tanimi html tasla...	[tanimi, html, ta...	(1000,[1,19,20,26...	(1000,[1,19,20,26...	
danismanligini ya...	onyuz gelistirme ... web_tasarim	6.0 danismanligini ya...	[danismanligini, ...	(1000,[23,26,48,5...	(1000,[23,26,48,5...	
front developer g...	onyuz front yazil... web_tasarim	6.0 front developer g...	[front, developer...	(1000,[19,23,47,5...	(1000,[19,23,47,5...	
universitelerin i...	front developer a... web_tasarim	6.0 universitelerin i...	[universitelerin,...	(1000,[13,26,42,7...	(1000,[13,26,42,7...	
universitelerin i...	senior developer web_tasarim	6.0 universitelerin i...	[universitelerin,...	(1000,[20,26,35,7...	(1000,[20,26,35,7...	
front development...	frontend developer web_tasarim	6.0 front development...	[front, developme...	(1000,[19,23,26,7...	(1000,[19,23,26,7...	
bilsoft yazilim y...	yazilim uzman web_tasarim	6.0 bilsoft yazilim y...	[bilsoft, yazilim...	(1000,[16,50,73,7...	(1000,[16,50,73,7...	
consulting verini...	designer web_tasarim	6.0 consulting verini...	[consulting, veri...	(1000,[5,16,20,26...	(1000,[5,16,20,26...	
sisli osmanbey bu...	tasarim stajyeri web_tasarim	6.0 sisli osmanbey bu...	[sisli, osmanbey,...	(1000,[9,26,34,42...	(1000,[9,26,34,42...	
sirketimizde cali...	tasarim uzman web_tasarim	6.0 sirketimizde cali...	[sirketimizde, ca...	(1000,[50,78,203,...	(1000,[50,78,203,...	
tercihali html is...	frontend yazilim ... web_tasarim	6.0 tercihali html is...	[tercihali html is...	(1000,[13,16,10,2...	(1000,[13,16,10,2...	

+ Ardi Paraman

Figure 11: TF-IDF vectorizer

Figure 12’de,count vectorizer oznitelik cikarimi yonteminin kullanim ornegi bulunmaktadır.

```
%pyspark
#OZNITELIK CIKARIMI OLARAK COUNT_VECTORIZER YONTEMI KULLANILDI
cv = CountVectorizer(inputCol="words", outputCol="features", vocabSize=10000, minDF=5)
model = cv.fit(wordsData)
result = model.transform(wordsData)
result.show()
```

genel_nit_is_tanimi	ilan_baslik	label labelIdx	tum_ozellikler	words	features
universitelerin b...	front developer web_tasarim	6.0 universitelerin b...	[universitelerin,...	(2561,[0,1,2,3,4,...	
hedefini dunyanin...	yazilim uzman web_tasarim	6.0 hedefini dunyanin...	[hedefini, dunyan...	(2561,[0,1,3,4,5,...	
universitelerin e...	anal web_tasarim	6.0 universitelerin e...	[universitelerin,...	(2561,[0,1,2,5,8,...	
hedefini dunyanin...	junior yazilim uz... web_tasarim	6.0 hedefini dunyanin...	[hedefini, dunyan...	(2561,[0,1,3,4,6,...	
tanimi html tasla...	gelistirme uzman web_tasarim	6.0 tanimi html tasla...	[tanimi, html, ta...	(2561,[0,1,4,5,7,...	
danismanligini ya...	onyuz gelistirme ... web_tasarim	6.0 danismanligini ya...	[danismanligini, ...	(2561,[0,1,3,4,5,...	
front developer g...	onyuz front yazil... web_tasarim	6.0 front developer g...	[front, developer...	(2561,[0,4,5,6,8,...	
universitelerin i...	front developer a... web_tasarim	6.0 universitelerin i...	[universitelerin,...	(2561,[0,2,3,4,5,...	
universitelerin i...	senior developer web_tasarim	6.0 universitelerin i...	[universitelerin,...	(2561,[2,3,4,5,6,...	
front development...	frontend developer web_tasarim	6.0 front development...	[front, developme...	(2561,[1,2,3,4,5,...	
bilsoft yazilim y...	yazilim uzman web_tasarim	6.0 bilsoft yazilim y...	[bilsoft, yazilim...	(2561,[0,1,4,6,7,...	
consulting verini...	designer web_tasarim	6.0 consulting verini...	[consulting, veri...	(2561,[1,2,9,10,1...	
sisli osmanbey bu...	tasarim stajyeri web_tasarim	6.0 sisli osmanbey bu...	[sisli, osmanbey,...	(2561,[0,1,2,3,4,...	
sirketimizde cali...	tasarim uzman web_tasarim	6.0 sirketimizde cali...	[sirketimizde, ca...	(2561,[32,91,182,...	
tercihali html is...	frontend yazilim ... web_tasarim	6.0 tercihali html is...	[tercihali html is...	(2561,[2,3,16,17...	

Figure 12: Count vectorizer

6 Classification

Siniflandirma icin kullanılacak olan ozellikler olusan features sutunu ve labelIdx sutunu(Figure 13). Bu calismada siniflandirma algoritmalarindan logistic regression, decision tree, random forest ve naive bayes kullanilmistir.

```
%pyspark
featured_data = result.select('labelIdx','features')
featured_data.show()
```

labelIdx	features
6.0	(1000,[20,26,41,7...]
6.0	(1000,[4,14,26,40...]
6.0	(1000,[11,19,21,2...]
6.0	(1000,[4,14,26,40...]
6.0	(1000,[1,19,20,26...]
6.0	(1000,[23,26,48,5...]
6.0	(1000,[19,23,47,5...]
6.0	(1000,[13,26,42,7...]
6.0	(1000,[20,26,35,7...]
6.0	(1000,[19,23,26,7...]
6.0	(1000,[16,50,73,7...]
6.0	(1000,[5,16,20,26...]
6.0	(1000,[9,26,34,42...]
6.0	(1000,[50,78,203,...]

Figure 13: Siniflandirma icin kullanılan özellikler

Figure 14’de goruldugu gibi veri seti; train %70 ve test %30 olacak sekilde bolunmustur.

```
%pyspark
#VERI SETINI BOLUYORUZ
splits = featured_data.randomSplit([0.7, 0.3])
train = splits[0]
test = splits[1]
train_rows = train.count()
test_rows = test.count()
print "Training Rows:", train_rows, " Testing Rows:", test_rows
```

Figure 14: Veri setinin bolunmesi

Figure 15’de decision tree classifier algoritmasinin kullanimi gosterilmistir.

```
%pyspark
#SINIFLANDIRMA-1
dt = DecisionTreeClassifier(labelCol="label", featuresCol="features")
dt_model = dt.fit(train)
```

Figure 15: Decision tree classifier kullanimi

Figure 16’da logistic regression algoritmasının kullanımı gösterilmistir.

```
%pyspark
#SINIFLANDIRMA-2
lr = LogisticRegression(maxIter=20, regParam=0.3, elasticNetParam=0,labelCol="label", featuresCol="features")
lrModel = lr.fit(train)
lr_predictions = lrModel.transform(test)
```

Link 1 can 1 set updated by admin at August 26 2019 3:16:54 PM

Figure 16: Logistic regression kullanımı

Figure 17’de naive bayes algoritmasının kullanımı gösterilmistir.

```
%pyspark
#SINIFLANDIRMA-3
from pyspark.ml.classification import NaiveBayes
nb = NaiveBayes(smoothing=1)
model = nb.fit(train)
predictions = model.transform(test)
```

Figure 17: Naive bayes algoritmasının kullanımı

Figure 18’de decision tree algoritmasının kullanımı gösterilmistir.

```
%pyspark
#SINIFLANDIRMA-4
from pyspark.ml.classification import RandomForestClassifier
rf = RandomForestClassifier(labelCol="label", \
                             featuresCol="features", \
                             numTrees = 100, \
                             maxDepth = 4, \
                             maxBins = 32)
# Train model with Training Data
rfModel = rf.fit(train)
predictions = rfModel.transform(test)
```

Figure 18: Decision tree algoritmasının kullanımı

Figure 19’da, siniflandırma algoritmalarına ait doğruluk oranının nasıl alındığı gösterilmiştir.

```
%pyspark
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
evaluator = MulticlassClassificationEvaluator(predictionCol="prediction")
evaluator.evaluate(lr_predictions)

0.8841856466100212
```

Figure 19: Siniflandirma algoritmalarinin basari sonucunu alma

7 Sonuclar

Bu bolumde calisma sonucunda elde edilen siniflandirma basarilarinin degerleri verilmiştir. Tablo 1’de count vectorizer oznitelik cikarimi yonteminin farkli numWords degerlerindeki dogruluk oranlari verilmistir(sonuclar minDF=5 ve kariyer.net 5 sinifli veri seti icindir).

Table 1: Count vectorizer kariyer.net 5 sinifli veri seti dogruluk oranlari minDF=5 icin

Model	10000	1000	500	100
LR	92 %	90 %	87 %	80 %
RF	85 %	83 %	81 %	79 %
NB	90 %	88%	85%	81 %
DTC	73 %	71 %	71 %	70 %

Table 2’de, count vectorizer oznitelik cikarimi yonteminin iki farkli numWords degerinde ve minDF=2 alinmis sonuclari bulunmaktadir(Sonuclar kariyer.net 5 sinifli veri seti icindir).

Table 2: Count vectorizer kariyer.net 5 sinifli veri seti dogruluk oranlari minDF=2 icin

Model	1000	100
LR	90 %	78 %
RF	81 %	78 %
NB	88%	81 %
DTC	72 %	72 %

Table 3’de, tf-idf oznitelik cikarimi yonteminin farkli numFeatures degerlerindeki siniflandirma basarilari verilmistir(Sonuclar kariyer.net 5 sinifli veri seti icindir).

Table 3: Tf-idf kariyer.net 5 sinifli veri seti dogruluk oranlari

Model	1000	500	100
DTC	67 %	65 %	52 %
LR	89 %	86 %	74 %
NB	86 %	83%	73 %
RF	79 %	75 %	65 %

Tablo 4’de count vectorizer oznitelik cikarimi yonteminin farkli numWords degerlerindeki dogruluk oranlari verilmistir(sonuclar minDF=5 ve kariyer.net 10 sinifli veri seti icindir).

Table 4: Count vectorizer kariyer.net 10 sinifli veri seti dogruluk oranlari minDF=5 icin

Model	10000	1000	500	100
DTC	52 %	51 %	51 %	48 %
LR	88 %	83 %	82 %	72 %
NB	87 %	82%	85%	76 %
RF	80 %	80 %	84 %	72 %

Table 5’de, tf-idf oznitelik cikarimi yonteminin farkli numFeatures degerlerindeki siniflandirma basarilari verilmistir(Sonuclar kariyer.net 10 sinifli veri seti icindir).

Table 5: Tf-idf kariyer.net 10 sinifli veri seti dogruluk oranlari

Model	1000	500	100
DTC	39 %	36 %	32 %
LR	81 %	75 %	66 %
NB	81 %	74%	61 %
RF	62 %	56 %	53 %