

AIS VERİLERİNİ VERİ MADENCİLİĞİ KULLANILARAK GEMİ TİPLERİNİN MANEVR A KABİLİYETLERİNE ETKİSİNİN ARAŞTIRILMASI INVESTIGATION OF THE EFFECT OF SHIP TYPES ON MANEUVER CAPABILITIES USING AIS DATA

Ergün Elvan Bilse¹,

1. Bilişim Sistemleri Enstitüsü
 Gazi Üniversitesi
 bilse¹ergun@gmail.com

ÖZETÇE

Gemiler Uluslararası ticaretin önemli bir aracıdır. Ticaretin devamlılığı sağlanması için gemi trafiğinin denetlenmesi ve kararlı hale getirilmesi gereklidir. Gelişen teknoloji ile gemi trafiği radar ve ais ile kolayca izlenebilmektedir. Ancak izlenebilirlik çatışmaların önüne geçememektedir. Bu yüzden gemi manevralarının izlenmesi ve gemi tiplerine göre analiz edilmesi gemi trafiğinin daha akılcı yönetilmesine olanak sağlayacağı görüşü benimsenmektedir. Gemi manevralarını izlemek ve gemi tipleriyle arasındaki bağı kurabilmek için veri madenciliğinden yararlanılarak bir analiz çalışması yapılmıştır.

ABSTRACT

Ships are an important tool of international intermediary trade. Trade must be toured. With the developing technology, ship traffic can be followed by radar and ais. However, traceability cannot prevent conflicts. For this reason, monitoring ship maneuvers and analyzing them according to ship types is adopted because it will enable the ship's carriage to be managed more rationally. In order to monitor ship maneuvers and establish a connection with the ends of the ship, an analysis study has been carried out using data mining .

1. GİRİŞ

Tanım; AIS gemi konuşlu bir transponder sistemi olup geminin kimlik bilgisini, mevkisini, hızını ve diğer verileri, yakınındaki gemi ve sahil istasyonuna VHF radyo üzerinden sürekli olarak yayınlar.

AIS verileri; 2009 - Haziran 2020 takvim yıllarına ait ABD kıyı suları için kayıtları içeren bir web sitesinden temin edilmiştir. Kayıtlar, Universal Transverse Mercator (UTM) bölgesi tarafından bir dakikaya kadar filtrelendir ve sıkıştırılmış, aylık dosyalar halinde kullanıma sunulmuştur.

Ais verileri içerisinde geçen kısaltmalar;

MMSI: Bir Deniz Mobil Hizmet Kimliği, gemi istasyonlarını, gemi yer istasyonlarını, sahil istasyonlarını, sahil yer istasyonlarını ve grup çağrılarını benzersiz bir şekilde tanımlamak için bir radyo frekansı kanalı üzerinden dijital biçimde gönderilen dokuz basamaklı bir dizidir.

LAT: Latitude (Enlem) **LON:** Longitude (Boylam)

SOG: (Speed of Ground): Yere göre hız. Karadan bakıldığında geminin görünen hızı.

COG: (Course Over Ground): Yere göre rota. Geminin yere göre gittiği yönü gösterir. Mesela, akıntı tekneyi geri geri iterse, yere göre yön, rotanın tersi olur.

Heading: Pruva veya rota. Geminin pruvasının baktığı yön. Akıntı vs sebebiyle gittiği yönden farklı olabilir.

IMO: (International Maritime Organization) Uluslararası Denizcilik Örgütü numarası gemiler, kayıtlı gemi sahipleri ve yönetim şirketleri için benzersiz bir tanımlayıcıdır.

CallSign: Çağrı işareti, radyo haberleşmesinde kullanılan kullanıcıyı tanıtan bir kod.

VesselType:

- 60 Yolcu, bu tipteki tüm gemiler
- 70 Kargo, bu tipteki tüm gemiler
- 80 Tanker, bu türden tüm gemiler

Status:

- 0 Motor gücü ile yürütülen tekne (kısıtlama yok)
- 5 Demirde olan tekne

Draft: Su çekimi.

Amaç; Veri madenciliği tekniklerini kullanarak gemi tiplerine göre gemi manevra karakteristiğini ortaya çıkarmaktır. Bu sayede kanal veya boğaz geçişlerinde manevra kabiliyetleri gemi tipleri ile önceden ön görülerek daha stabil bir trafik

hattı oluşturulabilecektir. Kapsam; Aynı boyut ve karina yapısındaki gemilerin manevra kabiliyetlerine etki eden husular arasında gemi tiplerinin avantaj ve dezavantajlarının araştırılması.

Yöntem; Makalemizde yedi milyon satırlık veri seti python yazılım dilinin pandas kütüphanesi ile düzenlenmiş ve temizleme işlemlerinden geçmiştir. Veri setinde çıkan verileri görselleştirmek için matplotlib kütüphanesinden yararlanılmıştır. Gemilerin gittikleri yönlerin oranları ve gitmiş oldukları rota izleri bu sayede görselleştirilmiştir. Temizlenen ve kategorik şekilde ayrılan veri setleri tekrar birleştirilip eğitim ve test veri setleri oluşturulmuştur. Bu veri setleri çeşitli algoritmalarda test edilmiş ve en yüksek doğruluk payı sonucu olan karar ağacı sınıflandırılması kullanılmıştır.

Bulgular; AIS veri seti incelenerek yazılan üç adet makale incelenmiştir. Gemilerin sağlıklı manevra varsayımları için korelasyon matrisi ile gemilerin seyir esnasında drift oranları karşılaştırılmıştır. Aynı rotada seyreden gemiler son olarak bir veri setinde birleştirilerek hareket rotaları üzerinden sınıflandırmaya gidilmiştir.

2. KARAR AĞACI

Karar Ağacı sınıflandırma algoritması, veri miktarı, algoritmanın etkinliği ve mevcut belleğe göre seri veya paralel adımlarla yapılabilir. Seri ağaç, bir eğitim veri kümesi kullanılarak oluşturulan ikili ağaç olarak mantıksal bir modeldir. Kural değişkenleri kullanarak bir hedef değişkenin değerini tahmin etmeye yardımcı olur [7]. Hiyerarşik olarak düzenlenmiş kural kümelerinden oluşur. Gelecekteki bir örneğin mevcut önceden tanımlanmış sınıflarda sınıflandırıldığı ve gözlemleri birbirini dışlayan alt gruplara bölmeye çalıştığı bir karar prosedürünü temsil etmek için düz, yinelemeli bir yapıdır. Ağaçtaki her bölüm, orijinal veri kümesindeki bir veya daha fazla kayda karşılık gelir. En üstteki düğümler, kök düğüm ve verilen veri kümesindeki tüm satırları temsil eder [8].

Her bir düğüm, alt gruplar benzer anlamlı bölünmeye girmek için çok küçük olana veya daha fazla bölünerek istatistiksel olarak önemli hiçbir alt grup üretilmeye kadar alt düğümler üretir. Örneklemin bazı bölümleri büyük bir ağaçla sonuçlanabilir ve bazı bağlantılar aykırı değerler veya yanlış değerler verebilir. Bu tür dalların kaldırılması gerekir. Ağaç budama, modelin doğruluk oranını önemli ölçüde etkilemeyecek şekilde yapılmalıdır.

3. LİTERATÜR TARAMASI

AIS verilerinin veri madenciliğinde kullanmasını araştıran 3 makale bulunmuş ve incelenmiştir.

Makale İsmi	Yazarlar	Üniversite	Kullanılan Metod
Vessel track information mining using AIS data	Feng Deng, Sitong Guo , Yong Deng , HanyueChu, Qingmeng Zhu and Fuchun Sun	University of Chinese Academy of Sciences	Fp-Growth Algoritması ve Markov modeli
Ship Traffic Flow Prediction Based on AIS Data Mining	Jiadong LI,Xueqi LI,Lijuan YU	Wuhan University of Technology Wuhan	Kübik Spline İnterpolasyonu ve RBF Sinir Ağı Modeli
AIS Database for Maritime Trajectory Prediction and DataMining	Shangbo Mao, Enmei Tu, Guanghao Zhang2, Lily Rachmawati3, Eshan Rajabally, Guang-Bin Huang	Nanyang Technological University Singapore	Extreme Learning Machine (ELM)

3.1. Vessel track information mining using AIS data

Ekonomi geliştikçe, su yolu taşımacılığının ve Uluslararası Ticaretin artması deniz trafiğinin yükselmesine neden olur. Çin ile ilgili uluslararası ticaretin %90'ından fazlasının deniz yoluyla gerçekleştirildiği düşünüldüğünde, Çin için nakliye analizi ve denizcilik gözetimi büyük önem taşımaktadır. Geleneksel olarak, geminin izini sürmek ve deniz durumunun tespiti radar, kızılötesi vb. Ekipmanlarla yapılmaktadır. Günümüzde teknolojinin gelişmesiyle birlikte, geminin izini sürmek ve denizi gözetlemek için kendi kendini raporlama sistemi devreye girmiştir. En yaygın olarak uygulanan kendi kendini raporlama sistemi, Uluslararası Denizcilik Örgütü ve Denizde Can Güvenliği Uluslararası Sözleşmesi (SOLAS) tarafından sunulan Otomatik Tanımlama Sistemidir. Kargo gemileri, konteyner gemileri gibi sivil gemilerin çoğu, AIS cihazları ile donatılması zorunludur. Otomatik Tanımlama Sisteminin yaygın olarak uygulanmasıyla, büyük AIS verileri oluşturulur. AIS verilerinin enlem, boylam gibi

özelliklerini analiz ederek ,rotanın hareket modellerini çıkarabilir, hareket durumunu tahmin edebilir ve anormal hareket durumunun tespitini yapabilir.

3.2. Ship Traffic Flow Prediction Based on AIS Data Mining

Bu makale, büyük AIS verilerinin anormal olup olmadığını belirler, gürültü azaltma çalışmasını tamamlar ve kayıp verilerin yeniden yapılandırılmasını sağlamak için kübik spline interpolasyonunu kullanır. Temiz veri elde etme temelinde, geminin gözlem yüzeyine gelişinin düzenini saymak için bir ayırt edici fonksiyon oluşturulur, ve daha sonra, belirli bir günde farklı zaman dilimlerinde gözlem bölümünden geçen gemi trafik akışını modellemek için bir zaman serisi yöntemi kullanılır. Simülasyon deneyi, RBF sinir ağı modeliyle kapsamlı karşılaştırma yoluyla tahmin sonucunun rasyonelliğini doğrular ve denizcilik departmanının iyileştirilmiş yönetimi uygulaması için bir referans sağlar.

3.3. AIS DatabaseforMaritimeTrajectoryPrediction and DataMining

Otomatik Tanımlama Sistemi (AIS), gemideki alıcı-verici ve karasal ve / veya uydu baz istasyonlarının hareketini izler. AIS tarafından toplanan veriler, yayın kinematik bilgilerini ve statik bilgileri içerir. Her ikisi de denizcilik istihbaratında anahtar teknikler olan deniz anomalisi tespiti ve gemi rotası tahmini için kullanışlıdır. Bu makale, denizde yörünge öğrenme, tahmin ve veri madenciliği için standart bir AIS veri tabanı oluşturmaya ayrılmıştır. Extreme Learning Machine (ELM) tabanlı bir yol tahmin yöntemi bu AIS veri tabanında test edilmiştir ve test sonuçları, bu veri tabanının farklı yörünge tahmin algoritmaları ve diğer AIS veri tabanlı madencilik uygulamaları için standartlaştırılmış bir eğitim kaynağı olarak kullanılabileceğini göstermektedir.

4. İNCELENEN MAKALELERDE UYGULANAN YÖNTEMLER

4.1. FP-Growth algoritması

FP-Growth algoritması iki aşamadan oluşur: FP Ağacı'nın yapımı ve FP Ağacı'ndan sık kullanılan kalıpların çıkarılması. FP-Ağacı'nın yapısı veritabanı üzerinde iki tarama gerektirir. İlk tarama, F listesinin oluşturulması için daha sonra azalan sıraya göre sıralanan sık öğeleri seçer. İkinci tarama, FP-Ağacı' nı oluşturur. Öncelikle, işlemler sık olmayan öğeler kaldırılarak F listesine göre yeniden sıralanır. Ardından yeniden düzenlenen işlemler FP Ağacı'na eklenir. FP-Growth girişi, FP-Ağacı ve minimum destek sayısıdır. FP-Growth, FP-Ağacı'ndaki düğümleri F listesinde en az bulunan öğeden ayırır. Her bir düğümü ziyaret ederken, FP-Growth, yoldaki öğeleri düğümden ağacın köküne toplar. Bu öğeler, o maddenin koşullu kalıp tabanını oluşturur. Koşullu kalıp tabanı, öğe ile birlikte meydana gelen küçük bir desen veritabanıdır. Sonra FP-Growth oluşturulur. Koşullu desen tabanından küçük FP-Ağacı ve FP Ağacı üzerinde FP-Growth yürütülür. Koşullu desen tabanı oluşturulmadan işlem tekrarlamalı olarak yinelenir [1].

4.2. Markov Karar Süreci

Markov süreci, şu anda meydana gelen bir olayın gelecekteki durumu hakkında olasılıklı bilgiler edinmeyi sağlayan bir yöntemdir. Markov analizinde önceki durumlardan bağımsız olarak, yalnızca mevcut duruma bağlı olan sürecin, nasıl gelişeceğini içeren olasılıkları bulunduran bir özelliği bulunmaktadır [2].

Süreçlerin çoğu uygulamada bu tanıma uymakta ve dolayısıyla da bu süreçlere Markov analizi olasılık modeli uygulanabilmektedir. Markov süreci uygulamalarında, optimal bir sonuca ulaşmak yerine karar vermeye yardımcı olabilecek olasılıklı bilgiler sağlama amacı güdülür.

4.2.1. Saklı Markov Modeli

Markov zinciri modellerinde sistemin, bir durumdan diğer duruma geçişi söz konusudur. Sistemin bulunabileceği bu durumlar ve durumlar arası geçişler açık bir şekilde gözlemlenebilir durumda ise Markov zinciri söz konusudur. Saklı Markov Modelinde ise durumlar dışarıdan doğrudan gözlemlenemez, yalnızca her bir durumdan meydana gelen gözlem çıktıları gözlemlenebilir [3]. Gözlem çıktılarının bir araya gelmesi ile gözlem dizisi meydana gelmektedir.

4.3. Kübik Spline İnterpolasyonu

Kübik spline interpolasyonu: (x_k, y_k) ($k = 0(1)N$) noktaları verildiğinde bu noktalardan geçen eğriyi bulma işlemidir.[4]

4.4. RBF

Radyal temelli Fonksiyonlar (RBF) Radyal temelli fonksiyon ağı tasarımı ise çok boyutlu uzayda eğri uydurma yaklaşımıdır ve bu nedenle RBF'nin eğitimi, çok boyutlu uzayda eğitim verilerine en uygun bir yüzeyi bulma problemine dönüşür.

Radyal Tabanlı Yapay Sinir Ağları (RTYSA), biyolojik sinir hücrelerinde görülen etkileşimli davranışlarından esinlenilerek 1988 yılında geliştirilmiş ve filtreleme problemine uygulanarak YSA tarihine girmiştir. RTYSA modellerinin eğitimi çok boyutlu uzayda eğri uydurma yaklaşımı olarak görmek mümkündür.[5]

4.5. Extreme Learning Machine

Extreme learning machine (ELM), tek gizli katman ileri beslemeli sinir ağları için yeni bir öğrenme algoritmasıdır. Geleneksel sinir ağı öğrenme algoritması ile karşılaştırıldığında, yavaş eğitim hızının ve aşırı uyum sorunlarının üstesinden gelir.[6] ELM, ampirik risk minimizasyon teorisine dayanmaktadır ve öğrenme süreci sadece tek bir yinleme gerektirir. Algoritma, birden çok yinlemeyi ve yerel en aza indirmeyi önler. Daha iyi genelleme yeteneği, sağlamlığı ve kontrol edilebilirliği ve hızlı öğrenme oranı nedeniyle

5. AIS

Modern küreselleşmiş ekonomide, okyanus taşımacılığı, malların uzun mesafelerde taşınması için en verimli yöntem haline geliyor. Dünya ekonomisinin sürekli büyümesi, daha büyük gemi kapasitesi ve daha yüksek seyir hızı ile artan deniz taşımacılığı talebine yol açmaktadır. Bu durumda, güvenlik ve emniyet, deniz taşımacılığında kilit konular haline gelir. Otomatik Tanımlama Sistemi (AIS) verilerini kullanan akıllı deniz seyrüsefer sistemi, geleneksel deniz seyrüsefer sistemine kıyasla daha az maliyetle deniz güvenliğini artırır. AIS, Uluslararası Denizcilik Örgütü (IMO) tarafından uygulanan bir deniz güvenliği ve gemi trafik sistemidir. Otonom olarak yayınlanan AIS mesajları kinematik bilgileri (gemi konumu, hızı, yönü, dönüş oranı, varış yeri ve tahmini varış zamanı dahil) ve statik bilgileri (gemi adı, gemi MMSI kimliği, gemi tipi, gemi boyutu ve güncel saat dahil) içerir. Gemi yolu tahmini ve çatışmadan kaçınma gibi akıllı deniz trafiği manipülasyonları için yararlı bilgilere dönüştürülebilir ve böylece gelecekteki otonom deniz seyrüsefer sisteminde merkezi bir rol oynar. Son birkaç yıldır, gemilerden ve kıyı istasyonlarından AIS mesajları almak giderek daha sıradan hale gelmektedir.

5.1. AIS Veri Tabanı Yapısı

Bu bölüm, veri işleme aracını ve önerdiğimiz standart AIS veri tabanını oluşturmanın ayrıntılarını açıklamaktadır. Tüm süreç dört bölümden oluşur:

- Ham veri ön işleme.
- Ham veri seçimi.
- Aday veri temizleme.
- Eksik veri enterpolasyonu.

5.1.1. Ham Veri Ön İşleme

Bir AIS veri tabanı oluşturma'nın ilk adımı, ham veri tabanı dosyasını csv formatında <http://www.marinecadastre.gov/ais/> adresinden indirmektir. AIS verilerini önceden işlemek ve faydalı verileri seçmek için, csv format dosyası tablo veri kayıtlarının satırlarından oluştuğundan, pandas kütüphanesi kullanılmıştır. Yapılan işlemlerini kayıt altına alabilmek için çeşitli programlama dilleri için etkileşimli bir ortam sağlayan açık kaynak kodlu bir program olan Jupyter Notebook, kullanılmıştır. Şekil-1 de ais bilgilerinin işlenmemiş hali görülmektedir.

	MMSI	BaseDateTime	LAT	LON	SOG	COG	Heading	VesselName	IMO	CallSign	VesselType	Status	Length	Width	Draft	Cargo	Transci
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	True	False	False	True	True	True
2	False	False	False	False	False	False	False	False	False	False	False	True	False	False	True	True	True
3	False	False	False	False	False	False	False	False	True	False	True	False	False	False	False	False	True
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...
7095048	False	False	False	False	False	False	False	False	False	False	False	False	False	False	True	False	False
7095049	False	False	False	False	False	False	False	False	False	False	False	False	False	False	True	False	False
7095050	False	False	False	False	False	False	False	False	True	False	False	False	False	False	True	False	False
7095051	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
7095052	False	False	False	False	False	False	False	False	True	False	False	False	False	True	True	True	True

7095053 rows x 17 columns

Şekil 1 Ham Veri

5.1.2. Ham Veri Seçimi

Gemiler öncelikle şekil-2 de örnek (tanker gemisi) gösterimi yapıldığı üzere gemi tiplerine göre sıralandırılmıştır, ardından şekil-3 de gösterildiği üzere MMSI ile tekrar sıralanmıştır. Bu sayede gemi tiplerine göre gemi mevcut sayılarına ulaşılmıştır. Bir MMSI, tek bir gemiyi temsil eder. Böylelikle her bir geminin izi kronolojik sırada görüntülenebilir ve işlenmesi daha kolay olur.

Tanker gemisi		MMSI	BaseDateTime	LAT	LON	SOG	COG	\
5		477669300	2019-01-10T00:00:00	29.74776	-95.28695	0.0	-157.0	
114		241287000	2019-01-10T00:00:00	25.24730	-79.71349	18.6	200.0	
144		371018000	2019-01-10T00:00:00	42.33532	-70.14363	11.4	-133.5	
290		366962000	2019-01-10T00:00:01	37.92224	-122.41079	0.1	25.2	
321		477669300	2019-01-10T00:00:00	29.74776	-95.28695	0.0	-157.0	
...	
7094813		366273000	2019-01-10T23:59:38	29.58951	-93.72920	0.0	-201.3	
7094959		538003776	2019-01-10T23:59:53	23.80487	-83.71423	13.2	-128.6	
7095001		352388000	2019-01-10T23:59:57	29.32197	-94.53635	0.1	142.0	
7095006		311000223	2019-01-10T23:59:58	30.10282	-88.44697	0.1	146.4	
7095034		248464000	2019-01-10T23:59:31	30.22867	-93.25300	0.0	-133.6	

Şekil 2 Tanker Tipi Gemi Kayıtları

5.1.3. Aday Veri Temizleme

Gemi tipleri belirlendikten sonra geminin hareket durumuna göre sıralama işlemi yapılır. Bu sayede hareket halindeki verilerine ulaşılır. Gemilerin tipleri ve hareket durumlarına göre filtrelendikten sonra hareketlerinin sağlıklı analiz edilebilmesi ve karşılaştırılması için başlangıç noktaları filtrelenmektedir. Şekil-3 de kanal girişini veri setinden çekmek için yazılan python sınıfı görülmektedir.

```

class Kanal_giris:
    def __init__(self,x):
        self.x = x
        long_x_max = self.long_max(self.x)
        long_x_min = self.long_min(long_x_max)
        lat_x_max = self.lat_max(long_x_min)
        self.response = self.lat_min(lat_x_max)

    def long_max(self,x):
        return x[x["LON"] < -94.70 ]

    def long_min(self, x):
        return x[x["LON"] > -94.80 ]

    def lat_max(self, x):
        return x[x["LAT"] > 29.334745 ]
    def lat_min(self, x):
        return x[x["LAT"] < 29.37200 ]

```

Şekil 3 Python Kanal Sınıfı

5.1.4. Eksik Değer Enterpolasyonu

AIS veri dosyamızda veri eksikliklerine neden olan süreksizlik, öğrenme algoritmalarının performansını ve veri tabanının veri madenciliği kalitesini etkileyebilir. Ayrıca ham veriler, hatalı hız verileri de içerir. Şekil-4 de veri setindeki boş veya anlamsız sütunların toplam veriye olan oranı bulunmuştur. Enterpolasyon yapmadan önce hatalı veriler silinmiştir.

Kayıp Oranı	
MMSI	0.000000
BaseDateTime	0.000000
LAT	0.000000
LON	0.000000
SOG	0.000000
COG	0.000000
Heading	0.000000
VesselName	4.997651
IMO	54.735546
CallSign	13.204214
VesselType	6.354230
Status	19.166495
Length	12.906486
Width	25.921089
Draft	71.899731
Cargo	65.295721
TranscieverClass	0.000000
dtype:	float64

Şekil 4 Kayıp Oranı

6. Veri Setine Uygulanan Temizleme İşlemi

6.1. Tanımsız Verilerin Silinmesi

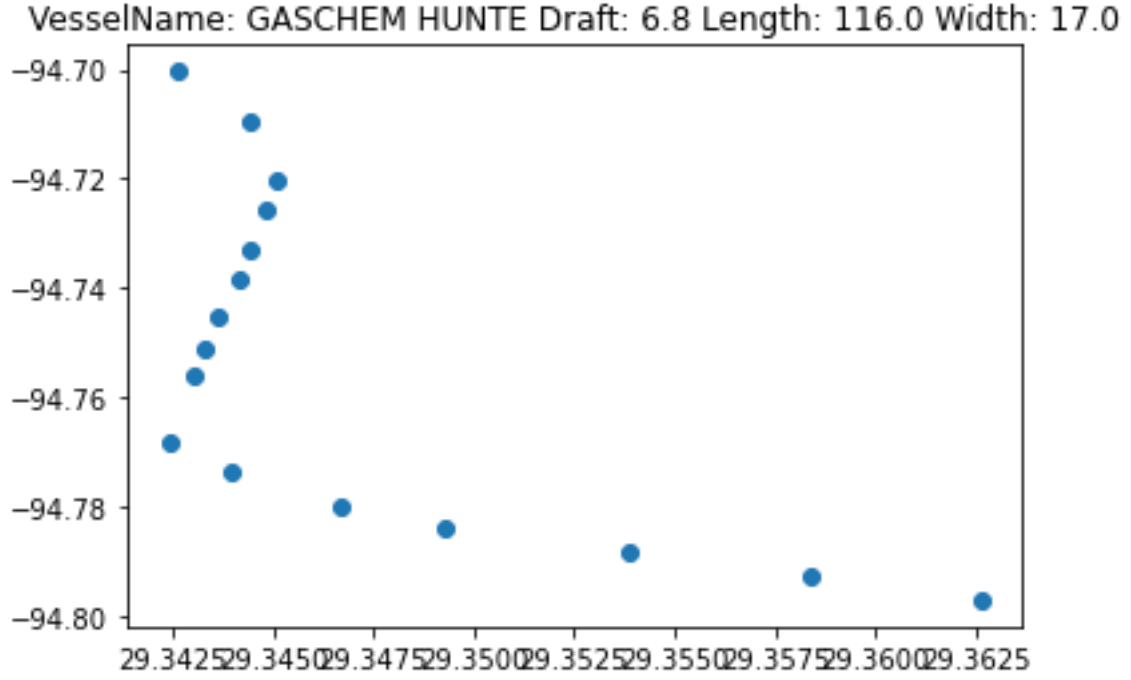
Veri setinin üzerindeki NaN, boşluk, NULL, undefined değerleri veri setinden çıkartılmıştır.

6.2. Merkezi Değer uygulaması

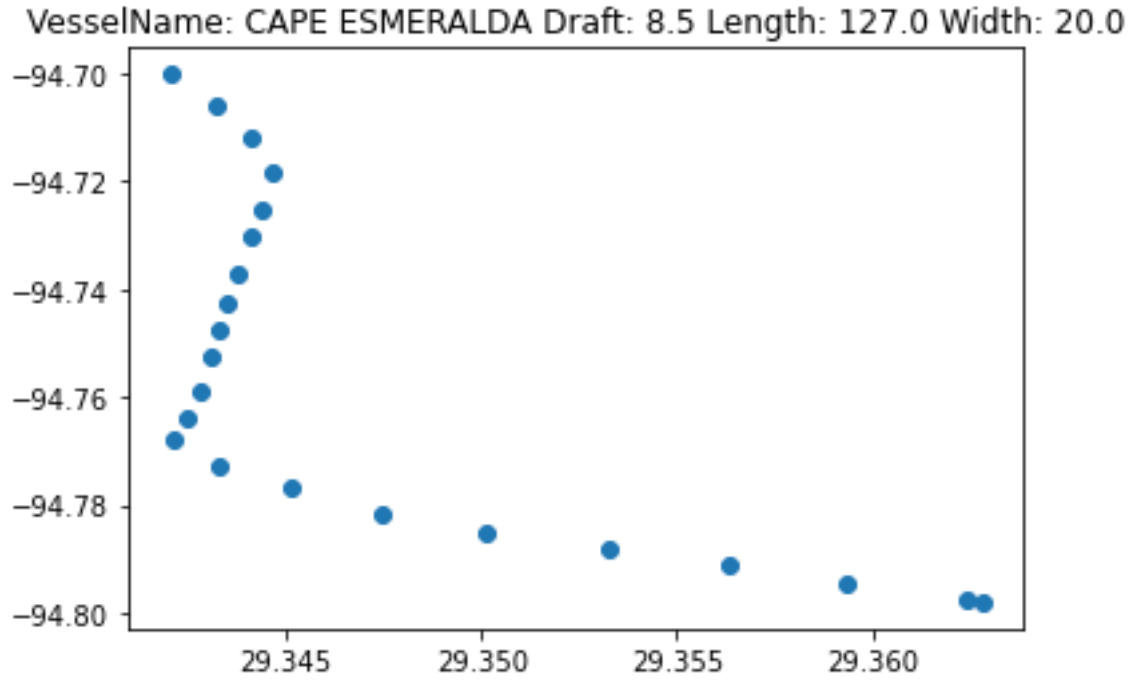
Merkezi değer atama , eksik değerlerin kendi merkezi eğilim ölçümleri, yani Ortalama , Medyan , Mod ile değiştirildiği bir yöntemdir . Ortalama veya Medyan kullanan sayısal değişkenler için tercih edilirken kategorik değişkenler için Mod kullanılır. Bunun arkasındaki sebep, kategorik değişkenler için ortalama ve medyanın bir anlamı olmamasıdır, çünkü kategorik değişkenlerin nicel özelliklerden ziyade nitel özellikleri vardır. Bu nedenle, merkezi eğilimi hesaba katmak için, Modu en sık ortaya çıkan değer olduğu için kullanılır.

7. DENEYSEL ÇALIŞMA

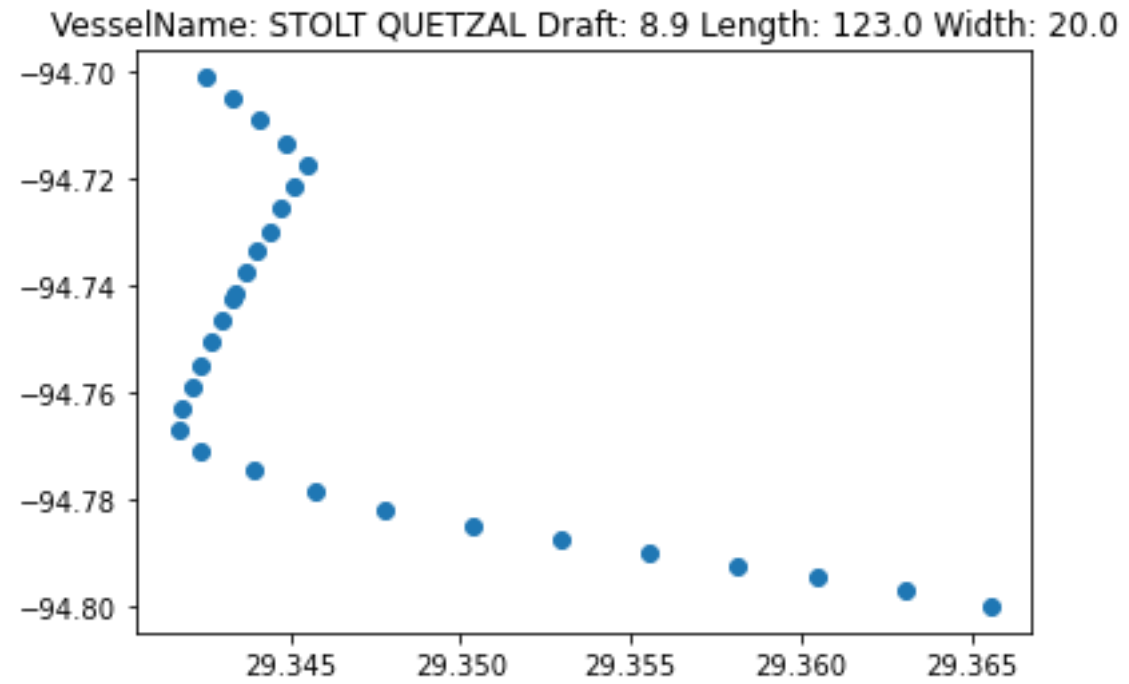
Kapsam kısmında bahsettiğimiz verileri pandas kütüphanesi yardımı ile önce gürültülü ve işlemimize olacak tanımsız değerlerden arındırdık. Veri setimizi sadece tanker ve kargo gemileri olacak şekilde ayrıştırdık. Ayrıştırdığımız bu veri setlerini Status Durumu 0 değeri olan gemiler şeklinde tekrar ayrıştırdık. Bu veri setini ise belirlediğimiz liman sınırları içerisinde tekrar ayrıştırıp veri setini birleştirdik. Birleştirdiğimiz veri setini tekrar draft ve gemi boylarına göre ayrıştırdık. Veri setinde bulunan her bir tanker ve kargo gemilerinin çizdiği rotayı oluşturduk. Oluşturduğumuz bu rotaları benzerliklerine göre ayrıştırdık ve aynı rota iz düşümü olan gemileri karar ağacı ile sınıflandırmaya çalıştık. Rotaları benzer olan tanker tipi gemiler özellikleri şeklin üst tarafında olacak şekilde; şekil-5, şekil-6, şekil-7, şekil-8, olarak tanımlanmıştır.



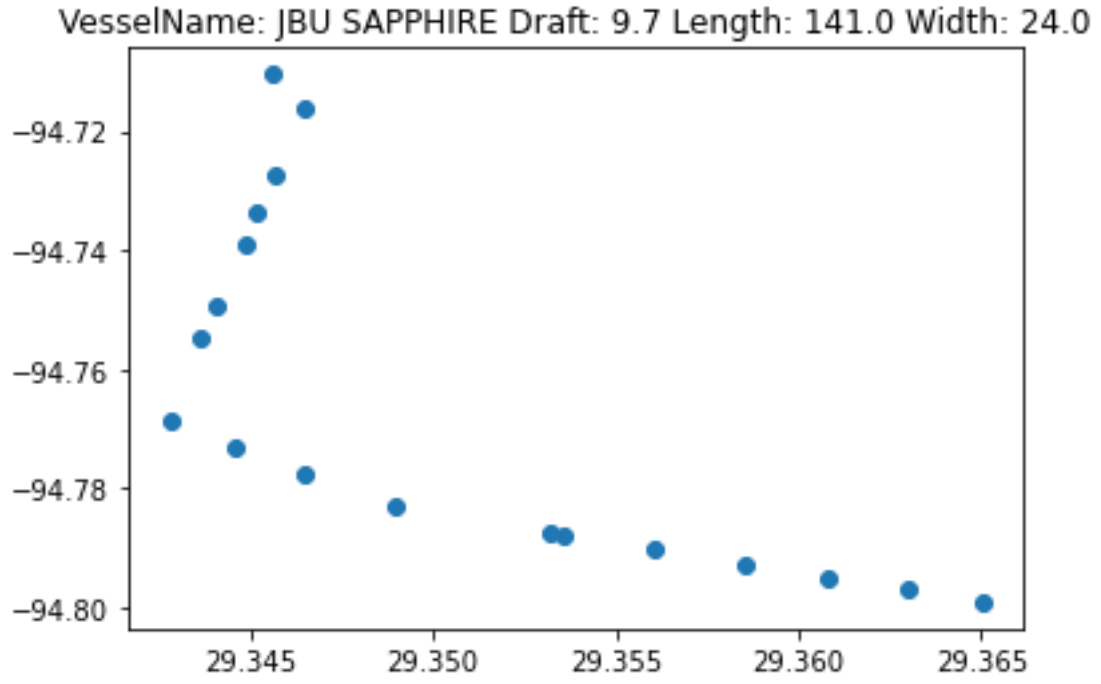
Şekil 5 M/T Gaschem Hunte Gemisi



Şekil 6 Cape Esmeralda Gemisi

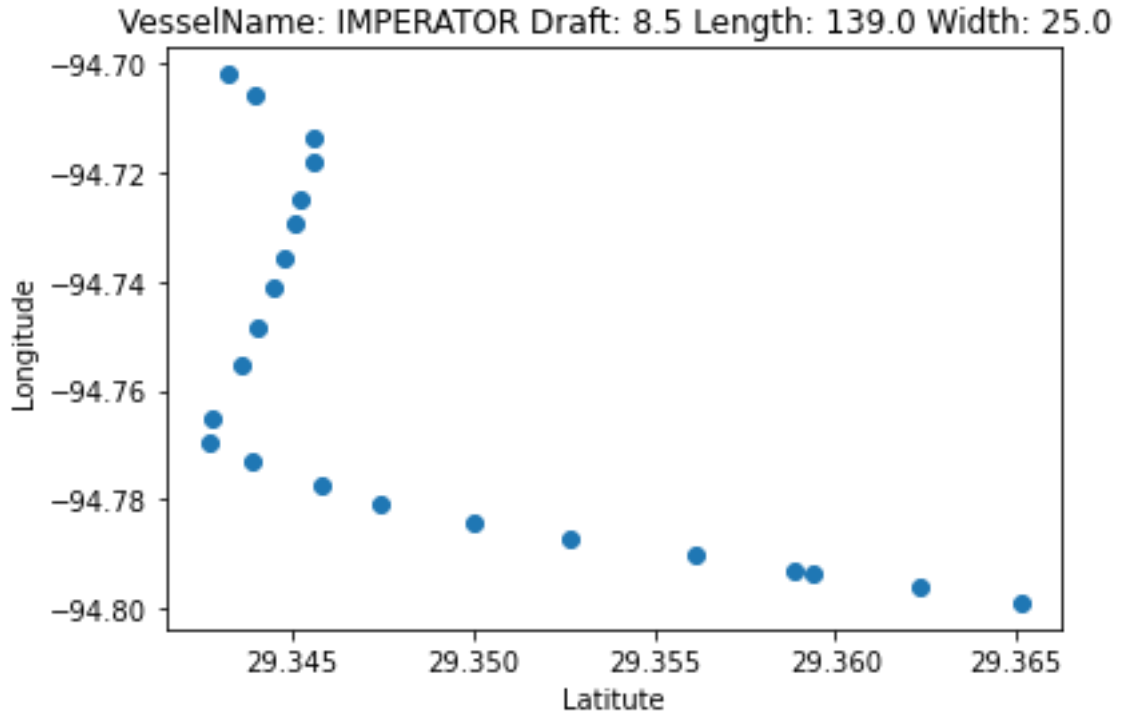


Şekil 7 M/T Stolt Quetzal Gemisi

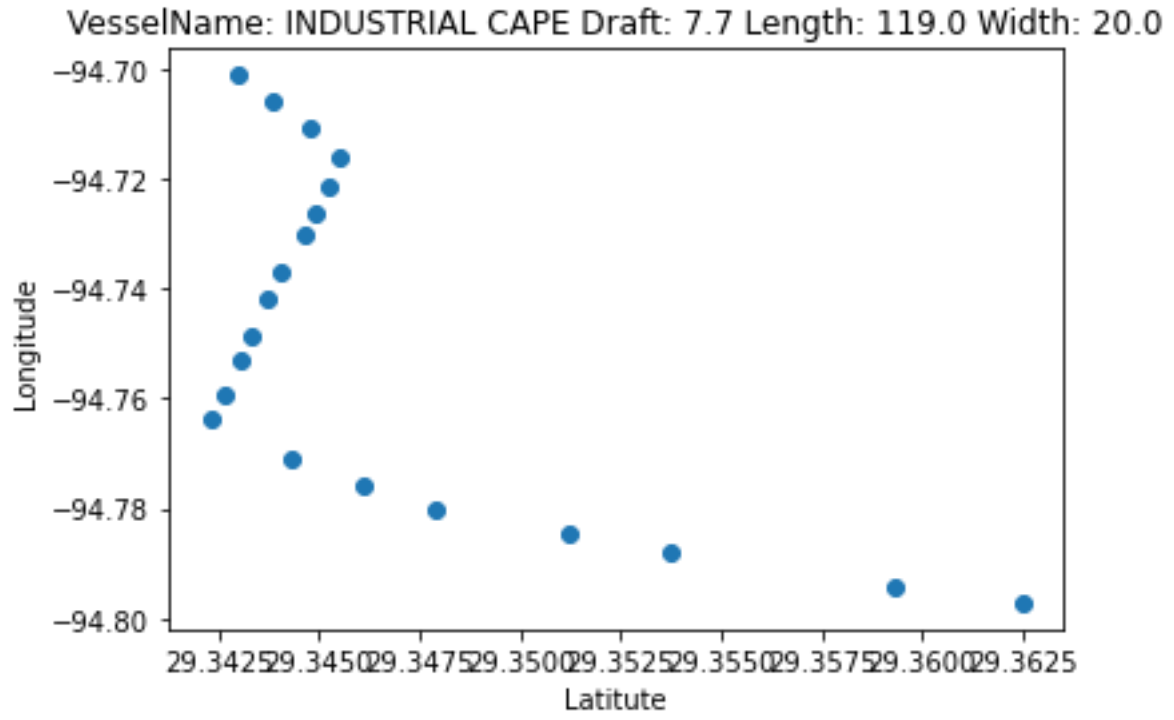


Şekil 8 M/T JBU SAPPHIRE Gemisi

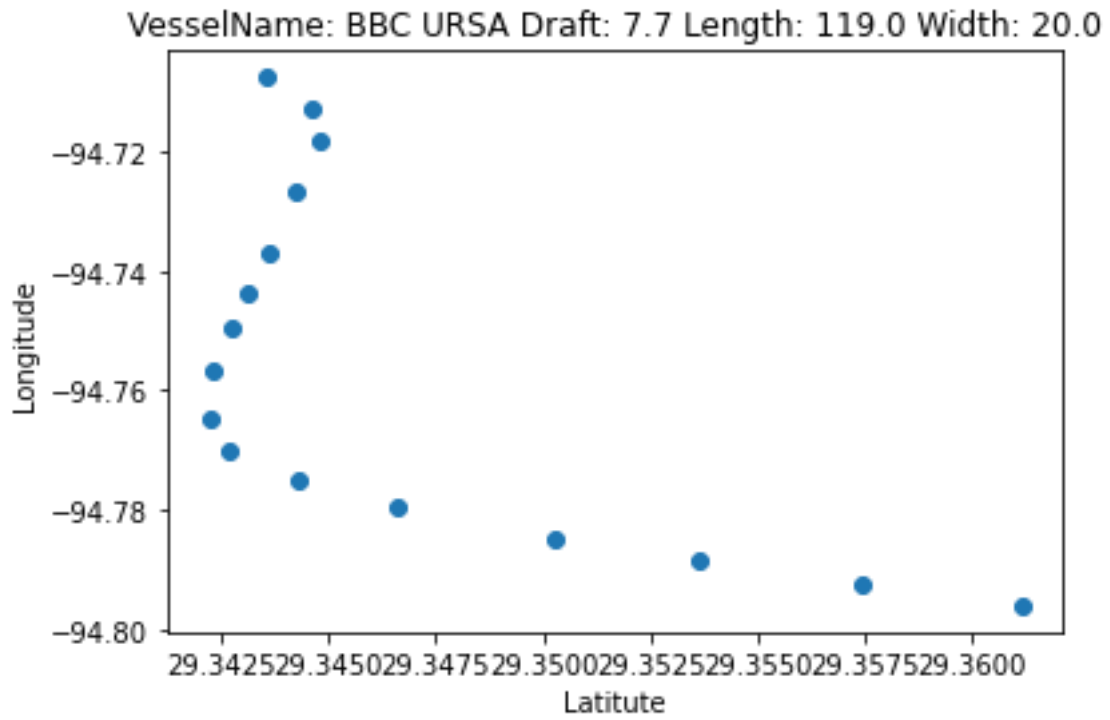
Rotaları benzer olan kargo tipi gemiler özellikleri şeklin üst tarafında olacak şekilde; şekil-5, şekil-6, şekil-7, şekil-8, olarak tanımlanmıştır.



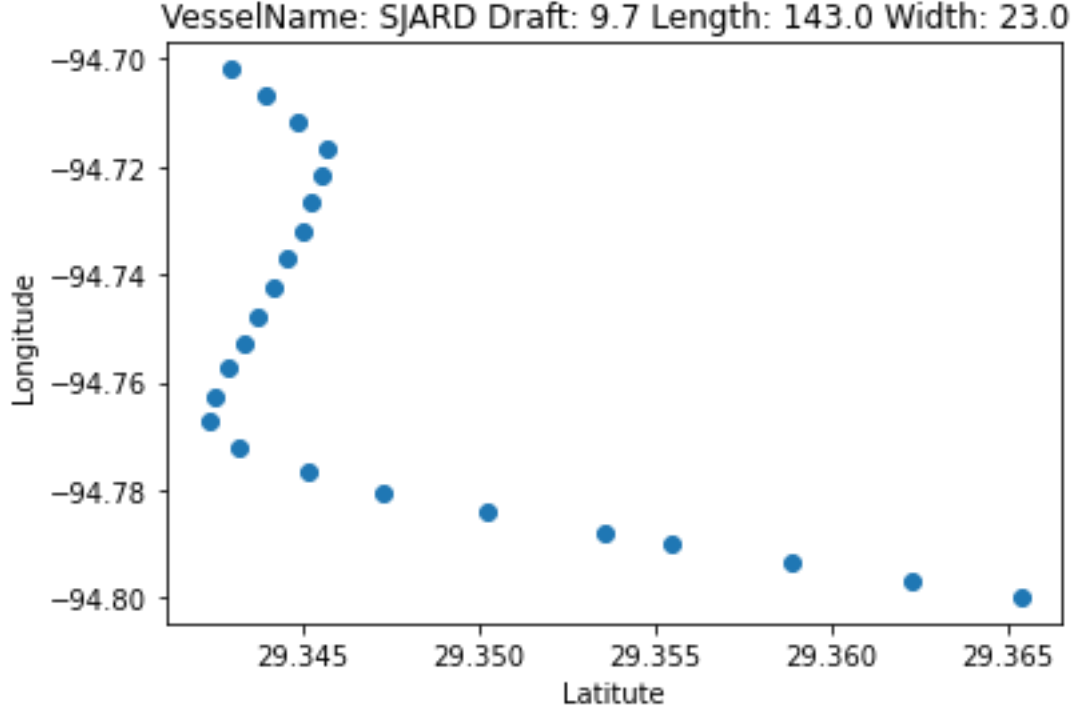
Şekil 9 M/V Imperator Gemisi



Şekil 10 M/V Industrial Cape Gemisi



Şekil 11 M/V BBC URSA Gemisi



Şekil 12 M/V SJARD Gemisi

7.1. Rota Metodu

Aşağıdaki şekilde görüldüğü gibi tüm kargo gemileri “data_tanker_name” kargo isimleri ile ayrıştırılarak değişkenlere atanacaktır. Bu değişken ship değişkeninin üzerine yazıldıktan sonra her bir geminin gemi ismi, draft bilgisi, gemi uzunluğu ve gemi eni bilgilerini de içeren rota iz düşümü haritası oluşturulacaktır. Aşağıda rotaları çizmede kullandığımız metodu görmekteyiz.

```
l = locals()
global ship
for cargo_name in uni_vessel_name(cargo_x_y):
    print(cargo_name)
    l['data_'+str(cargo_name)] = cargo_x_y[cargo_x_y['VesselName'] == cargo_name]
    ship = l['data_'+str(cargo_name)]
    plt.title('VesselName: ' + cargo_name + ' ' + 'Draft: ' + str(ship['Draft'][0:1].max()) + ' ' + 'Length: ' + str(ship['Length'][0:1].max()) + ' ' + 'Width: ' + str(ship['Width'][0:1].max()))
    plt.xlabel('Latitude')
    plt.ylabel('Longitude')
    plt.scatter(x=ship['LAT'], y=ship['LON'])
    plt.show()
```

Belirlenen liman limitleri, belirlenen 100 metreden büyük 150 metreden küçük maksimum su çekimi 9.8 metre ile 6.7 metre arasında olan ve üzerinde yol bulunan tanker gemilerinin rota iz düşümleri aşağıda görülmektedir.

7.2. Farklı Rotalar Seyreden Gemileri Temizleme İşlemi

Mevcut veri setimizden rota iz düşümleri uyuşmayan gemiler veri setinden silinmiştir. Şekil-13 de örnek olarak kargo gemileri veri setinden silinmiştir.

```
In [11]: ## Çıkartılacak gemiler cargo
print("BBC POLONIA, BBC NAPLES,INDUSTRIAL REVOLUTIO,NATIONAL GLORY")
BBC POLONIA, BBC NAPLES,INDUSTRIAL REVOLUTIO,NATIONAL GLORY
```

```
In [12]: cargo_clean.drop(cargo_clean.loc[cargo_clean['VesselName']=='NATIONAL GLORY'].index, inplace=True)
```

```
In [13]: cargo_clean.drop(cargo_clean.loc[cargo_clean['VesselName']=='BBC POLONIA'].index, inplace=True)
```

```
In [14]: cargo_clean.drop(cargo_clean.loc[cargo_clean['VesselName']=='BBC NAPLES'].index, inplace=True)
```

```
In [15]: cargo_clean.drop(cargo_clean.loc[cargo_clean['VesselName']=='INDUSTRIAL REVOLUTIO'].index, inplace=True)
```

```
In [16]: cargo_clean
```

```
Out[16]:
```

	Unnamed: 0.1	BaseDateTime	LAT	LON	SOG	COG	Heading	VesselName	tip
35	2156335	2019-01-04T07:31:58	29.34323	-94.70197	10.5	-127.6	283.0	IMPERATOR	cargo
36	2160554	2019-01-04T07:33:08	29.34398	-94.70587	10.6	-127.6	283.0	IMPERATOR	cargo
37	2174545	2019-01-04T07:35:28	29.34558	-94.71367	10.7	-125.6	273.0	IMPERATOR	cargo
38	2177912	2019-01-04T07:36:48	29.34557	-94.71805	10.5	-143.6	267.0	IMPERATOR	cargo
39	2185620	2019-01-04T07:38:48	29.34520	-94.72495	10.9	-142.6	268.0	IMPERATOR	cargo
...
250	4201154	2019-01-11T14:42:18	29.34900	-94.78235	11.4	-99.2	314.0	ONEGO TRADER	cargo
251	4223858	2019-01-11T14:43:37	29.35207	-94.78565	11.7	-92.7	317.0	ONEGO TRADER	cargo
252	4224934	2019-01-11T14:45:48	29.35739	-94.79112	11.7	-90.5	319.0	ONEGO TRADER	cargo
253	4226692	2019-01-11T14:47:07	29.36059	-94.79434	11.8	-91.5	319.0	ONEGO TRADER	cargo
254	4228625	2019-01-11T14:48:18	29.36348	-94.79730	11.8	-91.8	318.0	ONEGO TRADER	cargo

125 rows × 9 columns

Şekil 13 Silinen Kargo Gemiler

Şekil-14 de silinen tanker gemileri gösterilmektedir.

```
In [18]: tanker_clean.drop(tanker_clean.loc[tanker_clean['VesselName']=='CARIBE ILSE'].index, inplace=True)
```

```
In [19]: tanker_clean.drop(tanker_clean.loc[tanker_clean['VesselName']=='PGC ARATOS'].index, inplace=True)
```

```
In [20]: tanker_clean.drop(tanker_clean.loc[tanker_clean['VesselName']=='XENA'].index, inplace=True)
```

```
In [21]: tanker_clean.drop(tanker_clean.loc[tanker_clean['VesselName']=='STOLT FLAMENCO'].index, inplace=True)
```

```
In [22]: tanker_clean.drop(tanker_clean.loc[tanker_clean['VesselName']=='GOLDEN RAY'].index, inplace=True)
```

```
In [23]: tanker_clean.drop(tanker_clean.loc[tanker_clean['VesselName']=='BARBARICA'].index, inplace=True)
```

```
In [24]: tanker_clean.drop(tanker_clean.loc[tanker_clean['VesselName']=='NORDIC AKI'].index, inplace=True)
```

```
In [25]: tanker_clean
```

```
Out[25]:
```

	Unnamed: 0.1	BaseDateTime	LAT	LON	SOG	COG	Heading	VesselName	tip
0	659381	2019-01-03T02:23:51	29.34571	-94.71402	12.8	-134.3	267.0	SILVER RAY	tanker
1	666889	2019-01-03T02:21:51	29.34428	-94.70601	12.7	-126.5	283.0	SILVER RAY	tanker
2	670432	2019-01-03T02:27:11	29.34491	-94.72770	13.0	-144.2	267.0	SILVER RAY	tanker
3	670820	2019-01-03T02:25:00	29.34546	-94.71867	13.0	-144.1	266.0	SILVER RAY	tanker
4	677356	2019-01-03T02:29:10	29.34434	-94.73588	12.9	-144.4	266.0	SILVER RAY	tanker
...
724	5565013	2019-01-12T20:16:18	29.35050	-94.77803	11.4	-108.6	301.0	CARIBE MARIA	tanker
725	5565943	2019-01-12T20:17:29	29.35242	-94.78180	11.6	-111.6	301.0	CARIBE MARIA	tanker
726	5571727	2019-01-12T20:18:49	29.35447	-94.78620	11.6	-113.6	301.0	CARIBE MARIA	tanker
727	5582013	2019-01-12T20:19:58	29.35623	-94.78995	11.5	-107.6	310.0	CARIBE MARIA	tanker
728	5585046	2019-01-12T20:22:28	29.36173	-94.79617	11.2	-92.6	319.0	CARIBE MARIA	tanker

202 rows × 9 columns

Şekil 14 Silinen Tanker Gemileri

7.3. Verilerin Birleştirilmesi

Oluşturduğumuz ve temizlediğimiz verileri tek bir veri setinin içerisinde birleştirerek eğitim ve test verasetleri oluşturulur. Bu veri setleri ile mevcut algoritmaların doğruluk oranları hesaplanır. Mevcut doğruluk oranı en yüksek karar ağacı sınıflandırması sonucu çıkmıştır. Şekil-15 de test sonucunu görmekteyiz.

```
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC(gamma='auto')))
results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))

LR: 0.562963 (0.056475)
LDA: 0.543732 (0.064852)
KNN: 0.666239 (0.093695)
CART: 0.736182 (0.066293)
NB: 0.594160 (0.052026)
SVM: 0.689886 (0.080325)
```

Şekil 15 Algoritma Test Sonuçları

Şekilde görüldüğü üzere karar ağacı sınıflandırması bire en yakın değeri almaktadır. Veri setinde LAT, LONG, SOG ve Heading değerlerinden yararlanılarak bir karar ağacı oluşturulması sonucuna varılmıştır.

8. SONUÇLAR

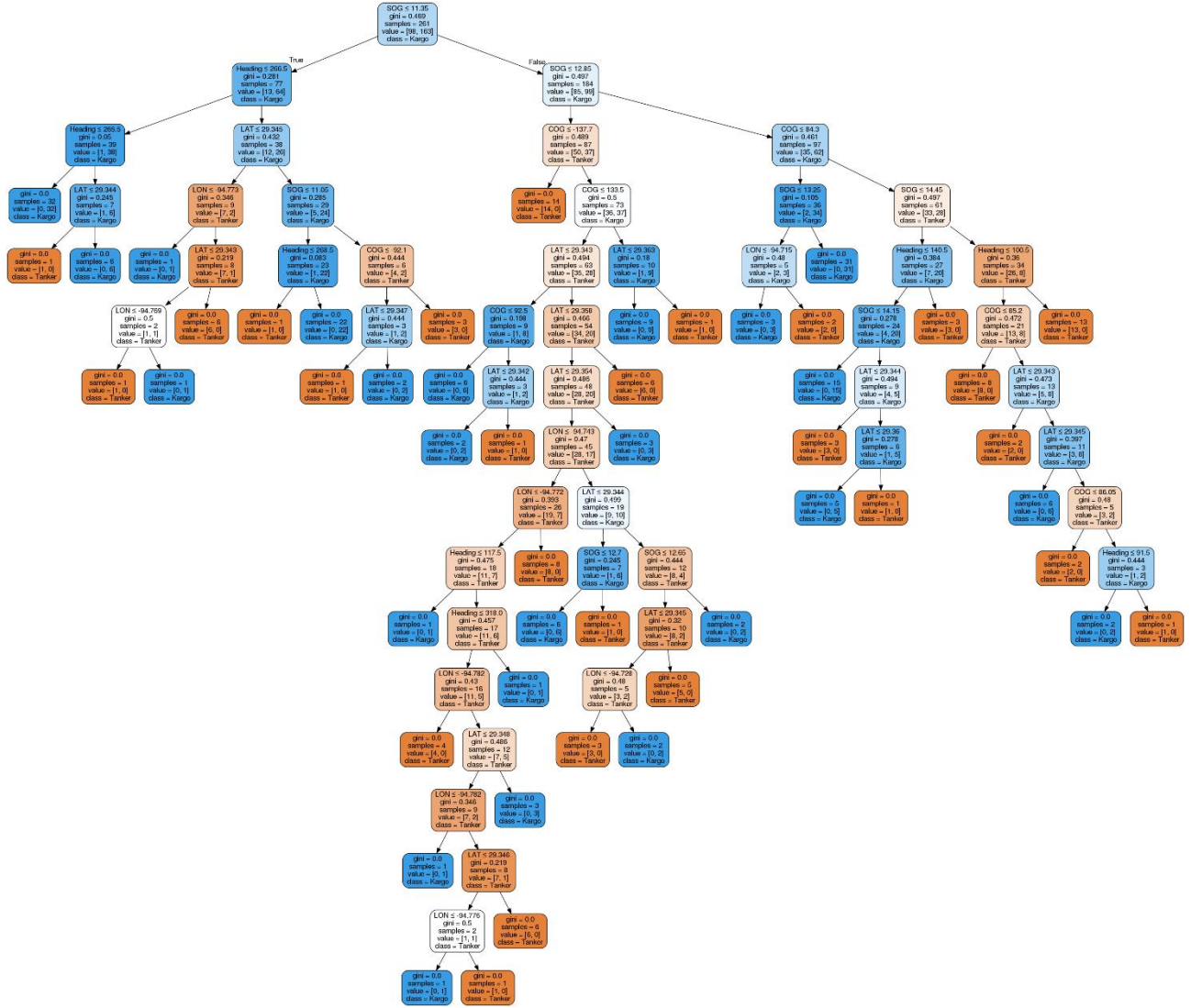
Veri madenciliği, ofis, hükümetler ve şirketler tarafından belirli amaçlarla eğilimleri tahmin etmek ve oluşturmak için kullanılan bir algoritmalar koleksiyonudur. Farklı yapı ve tiplerdeki gemilerin aynı rota üzerinde giderken oluşturdukları karakteristik özelliklerin karar ağacında yorumlanması için birleştirdiğimiz veri setini daha önce oluşturduğumuz eğitim veri setini kullanarak sınıflandırdık. Aşağıdaki şekilde karar ağacı fonksiyonunu görmekteyiz.

```
clf = DecisionTreeClassifier()
clf=clf.fit(X_train,Y_train)
y_pread = clf.predict(X_validation)
##Görselleştirme
feature_cols = ['LAT', 'LON', 'SOG', 'COG', 'Heading']
from sklearn.tree import export_graphviz
from six import StringIO
from IPython.display import Image
import pydotplus
dot_data = StringIO()
export_graphviz(clf, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True, feature_names = feature_cols, class_names=['Tanker', 'Kargo'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())
```

Şekil 16 Karar Ağacı Metodu

Çıkan sınıflandırmada oluşan ağacın budanarak ayrılan kollardan temizlenmesi gerekmektedir. Ağaç budama karmaşıklığı çıkan sonuçta çok fazladır. Optimum ağaç yapısı oluşturulmaya çalışılırken birden çok zor yapıda ve büyük ağaçlar oluşabilmektedir. Ağaç eğer bir tek veri kaydı kalana kadar büyütülürse, bu durum da birçok soru ve dal oluşturulmuş olmaktadır. Ancak karar ağacını bu denli büyütmek karmaşık ve büyük ağaçların oluşmasına neden olabilmektedir.

Maalesef karar ağacını oluşturduktan sonrada gemi tiplerinin manevraya etkisini etkileyecek somut verilere ulaşamadım. Şekil-17 dide karar ağacı sınıflandırmasını görmekteyiz.



Şekil 17 Karar Ağacı Sınıflandırması Şeması

9. KAYNAKÇA

- [1] FP-GROWTH ALGORİTMASI- WEKA UYGULAMASI, Öğr. Gör. Serpil SEVİMLİ DENİZ, Van Yüzüncü Yıl Üniversitesi
- [2] MARKOV ZİNCİRLERİ İLE PAZAR PAYI ARAŞTIRMA MODELİ VE OTOMOBİL LASTİĞİ PAZARINDA BİR UYGULAMA, Yavuz SOYKAN
- [3] Saklı Markov Modeli Kullanılarak İstanbul'daki Üniversite Öğrencilerinin GSM Operatör Tercihlerini Etkileyen Faktörlerin Analizi, Osman AYAZ1, Selçuk ALP, 1 Yıldız Teknik Üniversitesi, Makine Fakültesi, Endüstri Mühendisliği Bölümü 2018
- [4] CEBİRSEL KATSAYILI DİFERANSİYEL DENKLEMLERİN SPLİNE FONKSİYONU İLE ÇÖZÜMÜ Seval ÇATAL, DEU Fen Bilimleri Dergisi
- [5] Radyal Tabanlı Yapay Sinir Ağları ile Kemer Barajı Aylık Akımlarının Modellenmesi, Umut OKKAN, H.Yıldırım DALKILIÇ, İMO Teknik Dergi, 2012 5957-5966, Yazı 379
- [6] Extreme learning machine: algorithm, theory and applications, Shifei Ding•Xinzheng Xu•Ru Nic, Neural Comput,2013
- [7] N. Rahpeymai, Data Mining with Decision Trees in the Gene Logic Database: A Breast Cancer Study: Institutionen för datavetenskap, 2002.
- [8] Popular Decision Tree Algorithms of Data Mining Techniques: A Review, Abbas Alharan,Radhwan Alsagheer, 2017