**İnönü Üniversitesi**

Yazılım Gereksinimleri ve Analizi Dersi

Proje Ödevi

kedi, daire, ay, hafif içeren bir resim

Açıklama otomatik olarak oluşturuldu

Veriye Dayalı Satış Tahmini

**Data Sales Predictor**

**Tahir Tolu**

**İçindekiler**

1.Tanıtım........................................................................................................

1.1 Projenin Amacı............................................................................

1.2 Projenin Kapsamı.........................................................................

1.3 Genel Tanıtım..............................................................................

2.Çözümleme..................................................................................................

2.1 Sistem Ortamı...............................................................................

2.2 İşlevsel Gereksinimler Spesifikasyonu.........................................

2.2.1 Admin Kullanım Örnekleri.........................................................

2.2.2 Sistem Kullanım Örnekleri………………….......................

2.3 Kullanıcı Özellikleri....................................................................

2.4 İşlevsel Olmayan Gereksinimler……………………………….

3. Gereksinim Şartnamesi………….............................................................

3.1 Harici Arayüz Gereksinimleri……………….............................

3.2 İşlevsel Gereksinimler……………………….............................

3.2.1 Veri İmport Etme …………………………………........

3.2.2 Satış Tahmini ………….................................................

3.2.3 Hata Hesaplama …………………………………………..

3.2.4 Kriter Ve Ağırlık Belirleme ………………………………

3.2.5 Veri Puanlama …………………………………………….

3.2.6 Veri Temizleme …………………………………………...

3.2.7 Model Optimizasyonu ……………………………………………

4.Tasarım………………....................................................................................

4.1 Sistemin Tasarım Mimarisi...............................................................

4.2 Arabirimlerin ve Modüllerin Detayları.............................................

5.Gerçekleştirme………………………............................................................

5.1 Önerilen Sistemin Gerçekleştirilme Sürecindeki Ayrıntılar.............

5.2 Veri Tabanı Yönetim Sistemi Mimarisi............................................

5.2.1 Kullanıcı Arayüzü(User Interface)……………………….

5.2.2 Uygulama Katmanı(Application Layer)………………….

5.2.3 Veri Tabanı(Database)……………………………………

5.2.4 Kullanıcı Arayüzü

5.2.5 Uygulama Katmanı

5.2.6 Veri Tabanı Katmanı

5.3 Standartlar………………..................................................................

5.4 Olağan Dışı Durumlar ve Yönetimi...................................................

5.5 Kod Gözden Geçirme………………………………………………

6. Bakım………………………………………………………...........................

6.1 Yazılım Kurulum Aktiviteleri............................................................

6.2 Yerinde Destek Organizasyonu..........................................................

6.3 Kurulum ve Entegrasyon Aşamalarında Yapılacaklar........................

7. Sonuç……………………….............................................................................

7.1 Sonuç ve Değerlendirme……………………………..........................

7.2 Mevcut Sistemlerden Farkı………………………………..................

7.3 Hangi Probleme Çözüm Üretiyor…………………………................

7.4 Avantajlar ve Dezavantajlar………………………………………….

7.5 Gelecekte Nasıl İyileştirilebilir………………………………………

EK A: Sözlük..........................................................................................

EK B: Analiz Modelleri.........................................................................

8.Kaynakça.............................................................................................

**Şekiller:**

Şekil 1- Örnek bir mevcut sistemin use case diyagramı...……………..

Şekil 2- Örnek bir mevcut sistemin Admin Use Case diyagramı………

Şekil 3- Sistemin Akış Diyagramı……………………………………...

Şekil 4- Önerilen Sistem İçin Varlık İlişki Diyagramı(ERD)……….....

Şekil 5- Error Modülü Akış Diyagramı………………………...............

Şekil 6- Forecast Modülü Akış Diyagramı……………………………..

Şekil 7- Order Modülü Akış Diyagramı………………………………

Şekil 8-OrderDetail Modülü Akış Diyagramı………………………...

Şekil 9-Product Modülü Akış Diyagramı……………………………..

Şekil 10-Scoring Modülü Akış Diyagramı……………………………..

Şekil 11- Transaction Modülü Akış Diyagramı………………………..

Şekil 12- Realization Modülü Akış Diyagramı…………………………

**1.Tanıtım**

Data Sales Predictor, veri bilimi ve makine öğrenmesi prensipleri kullanarak bir işletmenin gelecekteki satışlarını tahmin etme amacı güden bir projedir. Sistemin geliştirilmesi ve implementasyonu, işletmelerin satış hedeflerini daha doğru ve verimli bir şekilde belirlemelerine, ayrıca operasyonel ve stratejik planlarını bu hedeflere göre ayarlamalarına yardımcı olabilir.

* 1. **Projenin Amacı:**

Data Sales Predictor projesinin ana amacı, işletmelerin gelecekteki satışlarını daha doğru bir şekilde tahmin etmelerini sağlamaktır. Bu,veri bilimi ve makine öğrenmesi prensiplerine dayanan bir süreç ile gerçekleştirilir. Satış tahminleri, işletmelerin hedeflerini belirlemelerine , operasyonel ve stratejik planlarını bu hedeflere göre ayarlamalarına yardımcı olur. İşletmeler böylece kararlarını daha bilinçli bir şekilde verebilirler.

* 1. **Projenin Kapsamı:**

Bu proje. Hem teknik(veri bilimi ve makine öğrenmesi yetenekleri) hem de işletme(satış tahminleme ve stratejik planlama) yönlerini kapsar. Sistem, bir veritabanının temizlenmesi, ağırlıklar ve kriterlerin belirlenmesi, skorlama kriterlerinin oluşturulması, bir modelin optimize edilmesi, satış tahminlerinin yapılması ve hata hesaplamalarının gerçekleştirilmesi gibi bir dizi adımdan oluşan bir süreç izler.

* 1. **Genel Tanıtım:**

Data Sales Predictor, makine öğrenmesi ve veri bilimi prensiplerini kullanarak işletmelerin gelecekteki satışlarını tahmin etme amacı güden bir projedir. Bu sistem, belirli bir süreci izler ve bir dizi sınıftan oluşur. Her sınıf belirli bir işlevi temsil eder ve birbirleriyle etkileşim içindedir. Bu sınıflar arasında Sipariş(Order), Sipariş Detay(OrderDetail),İşlem(Transaction),Ürün(Product),Skorlama(Scoring),Tahmin(Forecasting), Hata(Error), Arama(Search) ve Gerçekleşme(Realization) bulunur. Sistemin uygulanması, işletmelerin daha bilinçli ve bilgiye dayalı kararlar almasına yardımcı olmayı hedefler.

**2.0 Çözümleme**

**2.1 Sistem Ortamı**

metin, diyagram, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 1- Örnek mevcut sistemin Genel Use Case Diyagramı*

**Import Database :** Belirlenen kriterlere dayalı olarak , ilgili veritabanı sistemi kullanılarak gerekli veri seti içe aktarılır. Bu veri seti, satış tahmin modelinin eğitilmesi ve test edilmesi için kullanılacak veri setidir.

**Input Weights & Criteria Value:** Kullanıcı, satış tahmin modeli için belirli bir kriter seti ve ağırlıkları belirler. Bu kriterler ve ağırlıklar, ürünün satışı üzerinde en etkili olan özellikleri belirlemek için kullanılır. Bu değerlerin doğru bir şekilde belirlenmesi, tahmin modelinin doğruluğu için büyük önem taşır.

**Sales Forecasting:** Puanlanmış veri seti, satış tahmin modeline beslenir. Model, belirli bir süre boyunca satışları tahmin etmek için eğitilir. Model eğitildikten sonra, gelecek satışları tahmin etmek için kullanılır.

**Scoring Criteria:** İçe aktarılan veri seti, belirlenen kriterlere göre puanlanır. Her bir satış örneği, belirli kriterlere göre ayrı ayrı puanlanır. Bu adım, veri model tarafından daha etkili bir şekilde kullanılmasını sağlar.

**Error Calculation:** Son olarak, modelin tahminleri ile gerçek satış verileri karşılaştırılır ve hata hesaplanır. Bu adım, modelin performansını değerlendirmek için büyük önem taşır. Eğer hata kabul edilebilir bir seviyede değilse, model yeniden eğitilebilir veya kriterler ve ağırlıklar ayarlanabilir.

**Mevcut Sistemin Eksik Yönleri:**

* **Veri temizleme ve ön işleme :** Mevcut sistemde, veri setinin temizlenmesi için herhangi bir adım bulunmamaktadır. Veri setlerinde genellikle eksik, hatalı veya çelişkili bilgiler bulunabilir. Bu tür veri problemleri, modelin performansını olumsuz etkileyebilir ve yanıltıcı sonuçlara yol açabilir. ***Önerilen sistemdeki çözüm için*,** “Data Cleaning” adlı bir adım eklenmiştir. Bu adımda hatalı veya eksik verilerin düzeltilmesi, çelişkili verilen temizlenmesi ve veri setinin genel kalitesinin iyileştirilmesi hedeflenmektedir.
* **Model optimizasyonu:** Mevcut sistemde, modelin performansını optimize etmek için herhangi bir süreç bulunmamaktadır. Ancak, genellikle her makine öğrenmesi modeli, belirli bir parametre seti ile en iyi sonuçları verir. ***Önerilen sistemdeki çözüm için,*** “Model Optimization” adı altında bir adım eklenmiştir. Bu adımda, modelin parametreleri sistemli bir şekilde ayarlanır ve modelin tahmin performansını en üst düzeye çıkarmak için en iyi parametre seti belirlenir.

**2.2 İşlevsel Gereksinimler Spefikasyonu**

Bu bölümde adminin sistem üzerindeki kullanım durumları özetlenmiştir.

2.2.1 Admin Kullanın Örnekleri

Admin aşağıdaki kullanım durumlarına sahiptir.

metin, diyagram, çizgi, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 2- Örnek bir mevcut sistemin Admin Use Case Diyagramı*

Kullanım örneği: import database

**Şema:**

Admin

**Görev: Veri Seti İçeri Aktarma**

**Kısa Açıklama**

Admin, yeni bir veri setini sisteme yükler.

**Adım Adım Açıklama**

Bu kullanım senaryosunun başlatılabilmesi için önce Admin sisteme giriş yapmış olmalıdır.

1. **Admin, "Veri Seti Yükle" seçeneğini seçer.**
2. **Admin, "Dosya Seç" butonuna tıklar ve yüklemek istediği dosyayı bilgisayarından seçer.**
3. **Admin, "Yükle" butonuna tıklar.**.
4. **Sistem, dosyanın formatını ve içerik yapısını kontrol eder.**
5. **Sistem, yüklenen veri setinin özetini ve önizlemesini Admin'e sunar.**
6. **Admin, "Veri Setini Onayla" butonuna tıklar.**
7. **Sistem, Admin'i veri seti yönetim sayfasına yönlendirir.**

Kullanım örneği: input Weights & Vriteria Value

**Şema:**

Admin

**Görev: Kriter Seti ve Ağırlık Değerleri Girişi**

**Kısa Açıklama**

Kullanıcı, satış tahmin modeli için belirli bir kriter seti ve ağırlıkları belirler.

**Adım Adım Açıklama**

**1. Kullanıcı, "Kriter ve Ağırlık Ayarları" seçeneğini seçer.**

**2. Kullanıcı, "Yeni Kriter Ekle" butonuna tıklar.**

**3. Kullanıcı, kriterin adını ve açıklamasını girer.**

**4. Kullanıcı, kriterin ağırlık değerini belirler.**

**5. Kullanıcı, "Kaydet" butonuna tıklar.**

**6. Kullanıcı, eklenen kriterler ve ağırlıkları gözden geçirir ve gerektiğinde düzenler.**

**7. Kullanıcı, "Kriter Setini Onayla" butonuna tıklar.**

**8. Sistem, Kullanıcı'yı ana sayfaya yönlendirir.**

Kullanım örneği: Scoring Criteria

**Şema:**

Admin

**Görev: Puanlama Kriterleri**

**Kısa Açıklama**

İçe aktarılan veri seti, belirlenen kriterlere göre puanlanır. Her bir satış örneği, belirli kriterlere göre ayrı ayrı puanlanır. Bu adım, verinin model tarafından daha etkili bir şekilde kullanılmasını sağlar.

**Adım Adım Açıklama**

Bu kullanım senaryosunun başlatılabilmesi için önce Admin sisteme giriş yapmış ve veri setini içe aktarmış olmalıdır.

**1. Admin, "Veri Setini Puanla" seçeneğini seçer.**

**2. Admin, puanlama kriterlerini ve ağırlıklarını girer veya mevcut kriterlerden seçim yapar.**

**3. Admin, "Puanlama İşlemini Başlat" butonuna tıklar.**

**4. Sistem, her bir satış örneğini belirlenen kriterlere göre puanlar.**

**5. Sistem, puanlama sonuçlarını Admin'e sunar.**

**6. Admin, puanlama sonuçlarını inceler ve onaylar.**

**7. Sistem, puanlanmış veri setini eğitim için hazırlar ve ilgili sayfaya yönlendirir.**

**8. Admin, puanlanmış veri setini kullanarak diğer işlemleri gerçekleştirebilir.**

Kullanım örneği: Sales Forecasting

**Şema:**

Admin

**Görev: Satış Tahmin Modeli Eğitimi ve Kullanımı**

**Kısa Açıklama**

Puanlanmış veri seti, satış tahmin modeline beslenir. Model, belirli bir süre boyunca satışları tahmin etmek için eğitilir ve optimizasyon işlemlerinden geçer. Model eğitildikten sonra, gelecek satışları tahmin etmek için kullanılır.

**Adım Adım Açıklama**

Bu kullanım senaryosunun başlatılabilmesi için önce Kullanıcı sisteme giriş yapmış ve puanlanmış veri setini yüklemiş olmalıdır.

**1. Kullanıcı, "Modeli Eğit" seçeneğini seçer.**

**2. Kullanıcı, eğitimde kullanılacak veri setini seçer.**

**3. Sistem, veri seti üzerinde temizlik işlemi (data cleaning) yapar.**

**4. Sistem, veri setini satış tahmin modeline besler.**

**5. Sistem, modelin eğitim sürecini başlatır ve optimizasyon işlemlerini gerçekleştirir.**

**6. Sistem, eğitim sürecini tamamlar ve modelin performansını değerlendirir.**

**7. Kullanıcı, eğitilen modelin performans raporunu inceler ve onaylar.**

**8. Kullanıcı, "Gelecek Satışları Tahmin Et" butonuna tıklar.**

**9. Sistem, tahmin sonuçlarını kullanıcıya sunar.**

**10. Sistem, kullanıcıyı ana sayfaya yönlendirir.**

Kullanım örneği: Error Calculation

**Şema:**

Admin

**Görev: Hata Hesaplama**

Son olarak, modelin tahminleri ile gerçek satış verileri karşılaştırılır ve hata hesaplanır. Bu adım, modelin performansını değerlendirmek için büyük önem taşır. Eğer hata kabul edilebilir bir seviyede değilse, model yeniden eğitilebilir veya kriterler ve ağırlıklar ayarlanabilir.

**Adım Adım Açıklama**

Bu kullanım senaryosunun başlatılabilmesi için önce Kullanıcı sisteme giriş yapmış ve satış tahmin modeli eğitilmiş olmalıdır.

**1. Kullanıcı, "Hata Hesapla" seçeneğini seçer.**

**2. Kullanıcı, modelin tahmin sonuçlarını ve gerçek satış verilerini sisteme yükler.**

3**. Sistem, tahmin sonuçları ile gerçek satış verilerini karşılaştırır.**

**4. Sistem, hesaplanan hata değerlerini kullanıcıya sunar.**

**5. Kullanıcı, hesaplanan hata değerlerini inceler ve değerlendirir.**

**6. Eğer hata kabul edilebilir seviyede değilse, kullanıcı model optimizasyon veya yeniden eğitim seçeneklerini seçebilir.**

**7. Sistem, kullanıcı tarafından yapılan ayarlamaları uygular ve güncellenmiş modelle yeni tahminler yapar.**

**8. Kullanıcı, hata hesaplama sürecini tamamlar ve sistem ana sayfasına döner.**

**Mevcut Sistemin Eksik Yönleri ve Çözüm Önerileri**

Veri Temizleme ve Ön İşleme:

- Eksik Yön: Mevcut sistemde, veri setinin temizlenmesi için herhangi bir adım bulunmamaktadır. Veri setlerinde genellikle eksik, hatalı veya çelişkili bilgiler bulunabilir. Bu tür veri problemleri, modelin performansını olumsuz etkileyebilir ve yanıltıcı sonuçlara yol açabilir.

- Çözüm: "Data Cleaning" adımı eklenmiştir. Bu adımda hatalı veya eksik verilerin düzeltilmesi, çelişkili verilerin temizlenmesi ve veri setinin genel kalitesinin iyileştirilmesi hedeflenmektedir.

Model Optimizasyonu:

- Eksik Yön: Mevcut sistemde, modelin performansını optimize etmek için herhangi bir süreç bulunmamaktadır. Ancak, genellikle her makine öğrenmesi modeli, belirli bir parametre seti ile en iyi sonuçları verir.

- Çözüm: "Model Optimization" adımı eklenmiştir. Bu adımda, modelin parametreleri sistemli bir şekilde ayarlanır ve modelin tahmin performansını en üst düzeye çıkarmak için en iyi parametre seti belirlenir.

2.2.2 Sistem Kullanın Örnekleri

Kullanım örneği : Data Cleaning

sistem

**Görev: Veri Gürültüsü Temizliği ve Hata Hesaplama**

**Kısa Açıklama**

Sistem, veri seti üzerinde gürültü temizliği yapar ve ardından modelin tahminleri ile gerçek satış verilerini karşılaştırarak hata hesaplar. Veri temizliği, modelin performansını artırmak için büyük önem taşır.

**Adım Adım Açıklama**

Bu kullanım senaryosunun başlatılabilmesi için önce sistem, veri setini içe aktarmış olmalıdır.

**1. Sistem, veri gürültüsü temizleme sürecini başlatır.**

**2. Sistem, tespit edilen veri problemlerini temizler.**

**3. Sistem, temizlenmiş veri setini hazırlar.**

**4. Sistem, tahmin sonuçlarını ve gerçek satış verilerini karşılaştırır.**

**5. Sistem, hata hesaplama işlemini başlatır.**

**6. Sistem, hesaplanan hata değerlerini kullanıcıya sunar.**

**7. Kullanıcı, hesaplanan hata değerlerini inceler ve değerlendirir.**

**8. Eğer hata kabul edilebilir seviyede değilse, kullanıcı model optimizasyon veya yeniden eğitim seçeneklerini seçebilir.**

**9.Sistem, kullanıcı tarafından yapılan ayarlamaları uygular ve güncellenmiş modelle yeni tahminler yapar.**

**10. Kullanıcı, veri gürültüsü temizliği ve hata hesaplama sürecini tamamlar ve sistem ana sayfasına döner.**

Kullanım örneği:Model optimizasyon

sistem

**Görev: Model Optimizasyonu**

**Kısa Açıklama**

Modelin performansını optimize etmek için belirli parametreler ayarlanır. Bu adım, modelin tahmin doğruluğunu en üst düzeye çıkarmak için büyük önem taşır.

**Adım Adım Açıklama**

Bu kullanım senaryosunun başlatılabilmesi için önce sistem, veri temizleme işlemini gerçekleştirmiş ve model eğitilmiş olmalıdır.

**1. Sistem, model optimizasyon sürecini başlatır.**

**2. Sistem, modelin performansını artırmak için uygun optimizasyon tekniklerini belirler.**

**3. Sistem, modelin parametrelerini sistemli bir şekilde ayarlar ve yeniden eğitir.**

**4. Sistem, optimize edilen modelin performansını değerlendirir.**

**5. Sistem, optimize edilen modelin performans raporunu kullanıcıya sunar.**

**6. Kullanıcı, optimize edilen modelin performansını inceler ve onaylar.**

**7. Sistem, optimize edilen modeli gelecekteki tahminlerde kullanıma hazır hale getirir.**

**8. Kullanıcı, model optimizasyon sürecini tamamlar ve sistem ana sayfasına döner.**

**2.3 Kullanıcı Özellikleri**

- İnternet Okuryazarlığı:\*Admin, interneti etkin bir şekilde kullanabilmeli ve bir arama motorunda arama yapabilmelidir.

- Veri Analizi Bilgisi: Veri yükleme, veri temizleme, model eğitme, model optimizasyonu ve hata hesaplama işlemlerini gerçekleştirebilmesi için temel veri analizi ve makine öğrenmesi bilgisine sahip olmalıdır.

- Kullanıcı Arayüzü Yetkinliği: Sistemin kullanıcı arayüzünde kolayca gezinebilmeli, formları doldurabilmeli ve raporları inceleyebilmelidir.

- Veri Temizleme Yetkinliği: Admin, veri setindeki hatalı, eksik veya çelişkili verileri tespit edebilmeli ve temizleyebilmelidir.

- Model Eğitme ve Optimizasyon Bilgisi: Makine öğrenmesi modellerini eğitebilmeli ve performanslarını optimize edebilmelidir.

- Hata Hesaplama ve Değerlendirme: Modelin tahminleri ile gerçek satış verilerini karşılaştırarak hata hesaplayabilmeli ve sonuçları değerlendirebilmelidir.

- Stratejik Karar Verme: Model sonuçlarını yorumlayarak iş süreçlerine entegre edebilmeli ve stratejik kararlar alabilmelidir.

Bu özellikler, adminin sistemin tüm fonksiyonlarını etkin bir şekilde kullanabilmesi ve iş süreçlerine katkı sağlayabilmesi için gereklidir. Adminin bu yetkinliklere sahip olması, sistemin genel performansını ve doğruluğunu artıracaktır.

**2.4 İşlevsel Olmayan Gereksinimler**

1. Performans Gereksinimleri:

- Veri temizleme işlemi, büyük boyuttaki veri setleri üzerinde hızlı ve etkili bir şekilde gerçekleştirilmelidir.

- Model optimizasyonu süreci, optimize edilmiş parametrelerle birlikte makine öğrenmesi modelinin hızlı bir şekilde yeniden eğitimini sağlamalıdır.

2. Güvenlik Gereksinimleri:

- Sistem, kullanıcıların veriye yetkisiz erişimini önlemek için uygun güvenlik önlemlerine sahip olmalıdır.

- Veri temizleme ve model optimizasyonu işlemleri sırasında hassas verilerin güvenliği sağlanmalıdır.

3. Kullanılabilirlik Gereksinimleri:

- Kullanıcı arayüzü, kullanıcıların veri temizleme ve model optimizasyonu işlemlerini kolayca gerçekleştirebilecekleri kullanıcı dostu bir şekilde tasarlanmalıdır.

- Kullanıcılar, sistemi etkin bir şekilde kullanabilmek için gerekli eğitim ve kullanım kılavuzlarına erişebilmelidir.

4. Taşınabilirlik Gereksinimleri:

- Sistem, farklı işletim sistemleri üzerinde sorunsuz bir şekilde çalışabilmelidir (örneğin, Windows, Linux, macOS).

5. Belgelendirme Gereksinimleri:

- Kullanıcılar için detaylı kullanım kılavuzları ve sistem belgeleri sağlanmalıdır.

- Sistem yöneticileri için veri temizleme ve model optimizasyonu işlemleri için adım adım kılavuzlar sunulmalıdır.

6. Performans Gereksinimleri:

- Veri temizleme ve model optimizasyonu işlemleri, kabul edilebilir bir süre içinde tamamlanmalıdır.

- Sistem, yoğun kullanım durumlarında bile stabil bir performans sergilemelidir.

**3.0 Gereksinim Şartnamesi**

**3.1 Harici Arayüz Gereksinimleri**

Kullanıcı Arayüzü:

- Web Arayüzü: Kullanıcılar için anlaşılır ve kolay kullanılabilir bir web arayüzü olmalıdır. Kullanıcılar, veri setlerini yüklemek, kriter ve ağırlıkları belirlemek, model sonuçlarını görmek ve hata hesaplamalarını incelemek gibi işlemleri bu arayüz üzerinden gerçekleştirebilmelidir.

- Admin Paneli: Adminler için detaylı kontrol ve yönetim özelliklerine sahip bir yönetici paneli bulunmalıdır. Adminler, veri setlerini yükleyip yönetebilir, kriter ve ağırlıkları belirleyebilir, model eğitimini başlatabilir ve sonuçları gözden geçirebilirler.

Donanım Arayüzü:

- Sunucu: Sistem, yüksek performanslı bir sunucu üzerinde çalışmalıdır. Donanım gereksinimleri proje sahibi tarafından belirlenecektir. Bu sunucu, veri işlemleri ve model eğitimi gibi yoğun işlemleri destekleyecek kapasitede olmalıdır.

- Veritabanı Sunucusu: Yüksek hızlı erişim ve veri bütünlüğü için güvenilir bir veritabanı sunucusu kullanılmalıdır. Bu sunucu, büyük veri setlerini hızlı bir şekilde işleyebilmeli ve güvenli bir şekilde saklayabilmelidir.

Yazılım Arayüzü:

- Veritabanı Yönetim Sistemi (DBMS): Sistem, güvenilir bir DBMS (örn. MySQL, PostgreSQL) ile entegre edilmelidir. Veritabanı, kullanıcıların yüklediği veri setlerini, kriter ve ağırlıkları ve model sonuçlarını saklamalıdır.

- Veri Temizleme Modülü:\*Veri temizleme işlemlerini gerçekleştiren bir yazılım modülü bulunmalıdır. Bu modül, yüklenen veri setlerindeki eksik, hatalı veya çelişkili verileri temizleyerek modelin doğruluğunu artırmayı amaçlar.

- Model Optimizasyon Modülü: Model optimizasyon işlemlerini gerçekleştiren bir yazılım modülü bulunmalıdır. Bu modül, modelin parametrelerini optimize ederek en iyi performansı elde etmeyi hedefler.

Harici Sistemlerle Bağlantı:

- Tarih Kurumu Veritabanı (HS): Harici bir sistem olarak Tarih Kurumu Veritabanı ile bağlantı sağlanacaktır. Admin, HS Veritabanı üzerinden gözden geçirici üyeliklerini doğrulayabilir. HS Veritabanı, üyenin adı, üyelik (ID) numarası ve e-posta adresi gibi bilgileri içerir. Üyelik doğrulama işlemi için, gözden geçiricinin kimlik bilgileri HS Veritabanı'na gönderilir ve üyelik durumu Boolean bir değer olarak döndürülür. Yeni gözden geçirici eklerken, üye adları, üyelik numaraları ve e-posta adreslerinin bir listesi istenebilir. Göz

**3.2 İşlevsel Gereksinimler**

3.2.1 Veri Import Etme

|  |  |
| --- | --- |
| **Kullanım Örneği Adı** | Veri Import Etme |
| **XRef** | |  | | --- | |  |  |  | | --- | | Bölüm 2.2.1, Güncelle Veri | |
| **Tetikleyici** | Admin yeni veri setini yüklemek için veri import ekranını seçer. |
| **Ön Koşul** | Admin sisteme giriş yapmış ve veri import ekranına erişmiş olmalıdır. |
| **Temel Yol** | 1. Admin veri import ekranını açar.  2. Admin, yüklemek istediği veri dosyasını seçer.  3. Sistem, dosyanın formatını ve içeriğini kontrol eder.  4. Sistem, dosyayı veritabanına import eder. |
| **Alternatif Yollar** | |  | | --- | |  |  |  | | --- | | 3.adımda eğer dosya formatı geçersizse, sistem hata mesajı gösterir ve admin dosyayı yeniden seçer. Eğer dosya içeriğinde hatalar varsa, sistem hata mesajı gösterir ve admin dosyayı düzeltir. | |
| **Son Koşul** | |  | | --- | |  |  |  | | --- | | Veri seti başarıyla veritabanına import edilmiştir. | |
| **İstisna Yolları** | Admin herhangi bir zamanda operasyonu terk edebilir. |
| **Diğer** | Veri import işlemi sırasında dosya formatı (örn. CSV, Excel) ve içerik yapısına dikkat edilmelidir. |

3.2.2 Satış Tahmini

|  |  |
| --- | --- |
| **Kullanım Örneği Adı** | Satış Tahmini |
| **XRef** | |  | | --- | |  |   Bölüm 2.2.1, Güncelle Model |
| **Tetikleyici** | Admin, satış tahmin modeli oluşturmak için ilgili ekranı seçer. |
| **Ön Koşul** | |  | | --- | |  |  |  | | --- | | Admin sisteme giriş yapmış ve satış tahmin ekranına erişmiş olmalıdır. | |
| **Temel Yol** | |  | | --- | |  |  |  | | --- | | 1. Admin, satış tahmin ekranını açar.  2. Admin, tahmin için gerekli parametreleri ve veri setini seçer.  3. Sistem, veri temizleme işlemini gerçekleştirir.  4. Sistem, tahmin modelini oluşturur ve optimize eder.  5. Sistem, tahmin sonuçlarını admin'e sunar. | |
| **Alternatif Yollar** | |  | | --- | |  |  |  | | --- | |  |   Eğer veri setinde eksiklik veya hata varsa, sistem hata mesajı gösterir ve admin veri setini düzeltir. |
| **Son Koşul** | |  | | --- | |  |   Satış tahmin modeli başarıyla oluşturulmuş ve sonuçlar admin'e sunulmuştur. |
| **İstisna Yolları** | Admin herhangi bir zamanda operasyonu terk edebilir. |
| **Diğer** | Satış tahmin modeli oluşturma işlemi sırasında, veri temizliği ve model optimizasyonuna dikkat edilmelidir. |

3.2.3 Hata Hesaplama

|  |  |
| --- | --- |
| **Kullanım Örneği Adı** | Hata Hesaplama |
| **XRef** | |  | | --- | |  |   Bölüm 2.2.1, Güncelle Model |
| **Tetikleyici** | Admin, model performansını değerlendirmek için hata hesaplama ekranını seçer. |
| **Ön Koşul** | Admin sisteme giriş yapmış ve hata hesaplama ekranına erişmiş olmalıdır. |
| **Temel Yol** | |  | | --- | |  |  |  | | --- | | 1. Admin, hata hesaplama ekranını açar.  2. Admin, model sonuçları ve gerçek verileri yükler.  3. Sistem, hata hesaplamalarını gerçekleştirir.  4. Sistem, hata sonuçlarını admin'e sunar. | |
| **Alternatif Yollar** | |  | | --- | |  |  |  | | --- | |  | |
| **Son Koşul** | |  | | --- | |  |  |  | | --- | | Hata hesaplamaları başarıyla tamamlanmış ve sonuçlar admin'e sunulmuştur. | |
| **İstisna Yolları** | Admin herhangi bir zamanda operasyonu terk edebilir. |
| **Diğer** | Hata hesaplama işlemi sırasında veri doğruluğu ve model performansı üzerinde dikkat edilmelidir. |

3.2.4 Kriter ve Ağırlık Belirleme

|  |  |
| --- | --- |
| **Kullanım Örneği Adı** | Kriter ve Ağırlık Belirleme |
| **XRef** | |  | | --- | |  |   Bölüm 2.2.1, Güncelle Kriter ve Ağırlık |
| **Tetikleyici** | Admin, satış tahmin modeli için kriter ve ağırlıkları belirlemek için ilgili ekranı seçer. |
| **Ön Koşul** | Admin sisteme giriş yapmış ve kriter belirleme ekranına erişmiş olmalıdır. |
| **Temel Yol** | |  | | --- | |  |  |  | | --- | | 1. Admin, kriter belirleme ekranını açar. <br> 2. Admin, model için kullanılacak kriterleri ve bunların ağırlıklarını girer. <br> 3. Sistem, girilen bilgileri doğrular ve kaydeder. | |
| **Alternatif Yollar** | |  | | --- | |  |  |  | | --- | |  |   Eğer girilen bilgilerde eksiklik veya hata varsa, sistem hata mesajı gösterir ve admin bilgileri düzeltir. |
| **Son Koşul** | |  | | --- | |  |  |  | | --- | |  |   Kriter ve ağırlıklar başarıyla belirlenmiş ve kaydedilmiştir. |
| **İstisna Yolları** | |  | | --- | |  |  |  | | --- | | Admin herhangi bir zamanda operasyonu terk edebilir | |
| **Diğer** | Kriter ve ağırlık belirleme işlemi sırasında modelin doğruluğunu etkileyecek faktörlere dikkat edilmelidir |

3.2.5 Veri Puanlama

|  |  |
| --- | --- |
| **Kullanım Örneği Adı** | Veri Puanlama |
| **XRef** | |  | | --- | |  |   Bölüm 2.2.1, Güncelle Veri Puanlama |
| **Tetikleyici** | Admin, veri setini belirlenen kriterlere göre puanlamak için ilgili ekranı seçer. |
| **Ön Koşul** | Admin sisteme giriş yapmış ve veri puanlama ekranına erişmiş olmalıdır |
| **Temel Yol** | 1. Admin, veri puanlama ekranını açar.  2. Admin, puanlanacak veri setini seçer.  3. Sistem, belirlenen kriterlere göre veri setini puanlar.  4. Sistem, puanlanmış veriyi admin'e sunar. |
| **Alternatif Yollar** | |  | | --- | |  |  |  | | --- | |  | |
| **Son Koşul** | |  | | --- | |  |  |  | | --- | |  |   Veri seti başarıyla puanlanmış ve sonuçlar admin'e sunulmuştur. |
| **İstisna Yolları** | Admin herhangi bir zamanda operasyonu terk edebilir. |
| **Diğer** | Veri puanlama işlemi sırasında kriterlerin doğru uygulanmasına dikkat edilmelidir. |

3.2.6 Veri Temizleme

|  |  |
| --- | --- |
| **Kullanım Örneği Adı** | Veri Temizleme |
| **XRef** | |  | | --- | |  |   Bölüm 2.2.2, Hata Hesaplama ve Veri Temizliği |
| **Tetikleyici** | Sistem, yeni bir veri seti üzerinde işlem yapmak üzere başlatılır. |
| **Ön Koşul** | Veri seti daha önce sisteme içe aktarılmış olmalıdır. |
| **Temel Yol** | |  | | --- | | 1. Sistem, veri gürültüsü temizleme sürecini başlatır. |  |  |  | | --- | --- | |  | 2. Sistem, tespit edilen veri problemlerini temizler. |  |  |  | | --- | --- | |  | 3. Sistem, temizlenmiş veri setini hazırlar. |  |  |  | | --- | --- | |  | 4. Sistem, tahmin sonuçlarını ve gerçek satış verilerini karşılaştırır. |  |  |  | | --- | --- | |  | 5. Sistem, hata hesaplama işlemini başlatır. |  |  |  | | --- | --- | |  | 6. Sistem, hesaplanan hata değerlerini kullanıcıya sunar. |  |  |  | | --- | --- | |  | 7. Kullanıcı, hesaplanan hata değerlerini inceler ve değerlendirir. | |
| **Alternatif Yollar** | |  | | --- | |  |  |  | | --- | | - Eğer hata hesaplama sürecinde bir problem oluşursa, sistem kullanıcıya hata mesajı ile durumu bildirir | |
| **Son Koşul** | |  | | --- | |  |  |  | | --- | |  |   Hata hesaplama raporu kullanıcıya başarılı bir şekilde sunulmuştur. |
| **İstisna Yolları** | Kullanıcı, herhangi bir aşamada işlemi iptal edebilir. |
| **Diğer** | Veri temizliği ve hata hesaplama işlemleri sırasında sistemin veri bütünlüğünü koruması gerekmektedir. |

3.2.7 Model Optimizasyonu

|  |  |
| --- | --- |
| **Kullanım Örneği Adı** | Model Optimizasyonu |
| **XRef** | |  | | --- | |  |   Bölüm 2.2.2, Model Performans Optimizasyonu |
| **Tetikleyici** | Sistem, modelin performansını artırmak için optimizasyon sürecini başlatır. . |
| **Ön Koşul** | Sistem, veri temizleme işlemini gerçekleştirmiş ve model eğitilmiş olmalıdır. |
| **Temel Yol** | |  |  | | --- | --- | |  | 1. Sistem, model optimizasyon sürecini başlatır. |  |  |  | | --- | --- | |  | 2. Sistem, optimizasyon tekniklerini belirler. |  |  |  | | --- | --- | |  | 3. Sistem, modelin parametrelerini ayarlar ve yeniden eğitir. |  |  |  | | --- | --- | |  | 4. Sistem, optimize edilen modelin performansını değerlendirir. |  |  |  | | --- | --- | |  | 5. Sistem, optimize edilen modelin performans raporunu sunar. |  |  |  | | --- | --- | |  | 6. Kullanıcı, optimize edilen modelin performansını inceler ve onaylar. |  |  |  | | --- | --- | |  | 7. Sistem, optimize edilen modeli gelecekteki tahminlerde kullanıma hazır hale getirir. |  |  |  | | --- | --- | |  | 8. Kullanıcı, model optimizasyon sürecini tamamlar ve sistem ana sayfasına döner. | |
| **Alternatif Yollar** | |  | | --- | |  |  |  | | --- | | - Kullanıcı, optimizasyon sürecinde önerilen değişiklikleri reddedebilir ve işlemi durdurabilir. | |
| **Son Koşul** | |  | | --- | |  |  |  | | --- | |  |   Optimize edilmiş model, başarılı bir şekilde kaydedilmiş ve kullanıma hazır hale getirilmiştir. |
| **İstisna Yolları** | Kullanıcı, herhangi bir aşamada işlemi iptal edebilir. |
| **Diğer** | Optimizasyon işlemi sırasında sistemin kaynak kullanımı ve zamanlama dikkatlice yönetilmelidir. |

**4.Tasarım**

**4.1.Sistemin Tasarım Mimarisi:**

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 3-Sistem Akış Diyagramı*

**Sistem Akış Diyagramı Açıklaması ve Seçim Nedenleri :**

* **Kullanıcı->Import Database:** Bu adımda kullanıcı, işlemek için gereken veriyi veritabanına aktarır. Kullanıcı, veritabanına veri aktarmanın yanı sıra, sistemin hangi verileri kullanacağını belirler.
* **Kullanım Amacı:** Kullanıcıların çeşitli formatlardaki verileri(.csv,xlsx,.txt vb.) sisteme kolaylıkla yüklemesini sağlamak.
* **Veri Modeli:** Genellikle tablo formatında, veritabanı olarak depolanır. Her bir sütun farklı bir özellik veya değişkeni, her satır ise bir örneği temsil eder.
* **Testler:** Yüklenen verinin formatı ve bütünlüğü kontrol edilir. Hatalı veya eksik verilerin sisteme girmesini önlemek için validasyon testleri uygulanır.
* **Performans Kriterleri:** Veri yükleme hızı ve yüklenen verinin doğruluğu.
* **Data Cleaning:** Bu adımda, alınan veriler kontrol edilir ve herhangi bir hata, tutarsızlık veya eksik değer varsa düzeltilir. Bu, verinin daha sonra modelleme ve tahminleme aşamalarında daha doğru ve güvenilir olmasını sağlar.
* **Kullanım Amacı:** Kullanıcılara veri temizleme ve önişleme seçenekleri sunmak. Bu, örneğin eksik değerlerin doldurulması, aykırı değerlerin ele alınması, gürültülü verinin düzeltilmesi gibi işlemleri içerir.
* **Veri Modeli:** Temizlenmiş ve işlenmiş veri, modelleme aşamalarında kullanılmak üzere veritabanına depolanır.
* **Testler:** Verinin temizlendiği ve uygun bir biçime getirildiği, veri bütünlüğünün korunduğu kontrol edilir.
* **Performans Kriterleri:** Veri temizleme hızı ve sonuçtaki veri bütünlüğü.
* **Scoring Criteria:** Bu adımda, tahminleme sürecindeki kullanılacak olan özelliklerin ve değişkenlerin ağırlıklarını belirlenir.
* Kullanım Amacı: Kullanıcılara, belirli özelliklerin ve değişkenlerin tahminleme sürecinde ne kadar önemli olduğunu belirlemelerine olanak sağlamak.
* Veri Model: Her bir özellik veya değişken için belirlenen ağırlıklar ve kriterler, daha sonra tahminler yapmak üzere depolanır.
* Testler: Kriterlerin ve ağırlıkların doğru şekilde ayarlandığı kontrol edilir.
* Performans Kriterleri: Kriter belirleme hızı ve sonuçtaki tahminlerin doğruluğu.
* **Sales Forecasting:** Bu adımda, belirlenen ağırlıklar ve kriterler kullanılarak satış tahminleri yapılır.
* Kullanım Amacı: Kullanıcılara, veriler ve belirlenen kriterler kullanılarak gelecekteki satışları tahmin etme yeteneği sunmak
* Veri Model: Tahminler, sonuçları analiz etmek ve paylaşmak üzere depolanır.
* Testler: Tahminlerin doğruluğu ve güvenirliği kontrol edilir.
* Performans Kriterleri: Tahminleme hızı ve sonuçların doğruluğu.
* **Model Optimization:** Bu adımda, tahmin sonuçlarına dayalı olarak modelin ağırlıklarını ve parametrelerini ayarlar.
* Kullanım Amacı: Kullanıcılara, modelin ağırlıklarını ve parametrelerini ayarlamalarına ve modelin performansını arttırmalarına olanak sağlamak.
* Veri Model: Optimize edilmiş model parametreleri, gelecekteki tahminler yapmak üzere depolanır.
* Testler: Modelin Optimal Olduğu ve tahminlerin doğruluğunun arttığı kontrol edilir.
* Performans Kriterleri: Model optimizasyon hızı ve sonuçtaki tahminlerin doğruluğu.
* **Error Calculation:** Bu adımda, tahmin sonuçlarının ne kadar doğru olduğunu belirlemek için hata hesaplaması yapılır.
* Kullanım Amacı: Kullanıcılara, tahminlerin ne kadar doğru olduğunu belirlemek için hata hesaplaması yapma yeteneği sunmak.
* Veri Model: Hata metrikleri, sonuçları analiz etmek ve paylaşmak üzere depolanır.
* Testler: Hata metriklerinin doğru hesaplandığı kontrol edilir.
* Performans Kriterleri: Hata hesaplama hızı ve sonuçların doğruluğu.

metin, ekran görüntüsü, diyagram, plan içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 4- Önerilen Sistem için Varlık İlişki Diyagramı(ERD)*

* **Input Weights & Criteria Value:** Bu son adımda, hata hesaplamalarına dayalı olarak girdi ağırlıklarını ve kriter değerlerini ayarlar.
* Kullanım Amacı: Kullanıcılara, girdi ağırlıklarını ve kriter değerlerini ayarlamalarına ve gelecekteki tahminlerin doğruluğunu artırmalarına olanak sağlamak.
* Veri Model: Ayarlanmış ağırlıklar ve kriterler, gelecekteki tahminler yapmak üzere depolanır.
* Testler: Ağırlıkların ve kriterlerin doğru şekilde ayarlandığı kontrol edilir.
* Performans Kriterleri: Kullanıcılar tarafından ayarlanan ağırlıkların ve kriterlerin modelin genel performansı ve tahmin doğruluğunu ne kadar iyileştirdiği önemli bir performans ölçütüdür. Diğer bir önemli performans kriteri ise bu ağırlık ve kriter ayarlamalarının ne kadar hızlı yapılabildiğidir.

**4.2.Arabirimlerin ve Modüllerin Detayları:**

**a.Home:**

* **Amaç:** Bu arabirim, uygulamanın ana sayfasıdır. Kullanıcıların sisteme giriş yapmasını ve genel bilgilere erişmesini sağlar.
* **Veri Modeli:** Kullanıcı bilgileri ve genel sistem verileri.
* **Testler:** Arabirimin tüm özellikleri, tıklanabilir öğeler, kullanılabilirlik ve kullanıcı deneyimi için kullanılabilirlik testleri.
* **Performans Kriterleri:** Arabirimin hızlı ve kesintisiz yüklenmesi, gecikme olmaması.
* **Modüller:** Kullanıcı oturum yönetimi, Genel bilgi panelleri.

**b.Forecast:**

* **Amaç:** Bu arabirim, tahminleme işlemlerini gerçekleştirir. Gelecekteki satışları ve diğer metrikleri tahmin etmek için kullanılır.
* **Veri Modeli:** Tarihsel satış verileri, skorlama kriterleri ve kullanıcı girdileri.
* **Testler:** Tahmin doğruluğu, algoritmanın performansı ve hızı.
* **Performans Kriterleri:** Tahminlerin doğruluğu ve hızlı hesaplama süresi.
* **Modüller:** Veri analizi ve işleme, tahminleme algoritması.

**c.Transaction:**

* **Amaç:** Bu arabirim, tüm işlem bilgilerini görüntüler ve yönetir.
* **Veri Modeli:** İşlem verileri, ürün bilgileri ve diğer ilgili detaylar.
* **Testler:** Veri doğruluğu ve tutarlılığı, arayüzün kullanılabilirliği ve performansı.
* **Performans Kriterleri:** Veri doğruluğu ve hızlı veri yükleme süreleri.
* **Modüller:** İşlem yönetimi ve veri görüntüleme.

**d. Scoring:**

* **Amaç:** Bu arabirim, skorlama kriterlerini yönetir ve puanlamayı gerçekleştirir.
* **Veri Modeli:** Skorlama kriterleri, kullanıcı girdileri ve ürün bilgileri.
* **Testler:** Algoritmanın doğruluğu, kriterlerin doğru bir şekilde uygulanması.
* **Performans Kriterleri:** Skorlama doğruluğu ve hızlı hesaplama süresi.
* **Modüller:** Kriter yönetimi, skorlama algoritması.

**e. Backup:**

* **Amaç:** Bu arabirim, veri yedeklemeyi yönetir. Sistemdeki verilerin düzenli olarak yedeklenmesini sağlar.
* **Veri Modeli:** Tüm sistem verileri.
* **Testler:** Yedeklemenin doğru ve düzenli olarak gerçekleştirilmesi, yedeklerin doğruluğu ve eksiksiz olması.
* **Performans Kriterleri:** Düzenli ve eksiksiz yedekleme, hızlı yedekleme süresi.
* **Modüller:** Veri yedekleme yönetimi, veri saklama.

**f. Settings:**

* **Amaç:** Bu arabirim, kullanıcı ayarlarını ve sistem ayarlarını yönetir.
* **Veri Modeli:** Kullanıcı bilgileri ve sistem ayarları.
* **Testler:** Ayarların doğru bir şekilde uygulanması, kullanıcı deneyimi ve kullanılabilirlik.
* **Performans Kriterleri:** Ayarların hızlı ve doğru bir şekilde uygulanması.
* **Modüller:** Kullanıcı ayarları yönetimi, sistem ayarları yönetimi.

**g. Messages:**

* **Amaç:** Bu arabirim, sistem içindeki mesajlaşmayı yönetir.
* **Veri Modeli:** Mesaj verileri, kullanıcı bilgileri.
* **Testler:** Mesajların doğru bir şekilde iletilmesi ve alınması, kullanılabilirlik ve kullanıcı deneyimi.
* **Performans Kriterleri:** Hızlı ve kesintisiz mesaj iletimi.
* **Modüller:** Mesajlaşma yönetimi.

**h. Folders:**

* **Amaç:** Bu arabirim, sistem içindeki dosya ve klasörlerin yönetimini sağlar.
* **Veri Modeli:** Dosya ve klasör bilgileri.
* **Testler:** Dosya/klasör oluşturma, düzenleme ve silme işlemlerinin doğruluğu, kullanılabilirlik.
* **Performans Kriterleri:** Hızlı ve kesintisiz dosya/klasör yönetimi.
* **Modüller:** Dosya/klasör yönetimi.

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 5-Error Modülü Akış Diyagramı*

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 6-Forecast Modülü Akış Diyagramı*

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 7-Order Modülü Akış Diyagramı*

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 8-OrderDetail Modülü Akış Diyagramı*

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 9- Product Modülü Akış Diyagramı*

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 10-Scoring Modülü Akış Diyagramı*

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 11- Transaction Modülü Akış Diyagramı*

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Şekil 12-Realization Modülü Akış Diyagramı*

**5.Gerçekleştirme**

**5.1.Önerilen sistemin gerçekleştirilme sürecindeki ayrıntılar:**

**Programlama Dili ve Araçları:**Python dilini tercih ettik çünkü Python, veri analizi, makine öğrenmesi ve tahminleme algoritmalarının uygulanmasında oldukça güçlü ve popüler bir dildir. Ayrıca, Python'un geniş kütüphane ekosistemi (numpy, pandas, scikit-learn, keras, matplotlib, seaborn vb.) bu projenin uygulanmasında büyük kolaylık sağlar.

Veri tabanı yönetim sistem olarak PostgreSQL kullandık. PostgreSQL, açık kaynaklı bir veritabanı yönetim sistemidir ve geniş özellik seti, işletme sınıfı performansı ve açık kaynaklı olması nedeniyle tercih edilmektedir. PostgreSQL, JSON ve JSONB türleri gibi modern veri türlerini destekler ve bir dizi gelişmiş özellik sunar.

Veri modellemesi ve analizi için Jupyter Notebook kullandık. Jupyter Notebook, Python kodunun etkileşimli ve görsel bir şekilde çalıştırılmasını sağlar, bu da veri analizi ve makine öğrenmesi projeleri için idealdir.

**5.2.Veri Tabanı Yönetim Sistemi Mimarisi:**

**5.2.1.Kullanıcı Arayüzü (User Interface):** Kullanıcıların etkileşimde bulunduğu ve siparişleri oluşturduğu, sorguladığı veya güncellediği katmandır. Bu katman genellikle bir web tarayıcısı veya mobil uygulama içerisinde yer alır.

**5.2.2.Uygulama Katmanı (Application Layer):** İş mantığı, hesaplama ve tahminleme işlemlerinin gerçekleştiği katmandır. İçerisinde sipariş oluşturma, puanlama, tahminleme ve hata hesaplama gibi birçok işlemi içerir. Ayrıca, kullanıcı arayüzü ve veri tabanı arasında bir aracı görevi görür.

**5.2.3.Veri Tabanı (Database):** Tüm siparişler, sipariş detayları, ürünler, işlemler, skorlar, tahminler ve hataların saklandığı katmandır. Veri tabanı yönetim sistemi olarak genellikle SQL tabanlı bir DBMS (örneğin, MySQL, PostgreSQL veya SQL Server) kullanılır.

**5.2.4.Kullanıcı arayüzü,** kullanıcıların verileri görüntülemesi, eklemesi, düzenlemesi veya silmesi için gereken işlevleri sağlar. Kullanıcı arayüzünden gelen talepler, uygulama katmanına iletilir. Bu katman, kullanıcı taleplerini işler ve gerektiğinde veri tabanına sorgular gönderir veya veri tabanına veri yazar.

**5.2.5.Uygulama katmanı,** siparişlerin oluşturulması, puanlama kriterlerinin uygulanması, tahminlerin hesaplanması ve hataların bulunması gibi bir dizi işlevi gerçekleştirir. Bu işlemler, genellikle bir dizi sunucu üzerinde dağıtılmış olan ve büyük veri setleriyle çalışabilen ölçeklenebilir ve yüksek performanslı bir uygulama çerçevesinde gerçekleştirilir.

**5.2.6.Veri tabanı katmanı, t**üm verileri saklar ve uygulama katmanının bu verilere erişmesini sağlar. Veri tabanı, güvenli ve hızlı veri erişimi sağlamak için bir dizi veri tablosu ve indeks kullanır.

**5.3.Standartlar:**

Kodlama standartları olarak, Python dilinde genel olarak kabul görmüş PEP8 standartlarını kullandık. Bu standartlar, kodun okunabilirliğini ve tutarlılığını sağlar.

**5.4.Olağan Dışı Durumlar ve Yönetimi:**

Olağan dışı durumlar olarak, veri tabanı bağlantı hataları, eksik veya hatalı veri girişleri, makine öğrenmesi modelinin beklenenden düşük performans göstermesi gibi durumlar belirlenmiştir. Bu durumların yönetimi için hata yakalama ve loglama mekanizmaları, gelişmiş veri ön işleme teknikleri ve model optimizasyon stratejileri uygulanmıştır.

**5.5.Kod Gözden Geçirme:**

Kod gözden geçirme, ekibin kod kalitesini ve tutarlılığını sağlamak için uygulanan bir süreçtir. Bu süreçte, kodun okunabilirliği, performansı, işlevselliği ve uyumluluğu değerlendirilir. Kod gözden geçirme işlemi genellikle peer review (eş değerlendirme) şeklinde yapılır. Bu projede de bu metodoloji uygulanmıştır.

**6.Bakım**

**6.1.Yazılım Kurulum Aktiviteleri:**

* **Ön-Kurulum Kontrolü:** DataSalesPredictor kurulumuna başlamadan önce, hedef sistemlerin yazılım gereksinimlerini karşıladığından emin olunmalıdır. Bu, işletim sistemi uyumluluğu, yeterli disk alanı, gerekli üçüncü taraf yazılımların varlığı ve gereken sistem özelliklerini içerir.
* **Yazılım Kurulumu:** Kurulum süreci, genellikle kurulum sihirbazı veya komut satırı aracılığıyla gerçekleştirilir. Bu süreçte, yazılımın gerekli dosyaları hedef sistemlere kopyalanır ve gerekli yapılandırmalar gerçekleştirilir.
* **Yazılım Yapılandırması:** Kurulumun tamamlanmasının ardından, yazılım belirli bir sistem veya ağ ortamına uyacak şekilde yapılandırılmış olmalıdır. Bu, veritabanı bağlantıları, ağ ayarları, kullanıcı hesapları ve daha fazlasını içerir.
* **Testler:** Kurulum ve yapılandırmanın tamamlanmasının ardından, yazılımın düzgün çalışıp çalışmadığını doğrulamak için genel işlevsellik ve performans testleri gerçekleştirilmelidir.

**6.2.Yerinde Destek Organizasyonu:**

* **Teknik Destek Ekibi:** DataSalesPredictor, kullanıcıların her türlü teknik sorunu çözmelerine yardımcı olmak için bir teknik destek ekibine sahip olmalıdır. Bu ekip, telefon, e-posta veya çevrimiçi sohbet yoluyla erişilebilir olmalıdır.
* **Yazılım Güncellemeleri ve Bakım:** Yazılımın düzgün çalışmasını sürdürmek ve yeni özellikler eklemek için düzenli yazılım güncellemeleri ve bakımı sağlanmalıdır. Bu, genellikle otomatik güncelleme mekanizmaları veya manuel indirme ve yükleme prosedürleri ile gerçekleştirilir.

**6.3.Kurulum ve Entegrasyon Aşamalarında Yapılacaklar:**

* **Kurulum Sonrası Yapılandırma:** Kurulum sonrasında, yazılımı hedef sistem ve ağ ortamına entegre etmek için belirli bir dizi yapılandırma gereklidir. Bu, ağ bağlantıları, veritabanı bağlantıları, kullanıcı hesapları ve diğer sistem ayarlarını içerir.
* **Yazılım Entegrasyonu:** Yazılım, hedef sistemin diğer yazılım ve donanımlarıyla uyumlu olacak şekilde entegre edilmelidir. Bu, API bağlantıları, veritabanı bağlantıları ve belki de diğer yazılımlarla etkileşim için özel entegrasyon senaryolarını içerir.
* **Kullanıcı Eğitimi:** Yazılımın etkili bir şekilde kullanılabilmesi için kullanıcılar yazılımın temel işlevleri ve özellikleri konusunda eğitilmelidir. Bu genellikle çevrimiçi öğreticiler, kullanım kılavuzları ve canlı eğitim seansları aracılığıyla yapılır.

**7.Sonuç**

**7.1.Sonuç ve Değerlendirme:**

DataSalesPredictor, satış verilerinin tahminini yapmayı ve işletmelerin gelecekteki satış performanslarını daha iyi anlamalarını sağlamayı amaçlayan bir yazılım sistemidir.

**7.2.Mevcut Sistemlerden Farkı:**

DataSalesPredictor, mevcut sistemlere göre daha fazla özelleştirilebilirlik ve adaptasyon yeteneği sunmaktadır. Kullanıcıların kendi satış verilerini yüklemesine ve özelleştirilmiş tahminler oluşturmasına olanak sağlar. Ayrıca, öğrenme yetenekleri sayesinde, yazılım zaman içinde daha doğru tahminler yapmayı öğrenebilir.

**7.3.Hangi Probleme Çözüm Üretiyor:**

DataSalesPredictor, işletmelerin gelecekteki satış performansını tahmin etme problemine çözüm üretmektedir. Bu, işletmelerin daha etkili planlama ve karar verme yapmasını sağlar.

**7.4.Avantajlar ve Dezavantajlar:**

**Avantajları:**

* Yüksek derecede özelleştirilebilirlik
* Öğrenme yetenekleri ile zaman içinde daha doğru tahminler yapabilir
* Farklı para birimleri, biçimlendirme standartları ve dil sözleşmeleri ile uluslararasılaştırma ve yerelleştirme gereksinimlerini destekler.

**Dezavantajları:**

* Yeni bir sistem olduğu için, kullanıcılarının bu sistem üzerinde yeterince deneyim kazanması zaman alabilir.
* Mevcut veri tabanı sistemleri ile tam uyumluluğu sağlamak bazen karmaşık olabilir.

**7.5.Gelecekte Nasıl İyileştirilebilir:** DataSalesPredictor'un dezavantajları, kullanıcı eğitimine daha fazla odaklanarak ve daha fazla veri tabanı entegrasyon seçenekleri sunarak giderilebilir. Ayrıca, yazılımın tahmin algoritmaları sürekli olarak güncellenerek ve geliştirilerek daha doğru tahminler yapma yeteneği artırılabilir.

**EK A: Sözlük**

Bu bölüm, DataSalesPredictor SRS belgesi okuyucularının spesifik terimler, kısaltmalar ve belgeye özgü jargonu anlamasını sağlar. Veri Bilimi ve Makine Öğrenmesi gibi çeşitli alanlarda yaygın olarak kullanılan terimleri içerir. Aşağıda belirtilen terimlerin tanımları bu sözlük içerisinde yer almaktadır:

**DNN:** Derin Sinir Ağı (Deep Neural Network)

**ML:** Makine Öğrenmesi (Machine Learning)

**AI:** Yapay Zeka (Artificial Intelligence)

**API:** Uygulama Programlama Arayüzü (Application Programming Interface)

**ROI:** Yatırım Getirisi (Return on Investment)

Bu belgede belirtilen terimlerin tam listesi ve detaylı açıklamaları Sözlük bölümünde sunulmuştur. Herhangi bir kısaltmanın veya özel terimin anlamı hakkında şüpheniz olursa bu bölüme başvurmanız önerilir.

**EK B: Analiz Modelleri**

Analiz Modelleri bölümü, DataSalesPredictor projesi için oluşturulan özel analiz ve tasarım modellerini içerir. Bu modellerin amacı, yazılımın işlevselliğini ve iş akışını görselleştirmek ve sistemin genel yapısını anlamayı kolaylaştırmaktır.

**Dokümanda yer alan diyagramlar ve projedeki amaçları:**

* **Use Case Diyagramı:** Sistemdeki kullanıcıların davranışlarını ve sistemle olan etkileşimlerini gösterir. Kullanım senaryolarını ve kullanıcıların isteklerini temsil eder. Sistem gereksinimlerini anlamak, kullanıcı ihtiyaçlarını belirlemek ve sistemin işlevselliğini tanımlamak için kullanılır.
  + **UML Sınıf Diyagramı:** Proje içerisindeki sınıfları, sınıflar arasındaki ilişkileri ve sınıfların özelliklerini ve işlevlerini gösterir. Veri modelini, veri tabanı yapısını ve sınıflar arasındaki ilişkileri anlamak için kullanılır
* **Varlık İlişki Diyagramı:** Veri tabanı modelini temsil eder. Varlık tiplerini, varlık arasındaki ilişkileri ve veri tabanının yapısal bütünlüğünü gösterir. Veri tabanı tasarımını anlamak ve veri modelini oluşturmak için kullanılır.
* **Örnek Modelleme Şeması:** Belirli bir örneği modellemek ve sistemin nasıl çalıştığını anlamak için kullanılır. Örnekleme senaryolarını ve sistem davranışlarını temsil eder. Sistemin nasıl çalıştığını gösterir ve hataları tespit etmeye yardımcı olur.
* **Sistem Akış Diyagramı:** Sistemdeki bileşenlerin ve işlemlerin akışını gösterir. İşlemlerin sırasını, veri akışını ve bileşenler arasındaki etkileşimi temsil eder. Sistemin genel işleyişini anlamak ve işlemlerin nasıl gerçekleştiğini göstermek için kullanılır.
* **Modül Akış Diyagramları:** Her bir modülün işleyişini ve içerdiği işlemleri gösterir. Modül içindeki fonksiyonların sırasını, veri akışını ve modül arasındaki etkileşimi temsil eder. Modülün işlevselliğini anlamak ve modülün içindeki işlemleri takip etmek için kullanılır.

**8-Kaynakça:**

**1-2.Bölüm için:**

[1] Sarıtaş, S. (2018). Veri Bilimi ve Makine Öğrenmesi Kullanarak Satış Tahminleme: Bir Perakende İşletmesi Örneği. Journal of International Social Research, 11(62).

[2] Şahin, İ., & Ekinci, Y. (2020). Büyük Veri Analitiği ve Yapay Zekâ Temelli Makine Öğrenmesi Algoritmaları ile Satış Tahminlemesi: BİM Birleşik Mağazalar A.Ş. Üzerine Bir Uygulama. Business & Management Studies: An International Journal, 8(1), 311- 330.

[3] Jain, A., & Kumar, A. (2016). An hybrid approach for sales forecasting using clustering and regression models. International Journal of Computer Applications, 136(12).

<https://www.ijcaonline.org/archives/volume136/number12/24466-2016908801>

[4] Bal, İ. E., & Bozkurt, S. (2019). Yapay zekâ temelli tahminleme yöntemlerinin stok yönetimine etkisi: bir otomotiv yan sanayi firması örneği. Uluslararası İktisadi ve İdari İncelemeler Dergisi, 21(36).

[5] Cervone, G., Clemente, L., Alessandrini, A., & Delli, G. (2020). Sales forecasting using machine learning models: a case study with Spark MLlib. Big Earth Data, 4(1), 36-52.

**3-4.Bölüm için:**

[8] "The Elements of Statistical Learning" - Trevor Hastie, Robert Tibshirani, Jerome Friedman. (2009). Springer.

[9] "Practical Statistics for Data Scientists" - Peter Bruce and Andrew Bruce. (2017). O’Reilly Media.

**5.Bölüm İçin:**

[10] Güneş, B. (2021). Veritabanı Sistemleri: PostgreSQL ile Geliştirme. Ankara: Ay Yayıncılık.

[11] Ekşi, A. (2019). Python Programlama Dili ile Makine Öğrenmesi.

**6-7-8.Bölüm İçin:**

[14] Börüban, D. (2016). Yazılım Yönetimi: Planlama, Organizasyon ve Kontrol. Seçkin Yayıncılık.

[15] Tınmaz, E. (2013). Bilgi Teknolojileri Yönetimi. Seçkin Yayıncılık.

[16] Pressman, R. S. (2010). Software Engineering: A Practitioner's Approach. McGraw-Hill Higher Education.