INF103 Algoritma ve Ileri Bilgisayar Programlama

Proje 18/04/2019

1 Giriş

Dokümanda 2018-2019 Bahar dönemi INF103 Algoritma ve İleri Bilgisayar Programlama dersi proje konusu anlatılmıştır. Proje yapay öğrenme ve veri analizi konularını kapsar. Bu amaçla size verilen problemi bu dönem öğrenmiş olduğunuz algoritma ve veri yapılarını kullanarak çözmeniz istenmektedir. Problemde kullanılacak veri seti ev satışları ile ilgilidir.

Proje iki bölümden oluşur: Veri analizi ve Model tasarımı. Veri analizi bölümünde istenen veri setini ilgili dosyadan okumanız ve onun hakkında bilgi edinmenizdir. Bölüm 2'de istenenleri ayrıntılı bir şekilde bulabilirsiniz. Model tasarımı bölümünde istenen ise analizi yapılan veri üzerinden bir tahmin modeli oluşturmak ve yeni bir ev geldiğinde uygun fiyatı verebilmektir. Bölüm 3'de istenenleri ayrıntılı şekilde bulabilirsiniz.

1.1 Anahtar Kelimeler ve Tanımlar

Bu bölümde dokümanda kullanılan terimlerin ve anahtar kelimelerin tanımları verilmiştir.

- **Yapay öğrenme (Machine Learning)**, bilgisayar sistemlerini açık bir şekilde programlamadan "öğrenmek" (yani, belirli bir görev üzerindeki performansı sürekli olarak iyileştirme) yeteneği vermek için istatistiksel teknikleri kullanan bir bilgisayar bilimi alanıdır. Yapay öğrenme, veri üzerinden tahmin edebilen ve tahminlerde bulunabilen algoritmaların çalışmasını ve geliştirilmesini araştırır.
- **Veri analizi (Data Analysis)**, verileri tanımlamak, göstermek, özetlemek ve değerlendirmek için istatistiksel ve ya mantıksal teknikleri sistematik olarak uygulama sürecidir.
- **Veri seti (dataset)** veri topluluğudur. En sık olarak bir veri seti, tablonun her sütununun belirli bir değişkeni temsil ettiği ve her satırın söz konusu veri setinin verilen bir üyesine karşılık geldiği tablo içeriğine tekabül eder.

1.2 Proje gönderimi, Dosya Yapısı ve Rapor

Bu bölümde site üzerinde yüklenen iskelet dosyasının yapısı özetlenmiştir. Proje kaynak kodları `src` dizini altında olmalıdır. Eklemek istediğiniz kaynak dosyalarını bu dizin altına eklemelisiniz.

- `dataset` dizini altında iki tane veri dosyası vardır. `dataset/house_price_data.csv` ev bilgileri ile fiyat bilgilerini içeriyor. `dataset/house_price_test.csv` içerisinde sadece ev bilgileri vardır, fiyat bilgileri eksiktir.

- `src` altında `dataset.c` ve `dataset.h` kaynak dosyalarında veri dosyası ile ilgili fonksiyon tanımları ve fonksiyonlar olmalıdır.
- `src` altında `models.h` ve `models.c` kaynak dosyaları tahmin yöntemleri ile ilgili fonksiyon ve veri tanımlarını içerir.
- `src` altındaki `main.c` ana program dosyasıdır. Fonksiyonların çağrımını ve değişken tanımlarını burada yapmanız gerekiyor.

Programınızda kullandığınız veri tanımları ve fonksiyon açıklamalarını raporunuzda ayrıntılı bir şekilde açıklayınız. Elde ettiğiniz veri sonuçlarını da raporunuza ekleyiniz. Kodlarınız ile beraber raporunuzu (PDF olarak) sisteme yükleyebilirsiniz.

Projenin notlandırılmasında sadece doğruluk değil, aynı zamanda hızı ve kaynakları nasıl kullandığı da önemli olacaktır. Algoritma ve veri yapılarını kodlarken bunları göze alarak kodlayınız.

Grup olarak yapılan projelerde tek bir kişinin sisteme yüklemesi yeterli olacaktır. Dosya adı **SOYAD1_SOYAD2_PROJE** şeklinde olup sıkıştırılmış arşiv dosyası yüklenmelidir.

Rapor içeriği ve Yapısı (20 Puan)

Raporunuz **PDF** formatında gönderilmelidir. Kapak sayfasında adınız ve soyadınız ile tarih belirtilmelidir. Düzgün bölümlere ayrılmış, başlık yapısı düzgün seçilmiş olmalıdır. Seçtiğiniz veri yapısının ve algoritmaların nedenlerini bölümlerine uygun yerlere açıklamalısınız.

Rapor en az 8 sayfadan oluşmalıdır. Genel olarak şu bölümleri içerir:

- **Giriş** Bu bölümde projenin genel konusu, yapılış amacı, raporun akışı verilir.
- **Veri Analizi** Bu bölümde verinin incelenmesi ile ilgili alanlar anlatılır. Verideki kolonların açıklaması, verinin neyi ifade ettiği gibi. Programda hangi veri yapısının neden kullanıldığı da bu bölümde anlatılır. Neden böyle bir veri yapısının seçildiği avantajları ve dezavantajları anlatılır. Yine bu bölümde veri hakkında sizden istenen soruların cevapları bulunmalıdır. Kullandığınız algoritmaların akışı, aldığı ve geri döndürdüğü parametreler ile belirtilir. Kodlayacağınız emlak programının arayüzünün açıklaması ve kullanıcı rehberi de burada anlatılmalıdır.
- **Model Tasarımı** Model ile ilgili kodladığınız veri yapıları ve algoritmalar burada açıklanır. Modellerden elde ettiğiniz sonuçların karşılaştırması, hangi modelin daha iyi sonuç verdiği bilgisi buraya eklenmelidir.
- **Sonuç** Veri analizi ve model tasarımlarından çıkarımlarınız burada özetlenmelidir.
- **Ek** Gerekli durumda ek bilgiler bu bölüm altına ekleyebilirsiniz.

Rapor içermeyen projeler kodları olmasına rağmen 0 (SIFIR) puan alacaktır.

Dosya düzenine, kodlama düzenine, istenen isimlendirmelere uymayan projelerden not kırılacaktır.

2 Veri Analizi (Toplam 35 Puan)

Veri analizi elimizdeki bir veri topluluğundan anlam çıkartabilmek için kullandığımız istatistiksel yöntemlerin genel adıdır. Bu veri topluluğunun adı **veri seti (dataset)** olarak ifade edilir. Elimizdeki veri sadece rakamlardan ve yazılardan oluşurken, veri analizi ile ondan anlam çıkartma ve sonrasında tahmin yapma imkanı ortaya çıkar.

Bu projede kullanacağımız veri seti ev satış verilerini içerir. Veri analizinde kullanacağımız kısım `data/house_price_data.csv` dosyasındadır. Veri setinde 1360 tane ev için bilgi vardır. Her bir satırda bir ev satış bilgisi bulunur. İlk satır tablonun başlık kısmıdır ve satırlardaki bilgilerin ne olduğunu gösterir:

- **ID**: kimlik değeridir. Her ev için tek bir değer atanmıştır. 101 den başlayıp, 1460 a kadar gider.
- **LotArea**: Ev alanının metrakeresini verir.
- **Street**: Evin bulunduğu sokak bilgisini verir.
- **SalePrice**: Evin dolar olarak satış fiyatıdır.
- **Neighborhood**: Evin bulunuğu komşuluk ismidir.
- **YearBuilt**: Evin inşa edildiği yıl.
- **OverallQual**: Evin malzemelerinin genel kalitesini verir. 1 ve 10 arasında doğal sayı değerleri alır. 1 değeri kalitesi en düşük evleri gösterirken, 10 kalitesi en yüksek evleri gösterir.
- **OverallCond**: Evin genel durumunu gösterir. 1 ve 10 arasında doğal sayı değerleri alır. 1 değeri durumu en kötü evleri gösterirken, 10 durumu en iyi evleri gösterir.
- **KitchenQual**: Evin mutfağının kalitesini gösterir. İki karakterlik değer alır. Aldığı değerler:
  - Ex: Çok iyi (Excellent)
  - Gd: İyi (Good)
  - TA: Ortalama (Typical/Average)
  - Fa: Uygun (Fair)
  - Po: Zayıf (Poor)

Tablo 1 örnek ev verilerini içeriyor. Her satır bir ev bilgisini verir. Veri seti dosyasında bu şekilde tanımlı 1360 ev bilgisi vardır.

Projenin ilk kısmında sizden istenen elinizdeki bu verileri dosyadan okumanız ve verileri üzerine bir analiz yapmanızdır. Veri ile ilgili işlemleri ve kodlamarı `dataset.c` ve `dataset.h` dosyaları içinde yazmanız gerekiyor.

Id	LotArea	Street	SalePrice	Neighborhood	YearBuilt	OverallQual	OverallCond	KitchenQual
101	10603	Pave	205000	NWAmes	1977	6	7	Gd
102	9206	Pave	178000	SawyerW	1985	6	5	Gd
103	7018	Pave	118964	SawyerW	1979	5	5	TA
104	10402	Pave	198900	CollgCr	2009	7	5	Gd
105	7758	Pave	169500	IDOTRR	1931	7	4	TA
106	9375	Pave	250000	Somerst	2003	8	5	Gd

Table 1: Örnek Ev Fiyatı Verisi

Veri yapı tanımları ve veriyi nasıl saklamak istediğiniz sizin tercihinizdir. Bu adımda dikkat etmeniz gereken veriye erişimi sağlayacak hızlı bir veri yapısı seçmeniz, çünkü bundan sonraki işlemlerinizi bu veri yapısı üzerinden yapacaksınız.

Bu bölümde kodladığınız fonksiyonlar ayrı ayrı şu işlemleri gerçeklemelidir:

1. **(5 Puan)** `data/data_train.csv` dosyasını okuyan ve içindekileri bir değişkende saklayan bir fonksiyon yazınız. Bu fonksiyon sadece bir dosya adı almalı ve verileri okumalıdır. Dosyadaki her ev verisi saklanmalıdır. İstediğiniz bir veri yapısını kullanabilirsiniz. Dosya okuma fonksiyon adı `read_house_data()`.

   BONUS (20 PUAN) Karma Tablosu (Hash Table) veri yapısının mantıklı bir şekilde kullanılması bonus puanı almanızı sağlar.

   Tanımlanabilecek örnek veri yapısı şu şekilde olabilir:

```
typedef struct house{
  int id;
  int lotarea;
  char* street;
  int saleprice;
  char* neighborhood;
  int yearbuilt;
  int overallqual;
  int overallcond;
  char* kitchenqual;
} House;
```

   Bu veri yapısı bir evi ifade etmek için tanımlanmıştır. Siz de bu özellikleri temsil edecek bir veri yapısı ile evi tanımlayabilir ya da bu veri yapısını kullanabilirsiniz. Fonksiyonlarda verilen örneklerde bu veri yapısı kullanılmıştır, yapmanız gereken fonksiyonları kendi yapınız için uyarlamaktır.

2. Sırasıyla veri setinden şu bilgileri döndüren fonksiyonların yazınız:

   (a) **(2 Puan)** `print_house()`: verilen tek bir evin ekrana basar.

```
House information
  ID : 662
  Lot Area : 46589
```

```
Street : Pave
Sale Price : 402000
Built year : 1994
Overall Quality : 8
Overall Condition : 7
Kitchen Quality : Gd
```

(b) **(2 Puan)** `get_house_byid()`: ID değeri verilen bir evin bilgilerini geri döndüren fonksiyon. ID ile aranan evin diğer özelliklerini geri döndürür. Örnek fonksiyon tanımı şu şekilde olabilir:

```
House get_house_byid(int id);
```

Burda fonksiyon parametre olarak bir `id` değeri alıyor ve geriye bir `House` değişkeni döndürüyor. Bu işlemi kendi veri yapınıza göre uyarlayınız.

(c) **(2 Puan)** `get_neighborhoods()`: Parametre olarak verilen bir evin komşuluğunda bulunan diğer evleri geri döndüren fonksiyon. Örnek fonksiyon tanımı şu şekilde olabilir:

```
House* get_neighborhoods(House house);
```

Burda fonksiyon parametre olarak bir `House*` değişkeni alıyor ve geriye bir `House` dizisi döndürdüğü varsayılıyor. Bu işlemi yapacak fonksiyonu yazınız.

(d) **(4 Puan)** `mean_sale_prices()` :Aldığı kriter parametresine göre bütün evlerin fiyat ortalamasını geri döndürecek fonksiyondur. Örneğin kriter `neighborhood` ise bu durumda aynı bölgede olan evlere göre gruplama yaparak fiyat ortalamasını bölgelere göre geri döndürür. Sonuçları bir dosyada saklayabilir, ya da ekrana basabilirsiniz. Örnek fonksiyon tanımı şu şekilde olabilir:

```
float* mean_sale_prices(House** houses,char* criter_name);
```

Örnek fonksiyon `House` tipinde bir dizi alıyor ve bunların `saleprice` değerlerinin ortalamasını gönderiyor.

3. **(10 Puan)** Elimizdeki evleri istenen kritere göre sıralayacak bir fonksiyon yazınız. Fonksiyon parametre olarak ev listesini ve sıralama parametresini almalı ve sıralı listeyi bir dosyaya kaydetmelidir. Bu sıralamayı veri yapınıza uygun en hızlı yapacak algoritmayı yazmalısınız.

   Örnek fonksiyon tanımı şu şekilde olabilir:

```
void sort_houses(House houses[],char* criter_name);
```

Örnek fonksiyon `char` tipinde bir parametre ve `House` tipinde bir dizi alıyor. `criter_name` olarak verdiğimiz parametre hangi kritere göre sıralama yapılacağını gösterir. Örneğin,

```
House houses[10];
// houses 10 tane House tipinde degisken tasiyan bir dizi
...
char criter_name[10] = "yearbuilt";
...
sort_houses(houses,criter_name);
```

Yukarıdaki program parçacığında `houses` dizisi 10 elemanlı bir dizidir. Bu dizinin elemanları `yearbuilt` elemanına göre sıralanmak isteniyor. Sonuçta oluşan dosyada evler en yeniden en eskiye göre sıralı olmalıdır.

(10 Puan) Yazdığınız fonksiyonları test ettikten sonra `main.c` de bunları kullanacak bir program yazınız. Programın bir emlak programı olduğunu düşünebilirsiniz. Program parametre olarak veri seti dosya adını alıp çalışmalıdır. Örneğin komut satırından şu şekilde çalıştırılabilir:

```
./main data/house_prices_train.csv
```

Kullanıcıdan istediği işlem için bilgi alıp sonuçlarını ekrana basmalı ve kullanıcı çıkış yapmadan sonlanmamalıdır. Örnek bir program ara yüzü şu şekilde olabilir:

```
Emlak Programına Hoşgeldiniz!
Yapmak istediğiniz işlemi aşağıdan seçebilirsiniz.
  1 - Evleri listele
  2 - ID değeri ile ev bul
  3 - ID değeri verilen evin komşu evlerini bul
  4 - Semtlere göre ortalama fiyatı listele
  5 - En yüksek fiyata sahip ilk 10 evi göster
  6 - Sıralı ev listesini kaydet
Programdan çıkmak için 0 a basiniz.
```

Bu programa yeni sekmelere ekleyebilirsiniz. Ornek bir program kodlaması `main.c` içinde verilmiştir. Bunu kullanabilir ya da istediğiniz gibi değiştirebilirsiniz.

3 Model ve Fiyat Tahmini(Toplam 45 Puan)

Elimizdeki ev verilerinden yola çıkarak yeni bir ev geldiğinde fiyat tahmini yapmak istiyoruz. Bu tahmini iki farklı yöntem ile yapabiliriz:

1. Evin bilgilerini sahip olduklarımız ile karşılaştırıp en yakın bilgilere sahip ev fiyatını verebiliriz.

2. Ev bilgileri ile fiyat arasında doğrusal bir ilişki bulabiliriz.

Fiyat tahmini için bu iki yöntemi kullanıp sonuçları karşılaştıracağız. Yöntemlerle ilgili fonksiyonları ve veri tanımlarını `models.c` ve `models.h` içine yazmanız gerekiyor. Burda yaptığımız tahminleri test etmek için fiyat bilgisi verilmemiş daha küçük bir ev bilgisi dosyası kullanacağız. Bu dosya `dataset/data_test.csv` içinde bulunuyor. Her iki modeli kullanıp test verisindeki evler için bir fiyat tahmini yapacaksınız, sonrasında bu tahminleri kaydedip program dosyaları ile beraber yollayacaksınız.

3.1 Yöntem 1: Ev özelliklerine göre en yakını bulma (15 Puan)

Elimizde evlerin semtlere göre, yıllara göre ve kalitesine göre fiyat dağılımları var. Elimize fiyatını bilmediğimiz bir ev geldiğinde bunun özelliklerine en yakın evleri bulup ona göre atamasını yapacağız. Burada önemli olan iki ev arasındaki benzerliği bulan denklemi oluşturmaktır.

İki ev arasındaki benzerlik bir kaç farklı değişken ile bulunabilir. Biz sırasıyla evin bulunduğu bölge (`street` ve `neighborhood`), evin toplam alanı (`lotarea`), evin inşa edildiği yıl, genel kalite değerlerini (`overallqual`,`overallcond`,`kitchenqual`) kullanacağız. Bu benzerlik fonksiyonu ev veri setindeki tüm evleri gezip, elimizdeki eve en benzer evleri seçecek ve yeni eve bu evlerin fiyat ortalamasını atayacak. Fonksiyon adı `model_by_similarity()` olmalıdır. Örnek fonksiyon çağrısı:

```
int model_by_similarity(House houses[100],House new_house);
```

Burda fonksiyon iki parametre alıyor: bir tanesi `houses` adıyla `House` elemanlarından oluşan bir dizi, diğeri de fiyat bilgisini öğrenmek istediğimiz `new_house` değişkeni.

Fonksiyon adımları şu şekilde olabilir:

1. Fiyatı öğrenilmek istenen `new_house` ile aynı komşulukta bulunan evler `houses` dizisinden seçilir. Bu işlem için daha önce yazdığınız `get_neighborhoods()` fonksiyonunu kullanınız.

2. Bu komşuluktaki evler alanlarına `lotarea` göre sıralandıktan sonra `new_house` değişkeninin alanına en yakın değere sahip en çok 10 ev seçilir. Sıralama işlemi için `sort_houses()` fonksiyonunu kullanınız. Alanlara en yakın evleri seçerken elimizdeki evin alanından örneğin 1000 m^2 daha fazla ve daha az evleri seçebilirsiniz. Örneğin aradığımızın evin metrekaresi 8500 ise alt kümede 7500 ile 9500 değerleri arasında metrekare alana sahip evleri seçebilirsiniz.

3. Alanlara ve komşuluklara göre seçilen evler bu sefer inşa yıllarına `yearbuilt` göre sıralanır. Bu listeden elimizdeki ev ile en yakın zamanlarda inşa edilmiş evler seçilir. En yakın inşa edilmiş evleri seçerken önceki bölümde yaptığınız veri analizini düşünün. Ev fiyatları ile yapım yılı arasındaki ilişki burda önemli bir rol oynayacak. Buna göre elimizdeki evin inşa yılından 5 yıl önce ve ya 5 yıl sonra inşa edilmiş evleri seçebiliriz.

4. Artık elimizde küçük bir alt küme kalmış oluyor, bundan sonra kalan üç parametreye göre seçim yapabiliriz. Bunlar `overallqual`, `overallcond` ve `kitchenqual`. Bu parametreleri ayrı ayrı değerlendirebilir ya da üçünün ortalamasını alıp kullanabilirsiniz. Eğer önceki adımlardan kalan ev sayısı 5ten az ise bu parametreler ile kontrol etmeyip sonraki tahmin adımına geçebilirsiniz.

5. Parametrelere göre ev listesini daralttıktan sonra elimizde en benzer evler kaldı. Bundan sonra yapacağımız bu evlerin ortalama fiyatını bulup yeni ev için fiyat olarak sunmak olacaktır.

Örnek bir program akışı aşağıda verilmiştir. Bize verilen evin özelliklerinin şu şekilde olabilir:

```
Id                    201
```

```
LotArea            8546
Street             Pave
Neighborhood    Edwards
YearBuilt          2003
OverallQual           4
OverallCond           5
KitchenQual          TA
SalePrice             -
```

İlk yapacağımız bu ev ile aynı komşuluktaki evleri seçmek, yani elimizdeki 1360 evden `neighborhood` değeri `Edwards` olanları seçiyoruz. Bu durumda alt kümedeki eleman sayısı 970 oluyor. Bunları alanlarına göre sıralarsak liste şu şekilde başlıyor:

Index	Id	LotArea	Street	KitchenQual	Neighborhood	YearBuilt	OverallQual	OverallCond	SalePrice
1051	1152	17755	Pave	Edwards	1959	5	4	Fa	149900
1323	1424	19690	Pave	Edwards	1966	6	7	Gd	274970
463	564	21780	Pave	Edwards	1918	6	7	TA	185000
423	524	40094	Pave	Edwards	2007	10	5	Ex	184750
1198	1299	63887	Pave	Edwards	2008	10	5	Ex	160000

Table 2: Alt Kümede Kalan Evler

Bu evler içinde alanlarına göre en yakın evleri seçeceğiz. Burda alanı elimizdeki evden 2000 metrekare az/çok olanları seçiyoruz. Yani 10546 ile 6547 metrekare arasında olan evleri seçiyoruz, böylece alt kümenin eleman sayısı 60 oluyor. Bundan sonra inşa yıllarına göre seçiyoruz. Örneğin inşa yılı 10 yıl önce ve ya 10 yıl geç olanları seçersek, geriye 7 eleman kalıyor.

Index	Id	LotArea	Street	KitchenQual	Neighborhood	YearBuilt	OverallQual	OverallCond	SalePrice
262	363	7301	Pave	Edwards	2003	7	5	Gd	198500
1154	1255	6931	Pave	Edwards	2003	7	5	Gd	165400
100	201	8546	Pave	Edwards	2003	4	5	TA	140000
920	1021	7024	Pave	Edwards	2005	4	5	Gd	176000
780	881	7024	Pave	Edwards	2005	5	5	TA	157000
17	118	8536	Pave	Edwards	2006	5	5	TA	155000
111	212	10420	Pave	Edwards	2009	6	5	Gd	186000

Table 3: Alt Kümede Kalan Evler

Burdan sonra iki farklı seçim yapabiliriz: elimizdeki kümenin yeterince yakın olduğunu varsayıp fiyat ortalamasını alabiliriz. Bu da yeni evin fiyatını 168271 yapar. Ya da devam edip kalan parametrelere göre en yakın evleri bulup ortalamalarını alabiliriz. Eğer elimizde tek bir kalırsa bu durumda o evin fiyatını atayabiliriz.

Fonksiyonu kodladığınızda doğru çalıştığından emin olmak için verilen veri seti içinden rastgele evler seçip fiyat tahmini yapmayı deneyebilirsiniz. Yaptığınız tahminler gerçek değerlere ne

kadar yaklaştı?

Fonksiyonunuzun çalıştığından emin olduktan sonra yapacağımız ikinci test `dataset/data_test.csv` kullanarak olacak. Fiyat tahmini için şu adımları yapmalısınız:

1. `dataset/data_train.csv` dosyasını okuyup içindeki ev bilgilerini saklayın.
2. `dataset/data_test.csv` dosyasını okuyup ev bilgilerini almalısınız, dikkat etmeniz gereken bu evlerin fiyat bilgileri yok, bunlara başlangıç için 0 (sıfır) ataması yapabilirsiniz.
3. `dataset/data_test.csv` dosyasındaki her bir ev için `model_by_similarity()` kullanarak bir tahmin yapın.
4. Yaptığınız fiyatları evlerin `id` değerleri ile beraber `prices_by_similarity.txt` adında bir dosyaya yazın.

Oluşturduğunuz bu dosyayı program dosyalarınız ile beraber göndereceksiniz.

3.1.1 Yöntem 2: Doğrusal bir ilişki kurma (30 Puan)

Elimizde fiyat bilgisi ile doğrusal ilişki kurabileceğimiz en önemli ev özelliği evin alanıdır. Evin alanı arttıkça fiyatının artmasını bekleriz, bu doğru orantılı bir ilişkidir. Bu ilişkiyi formülize edersek:

$$Y = f(x) + \epsilon \tag{1}$$

Denklemde x elimizdeki verileri yani bir evin alanını, Y sistemin çıktısını yani fiyatı ifade eder. ϵ ise verimizdeki gürültüyü gösterir, ancak bu proje için bunu göz ardı edeceğiz. Yapay öğrenmede yapmaya çalıştığımız, elimizdeki x ve Y değerleri arasındaki bağlantıyı en iyi ifade edecek $f()$ fonksiyonunu bulmaktır.

Figüre 1 örnek bir veri setini koordinat düzleminde gösteriyor. Düzlemde yatay eksen elimizdeki x değerlerini, dikey eksen ise y değerlerini gösteriyor.

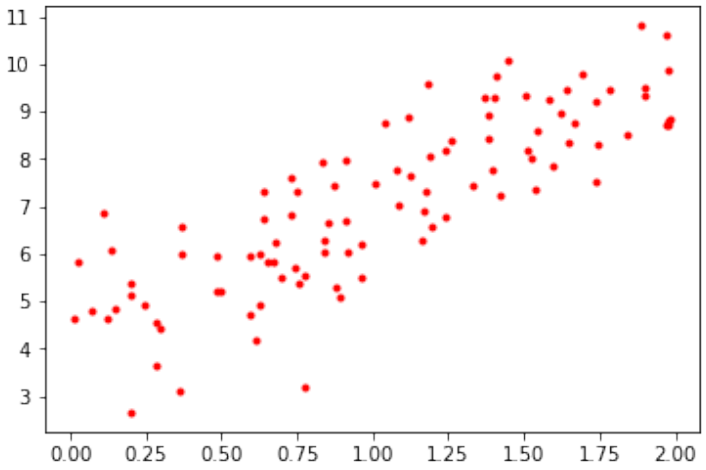

Figure 1: Örnek Veri

Grafikte anlaşılacağı gibi x ve y değerleri arasında doğrusal bir ilişki var. Burdan yola çıkarak ikisi arasında şöyle bir denklem kurabiliriz:

$$y = w_1 \times x + w_0 \tag{2}$$

Yani her bir y değeri x değerinin bir w_1 sabit sayısı ile çarpılıp w_0 sayısı ile toplanmasından oluşuyor. Öğrenmenin amacı burdaki w_1 ve w_0 sayılarını tahmin edebilmek. Eğer bu sayıları doğruya yakın bulabilirsek yeni bir x değeri geldiğinde bunun için y değerini de hesaplayabiliriz. Elimizde 100 tane (x, y) çifti olduğunu düşünürsek, tüm değerler bir doğrusal denklem sistemi oluşturur:

$$y_1 = w_1 \times x_1 + w_0 \tag{3}$$

$$y_2 = w_1 \times x_2 + w_0 \tag{4}$$

$$\vdots \tag{5}$$

$$y_{99} = w_1 \times x_{99} + w_0 \tag{6}$$

$$y_{100} = w_1 \times x_{100} + w_0 \tag{7}$$

Eğer bu denklem sistemini hesaplarsak w_1 ve w_0 değerlerini bulabiliriz. Denklem sistemini daha kullanışlı yazmak için matris formlarına dönüştürelim:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{99} \\ y_{100} \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_{99} \\ 1 & x_{100} \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

Matris formunda gösterim:

$$\mathbf{Y} = \mathbf{X} \times \mathbf{W} \tag{8}$$

Bu denklemde $\mathbf{X}$ matrisi 100×2 boyutlarında, $\mathbf{Y}$ matrisi 100×1 boyutlarında ve $\mathbf{W}$ matrisi 2×2 boyutlarındadır. $\mathbf{X}$ ve $\mathbf{W}$ matrisleri matris çarpımı ile çarpılıp $\mathbf{Y}$ elde ediliyor. Elimizdeki bu eşitlikten $\mathbf{W}$ matrisini bulmamız gerekiyor:

$$\begin{aligned} \mathbf{Y} &= \mathbf{XW} \\ \mathbf{X^T Y} &= \mathbf{X^T XW} \\ (\mathbf{X^T X})^{-1}\mathbf{X^T Y} &= (\mathbf{X^T X})^{-1}\mathbf{X^T XW} \\ (\mathbf{X^T X})^{-1}\mathbf{X^T Y} &= \mathbf{W} \end{aligned}$$

Yukarıdaki denklem adım adım $\mathbf{W}$ matrinin bulunmasını gösteriyor. Öncelikle her iki tarafı da $\mathbf{X}$ matrisinin devriği (transpose) ile çarpıyoruz. Sonrasında her iki tarafı da $(\mathbf{X^T X})^{-1}$ ile çarptığımızda denklemin sol kısmında sadece $\mathbf{W}$ matrisi kalıyor.

Figür 1 de verilen noktalar için bu matris hesabını yaptığımızda elde ettiğimiz $\mathbf{W}$ matrisi $[4.0154, 3.0193]$ oluyor. Bu matris ile ifade edilen $\mathbf{X} \times \mathbf{W}$ doğrusu ise Figüre 2 de kırmızı ile gösterilmiştir.

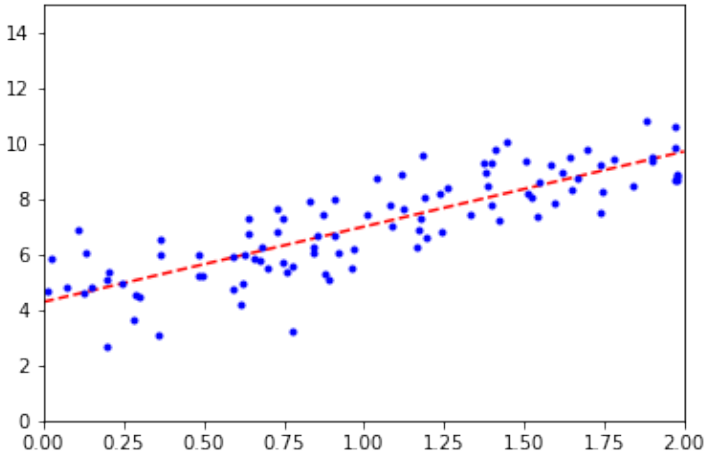

Figure 2: Örnek Veri ve Tahmin edilen doğru

Sonuç olarak elimizdeki doğrusal datayı basit bir şekilde ifade edebilecek bir model oluşturmuş olduk. Elde ettiğimiz doğru tüm veriyi ifade edemiyor, ama verinin altında yatan modeli bize sunabiliyor. **W** matrisini kullanarak şimdi yeni hesaplar yapabilir ve yeni x değerleri için tahminler yapabiliriz.

Doğrusal İlişki Tahmini - Kodlama Yöntemi Önceki bölümde anlatılan yöntemi kodlamanın temeli matris çarpımlarından oluşmaktadır. Bunun için daha önceki ödevlerde yazdığınız yöntemleri kullanabilirsiniz, özellikle Strassen Algoritması alıştırmasındaki yöntemler gerekli olabilir. Bu tahmin işlemini yapan fonksiyonları `models.c` ve `models.h` içine kodlamanız gerekiyor. Kodlama aşamasında şu adımları takip edebilirsiniz:

1. **(5 Puan)** Öncelikle elimizdeki ev bilgilerinden alan bilgisini ve fiyat bilgisini almamız gerekiyor. Eğer ev bilgileri için veri yapısı kullandıysanız sadece bu iki bilgiyi seçip gönderecek bir fonksiyon yazabilirsiniz. Bu fonksiyon adı `create_data_matrices()`. Bu fonksiyonun amacı elimizdeki evler bilgisinden iki tane matris oluşturmaktır. Bu matrislere **X** ve **Y** dersek **X** matrisi evlerin alan ölçülerini, **Y** matrisi de ilgili evin fiyatını saklar. Örneğin, Tablo 3 verilen 5 tane ev için oluşacak matrisler şu şekilde olmalıdır:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} 205000 \\ 178000 \\ 118964 \\ 198900 \\ 169500 \end{bmatrix} \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \\ 1 & x_5 \end{bmatrix} = \begin{bmatrix} 1 & 10603 \\ 1 & 9206 \\ 1 & 7018 \\ 1 & 10402 \\ 1 & 7758 \end{bmatrix}$$

   Dikkat ederseniz **X** matrisinin ilk kolonu 1 sayısı. Bu kolonun eklenmesindeki amaç **X** matrisi ile **W** matrisinin çarpımını tek işlem ile yapabilmektir.

   Ev verilerini istediğiniz veri tipinde saklayabilirsiniz. Örneğin 2-boyutlu `int` bir dizi yaratabilirsiniz, ya da tek boyutlu bir dizi yaratabilirsiniz (Strassen Algoritmasında yaptığımız gibi)

2. **(5 Puan)** Verilen bir matrisin devriğini hesaplayan fonksiyonu yazınız. Bu fonksiyon adı `get_transpose()` olmalıdır.

3. **(5 Puan)** Verilen bir matrisin tersini hesaplayan fonksiyonu yazınız. Bu fonksiyon adı `get_inverse()` olmalıdır. Fonksiyon öncelikle matrisin tersi alınıp alınamayacağını göstermeli, tersi alınımıyorsa hata verip çıkmalıdır.

4. **(5 Puan)** Verilen iki matrisin matris çarpımlarını hesaplayan fonksiyonu yazınız. Bu fonksiyon adı `get_multiplication()` olmalıdır.

5. **(5 Puan)** Yukarıdaki fonksiyonları kullanarak **W** matrisinin hesabını yapan bir fonksiyon yazınız. Fonksiyon adı `calculate_parameter()` olmalıdır. Bu fonksiyon kendisine parametre olarak verilen iki matris arasındaki doğrusal ilişkiyi sağlayan parametreyi geri döndürür. Verilen matrisleri ve önceki bölümde denklemi kullanarak **W** matrisini bulabilirsiniz.

(5 Puan) Fonksiyonları yazdıktan sonra sırasıyla `main.c` de yazdığınız emlak fonksiyonuna yeni bir sekme ekleyiniz. Örneğin,

```
Emlak Programına Hoşgeldiniz!
Yapmak istediğiniz işlemi aşağıdan seçebilirsiniz.
  1 - Evleri listele
  2 - ID değeri ile ev bul
  3 - ID değeri verilen evin komşu evlerini bul
  4 - Semtlere göre ortalama fiyatı listele
  5 - En yüksek fiyata sahip ilk 10 evi göster
  6 - Sıralı ev listesini kaydet
  7 - Evler icin fiyat tahmini yap
Programdan çıkmak için 0 a basiniz.
```

Kullanıcı bu sekmeyi seçtiğinde (7. sekme) program şu adımları takip etmelidir:

1. Eger daha onceden okunmadıysa `dataset/data_train.csv` dosyasını okumalı.

2. Elindeki listeden `create_data_matrices()` ile kullanacağımız iki matrisi oluşturmalı.

3. `calculate_parameter()` fonksiyonu ile veri setimizdeki doğrusal ilişki parametrelerini bulmalı. Bu parametre matrisi **W** olarak saklanabilir.

4. Hesaplanan parametre ve test dosya adı ile beraber program `make_prediction()` fonksiyonunu çağırır:

   (a) `dataset/data_test.csv` dosyasını okuyup, farklı bir dizide kaydetmeli Burdaki evlerin alanlarını alıp (`create_data_matrices()` kullanarak olabilir) **X** matrisi oluşturmanız gerekiyor. Bu matrisinde ilk kolonu 1 olmalıdır.

(b) Hesaplanan parametreleri kullanarak test verisinde fiyatı olmayan evler için fiyat tahmini yapmalı. Elde edilen **X** matrisi elinizdeki **W** ile matris çarpımı ile çarparsanız fiyat tahmini matrisini elde etmiş olursunuz.

(c) Yapılan tahminlerin evlerin id değerleri ile birlikte bir dosyaya kaydetmelidir. Bu değerleri bir dosyaya yazıp raporunuz ile birlikte gönderiniz.