



T.C.

ESKİŞEHİR TEKNİK ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

ENDÜSTRİ MÜHENDİSLİĞİ

ENM 421 – VERİ BİLİMİNE GİRİŞ

DR. ÖĞR. ÜYESİ ZELİHA ERGÜL AYDIN

HAZIRLAYANLAR

99735831140 Arda ÖZTÜNER

15025958656 Eray TUNÇ

30805259822 Mete Berk GÜNEY

44869010282 Yunus TÜRKŞEN

ESKİŞEHİR – 2023

Problem Tanımı: Beslenme Alışkanlıkları ve Sağlık İlişkisi Üzerine Veri Analizi

Günümüzde, beslenme alışkanlıkları ve sağlık arasındaki ilişki giderek önem kazanmaktadır. Bu bağlamda USDA'nın FoodData Central web sitesinden elde edilen geniş beslenme verilerini kullanarak çeşitli beslenme modellerini değerlendirmek ve sağlık üzerindeki etkilerini anlamak amacıyla bir analiz gerçekleştirmeyi hedefliyoruz.

Araştırma Hedefleri:

1. Besin Değerlerinin Karşılaştırılması: Farklı besin kategorileri arasında kalori, protein, karbonhidrat, yağ, kolesterol ve diğer besin değerleri üzerindeki varyasyonu inceleyerek besin profillerini karşılaştırmak.
2. Dengeli Beslenme Modeli: Genel olarak sağlıklı bir beslenme modeli için gerekli olan temel besin öğelerini belirleyerek dengeli bir beslenme modelinin ne şekilde olması gerektiğini analiz etmek
3. Besin Çeşitliliği ve Sağlık İlişkisi: C vitamini alımının maksimize edilmesi ve besin çeşitliliğinin sağlık üzerindeki etkilerini araştırarak, çeşitli besinlerin sağlık açısından önemini belirlemek.
4. Diyet ve Hidrasyon İlişkisi: Kalori ve su içeriği arasındaki korelasyonu inceleyerek, beslenme alışkanlıklarının enerji ve hidrasyon dengesine etkilerini değerlendirmek
5. Beslenme Düzenlerinin İncelenmesi: Sıfır karbonhidratlı ve yüksek proteinli diyetlerin sakıncalarını ele alarak, dengeli bir beslenme yaklaşımının sağlık üzerindeki avantajlarını vurgulamak.
6. Kalori Tahmin Modelleri: Geleneksel kalori tahmin modellerine meydan okuyarak, besin değerlerine dayalı daha güvenilir ve doğru kalori tahmin modellerini geliştirmek.
7. Bu problem tanımı doğrultusunda yapılacak analizler, beslenme alışkanlıklarının sağlık üzerindeki etkilerini daha iyi anlamamıza ve bireylere daha sağlıklı yaşam tarzlarına yönlendirmelerine yardımcı olacak önemli bilgiler sunabilir.

FoodData Merkezi Hakkında:

Gıda arzı ve diyetle alınan gıdalar ile sağlık arasındaki ilişkilerin bilimsel olarak anlaşılması yıllar içinde gelişmiştir. USDA'nın gıda bileşimi veri kaynakları da araştırmacılar, politika yapıcılar, beslenme ve sağlık uzmanları ve ürün geliştiriciler dahil olmak üzere çeşitli kullanıcıların ihtiyaçlarını karşılamak için gelişmektedir.

Son yıllarda, gıda arzında hızla artan değişim hızı ve gıda verilerinin artan kullanım çeşitliliği, gıdalar ve gıda ürünlerindeki besinler ve diğer bileşenler hakkında şeffaf ve kolay erişilebilir bilgilere olan ihtiyacı büyük ölçüde artırmıştır. Bu ihtiyaç, gıda profili bilgilerinin bilimsel olarak titiz bir şekilde analiz edilmesi, derlenmesi ve sunulması için yeni bir yaklaşım gerektirmektedir. FoodData Central bu yeni yaklaşımı somutlaştırmaya devam ediyor.

Keşifçi Veri Analizi:

Veri Setini Anlama:

Beslenme verilerini USDA'nın FoodData Central web sitesinden alınıyor. Bu veriler, 7.793 yaygın gıdanın kalori içeriğinin yanı sıra besin bileşimlerini de içerir. Her satır bir gıda maddesini temsil eder ve besin değerleri 100 g porsiyona dayanır. İşte sütunların bir açıklaması:

- **FDC_ID:** Veri tabanındaki her gıda maddesi için benzersiz bir tanımlayıcı. (Object)
- **Item:** Gıda ürününün adı veya açıklaması. (Object)
- **Category:** Gıda maddesinin kategorisi veya sınıflandırması, örneğin "Fırınlanmış Ürünler" veya "Sebzeler ve Sebze Ürünleri". (Object)
- **Calories:** Gıdanın kilokalori (kcal) cinsinden sunulan enerji içeriği. (Object)
- **Protein:** Gıdanın gram cinsinden ölçülen protein içeriği. (Object)
- **Carbohydrate:** Gıdanın gram cinsinden ölçülen karbonhidrat içeriği. (Object)
- **Total fat:** Gıdanın gram cinsinden ölçülen toplam yağ içeriği. (Object)
- **Cholesterol:** Gıdanın miligram cinsinden ölçülen kolesterol içeriği. (Object)
- **Fiber:** Gıdanın gram cinsinden ölçülen diyet lifi içeriği. (Object)
- **Water:** Gıdanın gram cinsinden ölçülen su içeriği. (Object)
- **Alcohol:** Gıdanın alkol içeriği (varsa), gram cinsinden ölçülür. (Object)
- **Vitamin C:** Gıdanın miligram cinsinden ölçülen C Vitamini içeriği. (Object)

Not: FDC_ID başlangıçta indeks olarak ayarlanmıştır.

Veri setinin boyutu:

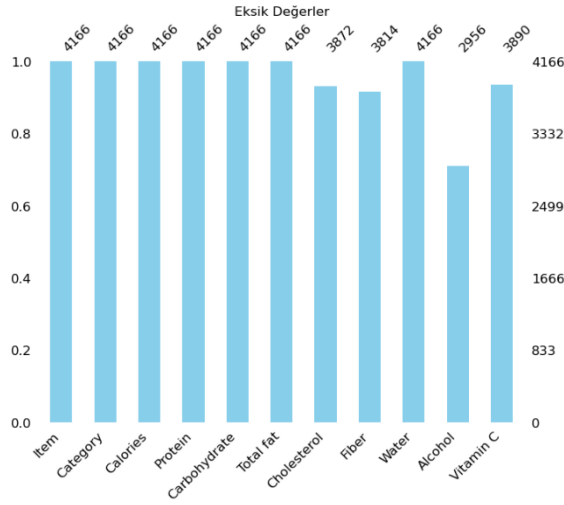
Satır Sayısı: 4166
Sütun Sayısı: 12

İlk İstatistiksel Özetler:

	Calories (kcal)	Protein (g)	Carbohydrate (g)	Total_fat (g)	Cholesterol (mg)	Fiber (g)	Water (g)	Alcohol (g)	Vitamin C (mg)	Real_Cal
count	4166.000000	4166.000000	4166.000000	4166.000000	4166.000000	4166.000000	4166.000000	4103.000000	4166.000000	4166.000000
mean	228.715795	8.710583	22.626500	11.641534	35.446023	1.734898	55.165540	0.226103	12.258689	230.122139
std	195.020692	10.131239	26.859411	19.188625	102.479961	3.321021	32.490975	2.135712	81.073235	197.916444
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	68.000000	1.352500	1.912500	0.370000	0.000000	0.000000	26.705000	0.000000	0.000000	67.495000
50%	176.000000	4.540000	9.400000	4.145000	2.000000	1.000000	65.040000	0.000000	0.700000	172.555000
75%	357.000000	13.182500	42.960000	15.000000	49.000000	2.200000	83.397500	0.000000	11.600000	364.525000
max	902.000000	85.600000	100.000000	100.000000	2552.000000	77.300000	100.000000	42.500000	2732.000000	900.260000

Not: Bu değerler veri türünün objeden float'a dönüştürüldükten sonra alınmıştır.

Eksik Verilerin Görselleştirilmesi:



Item	0
Category	0
Calories	0
Protein	0
Carbohydrate	0
Total fat	0
Cholesterol	294
Fiber	352
Water	0
Alcohol	1210
Vitamin C	276

Veri setindeki ilk problemimiz bütün veri özniteliklerinin obje tipinde olmasıdır. İlk ön işleme olarak bu tüm veri başlıklarının türü float'a çevrilmiş ve sütun isimleri değiştirilmiştir. Bu işlem tüm sütunlar için yapılmıştır.

Örnek olarak:

SÜTUN TIPLERİ VE İSİMLERİ DEĞİŞTİRME

```
: for index, value in df['Vitamin C'].items():
    if isinstance(value, str) and 'mg' in value:
        df.at[index, 'Vitamin C'] = value.replace(' mg', '')
df["Vitamin C"] = df["Vitamin C"].astype(float)
df = df.rename(columns={'Vitamin C': 'Vitamin C (mg)'})
```

Sonuç olarak:

Öncesi:

```
Data columns (total 11 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Item         4166 non-null    object
1   Category     4166 non-null    object
2   Calories      4166 non-null    object
3   Protein       4166 non-null    object
4   Carbohydrate  4166 non-null    object
5   Total fat     4166 non-null    object
6   Cholesterol   3872 non-null    object
7   Fiber         3814 non-null    object
8   Water         4166 non-null    object
9   Alcohol       2956 non-null    object
10  Vitamin C     3890 non-null    object
dtypes: object(11)
```

Sonrası:

```
Data columns (total 11 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Item         4166 non-null    object
1   Category     4166 non-null    object
2   Calories (kcal)  4166 non-null    float64
3   Protein (g)    4166 non-null    float64
4   Carbohydrate (g)  4166 non-null    float64
5   Total_fat (g)  4166 non-null    float64
6   Cholesterol (mg)  3872 non-null    float64
7   Fiber (g)      3814 non-null    float64
8   Water (g)      4166 non-null    float64
9   Alcohol (g)    2956 non-null    float64
10  Vitamin C (mg)  3890 non-null    float64
dtypes: float64(9), object(2)
```

Bir sonraki adımda eksik verilerin doldurulması yer almıştır. NaN değerlerin doldurulmasında pandas sonrasında ise tahmin edilerek eksik değerlerin doldurulması işlemi için sklearn kütüphanesinden Simple Imputer kullanılmıştır.

```
df.loc[df["Category"]=="Baby Foods", "Alcohol (g)"] = df[df["Category"]=="Baby Foods"]["Alcohol (g)"].fillna(0)
df.loc[df["Category"]=="Baked Products", "Alcohol (g)"] = df[df["Category"]=="Baked Products"]["Alcohol (g)"].fillna(0)
```

```
: df["Vitamin C (mg)"].isnull().sum()
```

```
: 276
```

```
: from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
df['Vitamin C (mg)'] = imputer.fit_transform(df[['Vitamin C (mg)']])
df["Vitamin C (mg)"].isnull().sum()
```

```
: 0
```

Kullanılan Makine Öğrenmesi Algoritmaları:

1. Linear Regression (Doğrusal Regresyon):

Doğrusal regresyon, bağımlı değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi modeller. İki değişken arasındaki ilişkiyi temsil eden bir doğru denklemi bulmaya çalışır.

2. Statsmodels ile Çoklu Regresyon:

Statsmodels, Python'da istatistiksel modeller oluşturmak için kullanılan bir kütüphanedir. Çoklu regresyon, birden fazla bağımsız değişkenin kullanıldığı regresyon analizidir.

3. Support Vector Regression (Destek Vektör Regresyonu):

Support Vector Regression, doğrusal olmayan regresyon problemlerini çözmek için kullanılır. Veri noktalarını bir hiper düzlem etrafında gruplamaya çalışır ve regresyon modelini bu düzlemle optimize eder.

4. Decision Tree Regressor (Karar Ağacı Regresyonu):

Karar ağacı, veri kümesini belirli kurallarla bölerek ve her bölge için bir regresyon modeli oluşturarak regresyon problemlerini çözer.

5. Random Forest Regressor (Rastgele Orman Regresyonu):

Rastgele Orman, birçok karar ağacının birleşimi olarak çalışır. Her ağaç veri kümesinin farklı alt kümeleri üzerinde eğitilir ve sonuçlar birleştirilir.

6. Gradient Boosting Regressor (Gradyan Artırma Regresyonu):

Gradyan Artırma, zayıf öğrenen modelleri (genellikle karar ağaçları) birbirine ekleyerek güçlü bir model oluşturur. Her aşamada hata düzeltilir.

Gerçek besin değerlerinin farklı regresyon modelleri ile tahminlenmesi:

Öncelikli olarak veri setinin eğitim ve test olarak bölünmesi,

```
X=df_model.loc[:,["Protein (g)","Carbohydrate (g)","Total_fat (g)"].values  
y=df_model.loc[:, "Calories (kcal)"].values.reshape(-1,1)
```

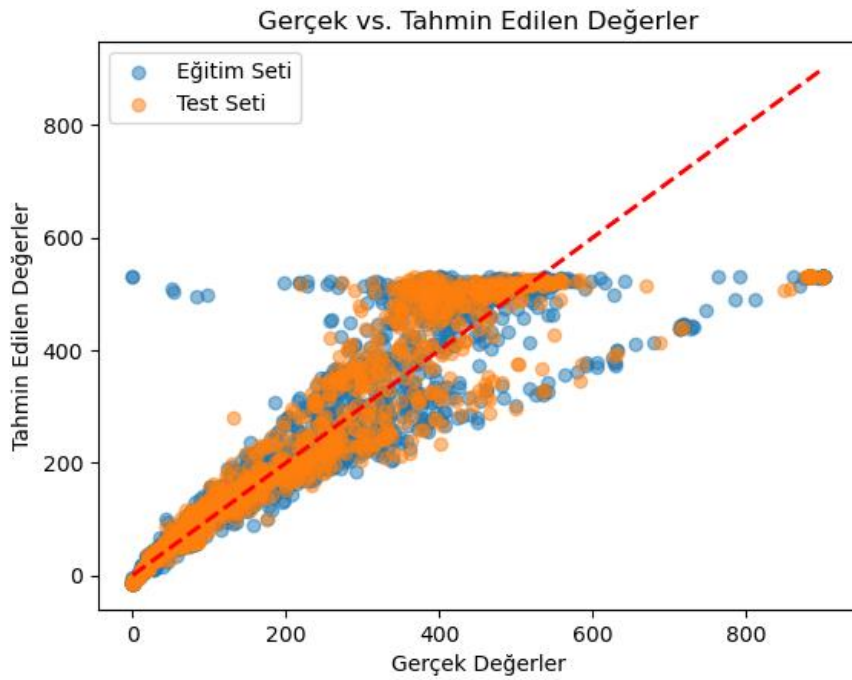
```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3,random_state=42,shuffle=True)
```

Sonrasında ise çeşitli modellerin denenmesi

1. Linear Regression (Doğrusal Regresyon):

İlk denemede alınan sonuçlar ve görselleştirilmiş hali aşağıdaki gibidir.

- Mean Squared Error (MSE): 6432.841880597647
- Root Mean Squared Error (RMSE): 80.20499909979208
- R-squared (R^2): 0.8228289307082204



2. Statsmodels ile Çoklu Regresyon:

Bağımlı ve bağımsız değişkenlerin belirlenmesi ve çoklu regresyon uygulanması,

```
y=df["Calories (kcal)"]
x1=df[["Protein (g)","Carbohydrate (g)","Total_fat (g)"]]

x=sm.add_constant(x1)
results=sm.OLS(y,x).fit()

print("Katsayılar:")
print(results.params)

Katsayılar:
const          2.959282
Protein (g)     4.058511
Carbohydrate (g) 3.851515
Total_fat (g)   8.869811
dtype: float64

# R-squared değeri
r_squared = results.rsquared
print("R-squared:", r_squared)

# Adjusted R-squared değeri
adjusted_r_squared = results.rsquared_adj
print("Adjusted R-squared:", adjusted_r_squared)

R-squared: 0.9904241594117287
Adjusted R-squared: 0.990417257075889
```

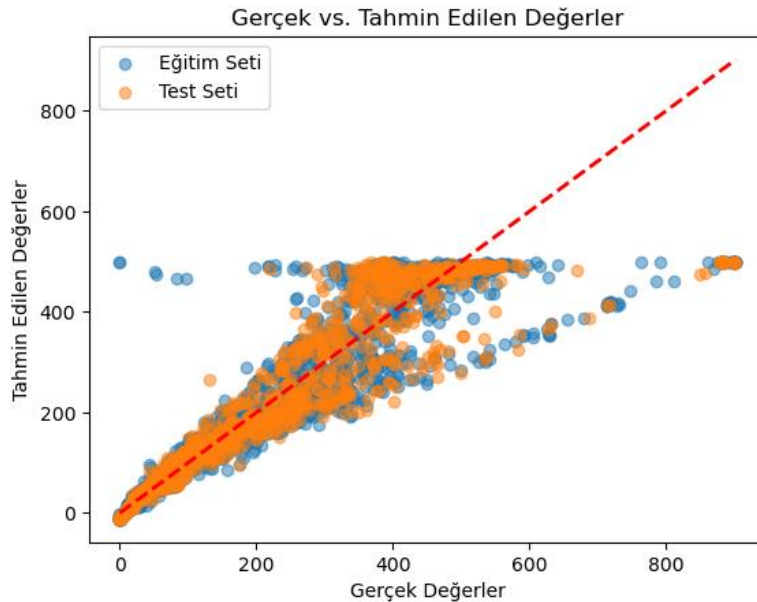
OLS Regression Results

Dep. Variable:	Calories (kcal)	R-squared:	0.990			
Model:	OLS	Adj. R-squared:	0.990			
Method:	Least Squares	F-statistic:	1.435e+05			
Date:	Fri, 29 Dec 2023	Prob (F-statistic):	0.00			
Time:	22:39:23	Log-Likelihood:	-18196.			
No. Observations:	4166	AIC:	3.640e+04			
Df Residuals:	4162	BIC:	3.642e+04			
Df Model:	3					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	2.9593	0.534	5.540	0.000	1.912	4.007
Protein (g)	4.0585	0.030	136.099	0.000	4.000	4.117
Carbohydrate (g)	3.8515	0.011	341.875	0.000	3.829	3.874
Total_fat (g)	8.8698	0.015	574.418	0.000	8.840	8.900
Omnibus:	3211.870	Durbin-Watson:	1.199			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2439237.713			
Skew:	2.396	Prob(JB):	0.00			
Kurtosis:	121.445	Cond. No.	65.8			

Cleveland Clinic web sitesine göre, bir gram yağ yaklaşık 9 kilokalori, bir gram protein ve bir gram karbonhidrat ise 4'er kilokalori içerir. Bizim sonuçlarımız doğruluğu ise 0,99 olduğundan oldukça başarılı bir tahminleme olmuştur.

3. Support Vector Regression (Destek Vektör Regresyonu):

- SVR Mean Squared Error: 6502.934580728067
- R-squared (R^2): 0.8208984622057829

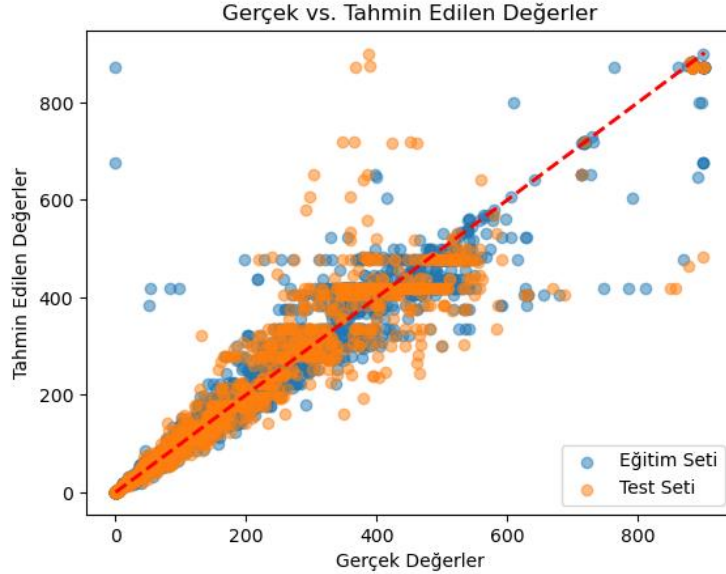


Daha çok doğrusal olmayan modellerde kullan SVR çok başarılı olamamıştır.

4. Decision Tree Regressor (Karar Ağacı Regresyonu):

- Decision Tree Mean Squared Error: 4480.213676639835
- R-squared (R^2): 0.8766075301586319

Destek Vektör Regresyonuna göre daha başarılı bir sonuç alınmıştır.



5. Random Forest Regressor (Rastgele Orman Regresyonu):

- Random Forest Mean Squared Error: 3476.905331685452
- R-squared (R^2): 0.9042402958327074

Her ağaç veri kümesinin farklı alt kümeleri ile eğitildiğinden karar ağacı regresyonuna göre daha iyi sonuç vermiştir.

6. Gradient Boosting Regressor (Gradyan Artırma Regresyonu):

- Mean Squared Error : 3676.8316839168897
- R-squared (R^2): 0.8987339945335437

İyi bir R^2 skoru elde edilmesine rağmen MSE oranı oldukça yüksektir.

Sonuç:

Birden fazla algoritma denenmesi ile en iyi sonucu Çoklu Lineer Regresyon vermiştir. R^2 Skorları iyi olmasına rağmen MSE skorlarının oldukça yüksek çıkmasının sebepleri,

- Outlier (Aykırı Değer) Etkisi:
 - R^2 skoru, outlier'ların etkisine karşı oldukça dirençlidir. Ancak MSE, outlier'ların etkisini karesel olarak büyüttüğü için outlier'lar MSE'yi yüksek çıkarabilir.
- Ölçek Farklılıkları:
 - R^2 skoru, bağımsız değişkenlerin ve bağımlı değişkenin ölçeğine karşı oldukça dirençlidir. Ancak MSE, hata karelerini topladığı için ölçek farklılıklarına daha duyarlıdır.

- **Hangi meyve en yüksek C vitamini içeriğine sahiptir?** Sorusunun cevabı yapılan analiz sonucu Papaya meyvesi çıkmıştır. Sonrasında ise Elma Sosu ve Guava meyvesidir.

```
def findMaxOfVitic(str):
    return ab[ab["Item"].str.contains(str,case=False,regex=True)][["Vitamin C (mg)"]].max()
maxsOfVitic=pd.DataFrame([findMaxOfVitic("papaya"), findMaxOfVitic("guava"),findMaxOfVitic("peaches"), findMaxOfVitic("pineapple"),findMaxOfVitic("apple"),findMaxOfVitic("orange"),findMaxOfVitic("lemon"),findMaxOfVitic("lime"),findMaxOfVitic("grape"),findMaxOfVitic("strawberry"),findMaxOfVitic("blueberry"),findMaxOfVitic("raspberry"),findMaxOfVitic("blackberry"),findMaxOfVitic("kiwi"),findMaxOfVitic("mango"),findMaxOfVitic("peach"),findMaxOfVitic("nectarine"),findMaxOfVitic("plum"),findMaxOfVitic("cherry"),findMaxOfVitic("apricot"),findMaxOfVitic("pear"),findMaxOfVitic("apple sauce"),findMaxOfVitic("orange juice"),findMaxOfVitic("lemon juice"),findMaxOfVitic("lime juice"),findMaxOfVitic("grape juice"),findMaxOfVitic("strawberry juice"),findMaxOfVitic("blueberry juice"),findMaxOfVitic("raspberry juice"),findMaxOfVitic("blackberry juice"),findMaxOfVitic("kiwi juice"),findMaxOfVitic("mango juice"),findMaxOfVitic("peach juice"),findMaxOfVitic("nectarine juice"),findMaxOfVitic("plum juice"),findMaxOfVitic("cherry juice"),findMaxOfVitic("apricot juice"),findMaxOfVitic("pear juice"),findMaxOfVitic("apple juice"),findMaxOfVitic("orange juice"),findMaxOfVitic("lemon juice"),findMaxOfVitic("lime juice"),findMaxOfVitic("grape juice"),findMaxOfVitic("strawberry juice"),findMaxOfVitic("blueberry juice"),findMaxOfVitic("raspberry juice"),findMaxOfVitic("blackberry juice"),findMaxOfVitic("kiwi juice"),findMaxOfVitic("mango juice"),findMaxOfVitic("peach juice"),findMaxOfVitic("nectarine juice"),findMaxOfVitic("plum juice"),findMaxOfVitic("cherry juice"),findMaxOfVitic("apricot juice"),findMaxOfVitic("pear juice"),findMaxOfVitic("apple juice")])
maxsOfVitic.columns=["Vitamin C (mg)"]
maxsOfVitic.sort_values("Vitamin C (mg)",ascending=False)
```

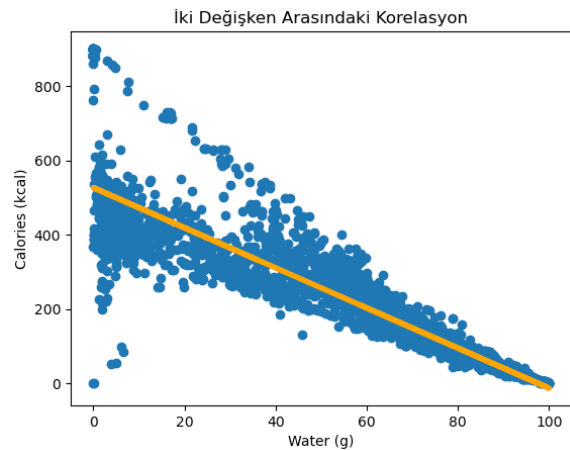
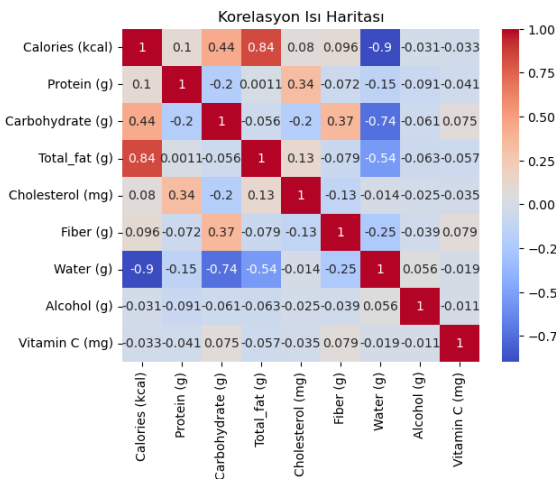
	Vitamin C (mg)
papaya	113.1
applesauce	113.1
guava	80.9
peache	46.1

En çok C vitamini içeren diğer C vitamini kaynaklarımız ise;

Item	Category	Calories (kcal)	Protein (g)	Carbohydrate (g)	Total_fat (g)	Cholesterol (mg)	Fiber (g)	Water (g)	Alcohol (g)	Vitamin C (mg)
"Babyfood, GERBER, 2nd Foods, apple, carrot and squash, organic"	Baby Foods	64.0	1.10	14.82	0.00	2.000000	1.2	83.83	0.0	2732.0
"Beverages, fruit-flavored drink, powder, with high vitamin C with other added vitamins, low calorie"	Beverages	227.0	0.25	91.00	0.16	0.000000	2.2	1.32	0.0	2400.0
"Beverages, Orange-flavor drink, breakfast type, low calorie, powder"	Beverages	217.0	3.60	85.90	0.00	0.000000	3.8	1.80	0.0	2400.0
"Peppers, sweet, red, freeze-dried"	Vegetables and Vegetable Products	314.0	17.90	68.70	3.00	0.000000	21.3	2.00	0.0	1900.0

Bebek mamaları, meyve aromalı içecekler ve sebze ve sebze içeren ürünler olmuştur.

- Bir gıda maddesinin kalorisi ve su içeriği arasındaki ilişkisi ise;



-0.9002 Korelasyon oranı ile güçlü negatif korelasyon ortaya çıkmıştır. Bunu;

Su Tüketimi Artarken Kalori Azalıyor mu?

Eğer su tüketimi arttıkça kalori alımı azalıyorsa, bu olumlu bir durum olabilir. Sağlıklı bir beslenme programında su tüketimi genellikle önerilirken, aşırı kalori alımının önlenmesi de sağlıklı bir yaşam tarzının bir parçasıdır.

- Beslenme Seçimleri: Belki de su içmek, içecekler arasında daha düşük kalorili bir seçenek olduğu için yüksek kalorili içeceklerin tüketimini azaltıyor.
- Sağlıklı Yaşam Tarzı İlişkisi: Genel olarak sağlıklı bir yaşam tarzı sürdüren bireyler, su içmeye ve düşük kalorili beslenmeye daha fazla eğilim gösterebilir.

Şeklinde açıklayabiliriz.

- **Sıfır karbonhidrat diyeti yapan bir bireyin yediği besinler ve yapmayan birinin yediği besinlerin karşılaştırılması**

Sıfır Karbonhidrat Diyeti:

Temel İlke:

Sıfır karbonhidrat diyeti, günlük karbonhidrat alımını minimal seviyelere indirerek vücudun enerjiyi genellikle yağlardan elde etmesini amaçlar. Bu diyet, karbonhidratların büyük ölçüde kesildiği veya sıfırlandığı bir beslenme planını içerir.

Anahtar Özellikler:

- Karbonhidrat içeren gıdalar (ekmek, makarna, şeker, meyveler vb.) sınırlı veya tamamen çıkarılır.
- Temel enerji kaynağı yağlar ve proteinlerdir.
- Yüksek yağlı ve proteinli gıdalar (et, balık, yumurta, avokado, yağlı tohumlar) öne çıkar.
- Vücut, ketozis denilen bir duruma geçebilir, bu da yağların enerji olarak kullanıldığı bir süreçtir.

Avantajlar:

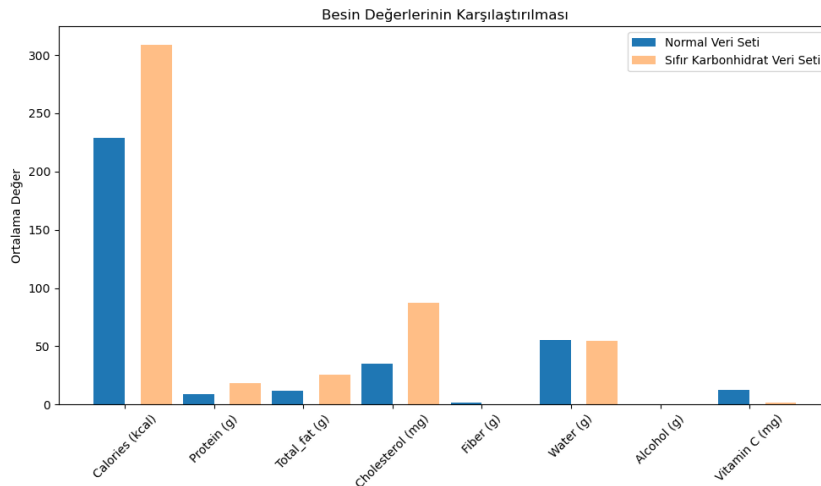
Kilo kaybı: Karbonhidrat sınırlaması, kilo kaybına katkıda bulunabilir.

İnsülin düzeyleri: Karbonhidrat alımının azaltılması, insülin seviyelerini düzenleyebilir.

Dezavantajlar:

Besin çeşitliliği sınırlaması: Bazı besin gruplarının sınırlanması, bazı vitamin ve minerallerin eksik alınmasına neden olabilir.

Kısa vadeli enerji eksikliği: Başlangıçta, vücut ketozise geçtiğinde enerji düzeylerinde dalgalanmalar olabilir.



Yüksek Proteinli Diyet:

Temel İlke:

Yüksek proteinli diyet, günlük protein alımını artırarak kas kütleini korumayı ve artırmayı amaçlar. Bu diyet, genellikle karbonhidrat ve yağ alımını belirli bir oranda sınırlar, ancak protein alımını vurgular.

Anahtar Özellikler:

- Protein kaynakları öne çıkar: Et, tavuk, balık, yumurta, süt ürünleri, baklagiller gibi yüksek proteinli gıdalar diyetin temelini oluşturur.
- Karbonhidrat ve yağ miktarı kontrol altında tutulabilir.
- Sporcular ve kas kütlesi artırmak isteyen bireyler tarafından sıklıkla tercih edilir.

Avantajlar:

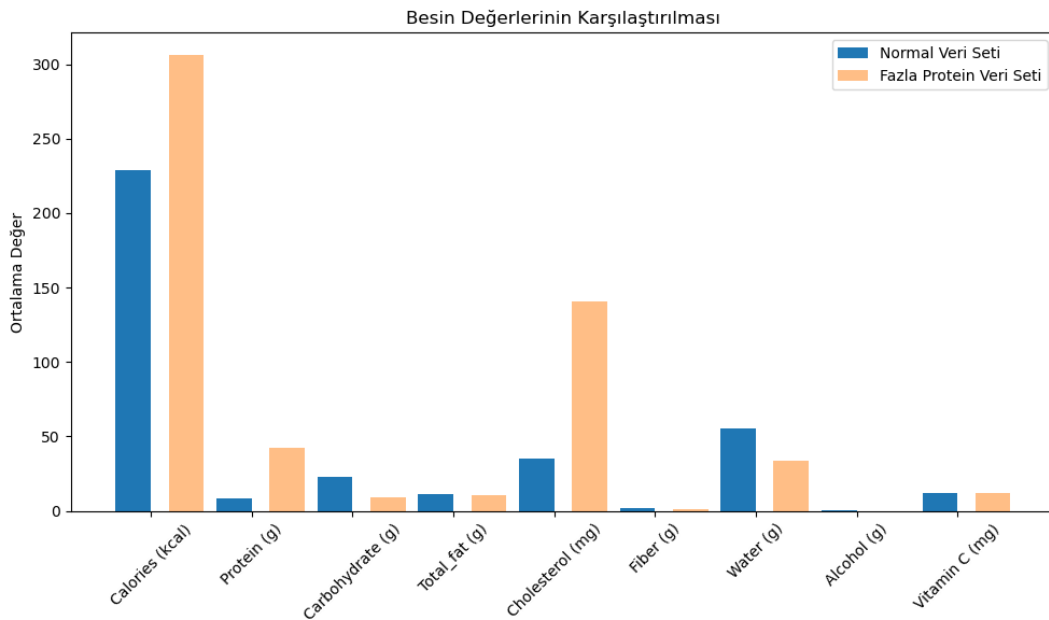
Kas kütlesi artışı: Yüksek protein alımı, kas kütleini artırabilir.

Tokluk hissi: Protein, tokluk hissi sağlayarak kilo kontrolüne yardımcı olabilir.

Dezavantajlar:

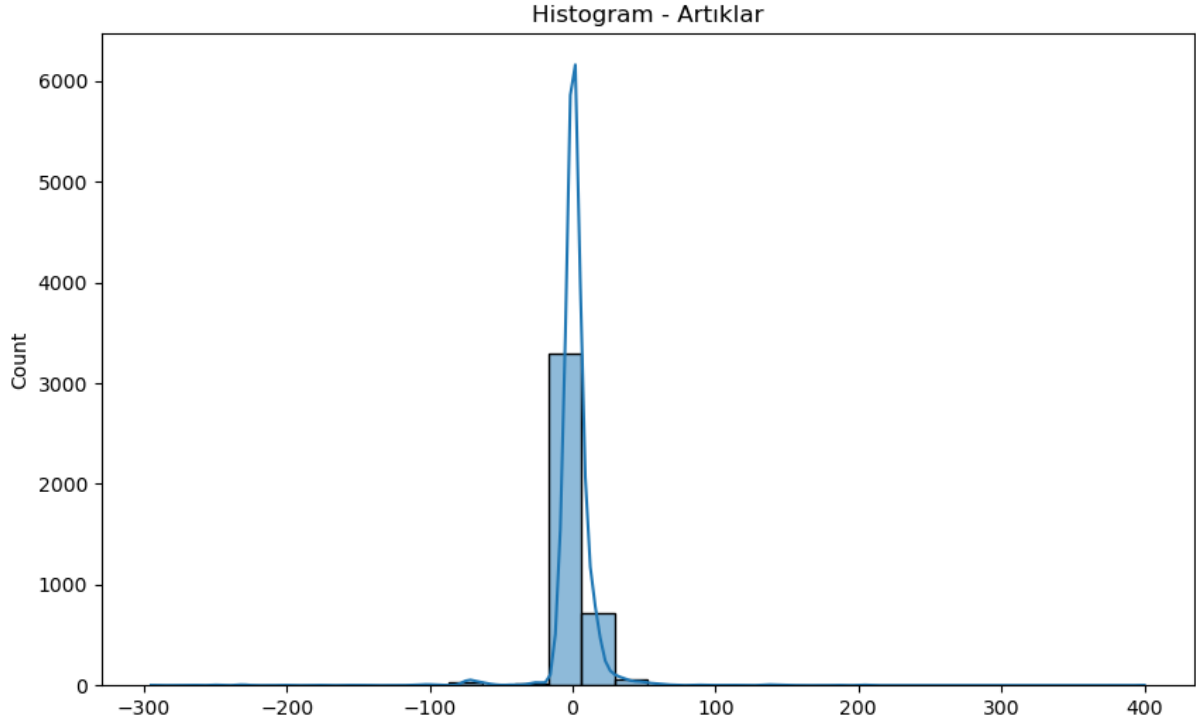
Fazla protein alımı: Çok fazla protein alımı bazı insanlar için sindirim sorunlarına neden olabilir.

Yağ ve karbonhidrat eksikliği: Bu diyetin aşırı uygulanması, yağ ve karbonhidrat alımını sınırlayarak gerekli enerji sağlanmasını engelleyebilir.



Artık Analizi:

Regresyon modelleri tahminlerle gerek deęerler arasındaki farklara "artık" denir. Artıkların analizi, modelin ne kadar iyi alıřtıęını deęerlendirmek ve modeldeki iyileřtirmeleri belirlemek amacıyla yapılır.



Sonuç:

Doęrusal model, verilen zelliklere dayanarak makul bir kalori tahmini sunsa da artıkların analizi, kalori ierięini etkileyen ve modelde hesaba katılmayan faktrlerin olduęunu gstermektedir. Bu durum, kalori tahminlerinin belirli noktalarda sapma gsterdięini ve modelin tam bir aıklama yapmadıęını iřaret eder. Kalori tahminlerinin doęruluęunu artırmak ve gıdalardaki gizli kalori kaynaklarını ortaya ıkarmak amacıyla, muhtemelen daha karmařık modelleme teknikleri veya ek zelliklerin dahil edilmesini ieren daha detaylı bir arařtırma yapılması nerilmektedir. Bu analiz, beslenme bilimine katkıda bulunmak ve daha etkili kalori tahminleme modelleri geliřtirmek adına nemli bir fırsat sunmaktadır.