



GUI Veri Analizi Uygulaması

Fen Bilimleri Enstitüsü Yazılım Mühendisliği Ana Bilim Dalı
Dönem Projesi

Doğancan Pehlivan
Y220234133

Proje Danışmanı: Prof. Dr. Aytuğ Onan

Haziran 2024

GUI Veri Analizi Uygulaması

Öz

Bu makalede bir firmanın bilgisayar sayımı ve onun yanında hurda bilgisayar ya da kullanılabilir bilgisayar olarak ikiye ayrılan kod dizini mevcuttur. Özgü olarak Machines.csv veri setimiz bulunmaktadır. Veri setimiz tam olarak 5002 adettir. Bilgisayar ayırımı olarak pc isimleri mevcut olup isimleri aşağıdaki gibidir.

PC isimleri ve anlamları:

VEHEDD (D): Desktop pc

VEHEDL (L): Leptop pc

VEHEDA (A): A10 pc anlamlarına gelir.

Veri setimizdeki arızalar LCD ve Batarya olarak işlenmiştir. LCD arızaları değişim, Batarya arızaları hurda olarak senaryolanmış ve Python programlama dili kullanılmıştır.

Yapılan işlemler ise görsel olarak tablolanmıştır.

Anahtar Sözcükler: Desktop(D), Laptop (L), All-in-One (A), Machines.csv , LCD ve BATARYA, Veri analizi, işlemlerin kodlanması, Tablolama, Grafikler, VEHEDD, VEHEDA, VEHEDL

*NOT: Uygulamanın devamı eklenmiştir. (***3.DÖNEM BİTİRME PROJESİ PART II**)

Sayfa (6) kısımdan itibaren projenin geliştirilmiş kısmı görseller ve açıklamalar ile anlatılmıştır.

GUI Data Analysis Application

Abstract

In this article we present a computer census of a company and its code directory, which is divided into two parts: scrap computers and usable computers. Specifically, we have the Machines.csv dataset. The dataset contains exactly 5002 machines. PC names are available as computer distinction and their names are as follows.

PC names and their meanings:

VEHEDD (D): Desktop pc

VEHEDL (L): Leptop pc

VEHEDA (A): It stands for Aio pc.

The failures in our data set are processed as LCD and Battery. LCD failures were scripted as replacement, Battery failures were scripted as scrap and Python programming language was used.

The operations performed are visually tabulated.

Keywords: Desktop (D), Laptop (L), All-in-One (A), Machines.csv , LCD and BATTERY, Data analysis, Coding of operations, Tabulation, Graphics, VEHEDD, VEHEDA,VEHEDL

İçindekiler

Öz	i
Abstract	ii
1 Giriş Yöntem ve Bulgular	1
2 Csv Data Analyzer GUI Modu Bitirme Projesi	6
2.1 Veri Ekleme	7
2.2 Veri Silme	7
2.3 Çoklu Güncelleme.....	7
2.4 Excel Formatında Dışarı Aktarma	8
2.5 Uygulama İçi Filtreleme Sistemi	9
2.6 Excel Tipi Filtreleme.....	10
2.7 Jira Tipi Filtreleme	10
2.8 Veri Analizi.....	11
3.Modüller ve Class Kodları.....	12
3.1 Verileri Analiz Et Butonu	13
3.2 VB.Net F4 Python Fonksiyonu	13
3.3 F5 Python Fonksiyonu	14
3.4 F3 Python Fonksiyonu	15
4.Database işlemleri	15
4.1 Load Data	16
4.2 Remove Data.....	17
4.3 Register Data.....	18
4.4 Update Data.....	18
Kaynaklar	20

1.Giriş Yöntem Ve Bulgular

Bu Python programının yapılış amacı; Bilgisayarların fazla olduğu firmada hangi bilgisayarın hurda hangi bilgisayarın kullanıma devam etmesi gerektiği hakkında yapılan bir çalışmadır.

Yukarıdaki özetle de belirttiğimiz gibi Batarya ve LCD arıza olarak ikiye ayrılmıştır. Bununla birlikte söz konusu olan bilgisayarın arıza tipine göre hurdaya veya parça değişimine gidilmektedir. Bu çalışmanın asıl amacı maliyet ve uzun süreli ürün kullanımınıdır. Aşağıda yazılan kod ve açıklamaları mevcuttur.

Yazılan Kod Parçalarının Açıklamaları:

Section 1

```
import numpy as np
import pandas as pd |
import os
for dirname, _, filenames in os.walk('../DataSet'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
../DataSet\DataSet.ipynb
../DataSet\id.txt
../DataSet\MachineGenerator.py
../DataSet\Machines.csv
../DataSet\Machines.txt
../DataSet\TempCode.txt
../DataSet\.ipynb_checkpoints\DataSet-checkpoint.ipynb
```

Açıklama: os Kütüphanesi kullanılarak belirtilen alandaki dosyalar listelenir.

Section 2

```
df = pd.read_csv('../DataSet/Machines.csv')
```

Açıklama: Pandas kütüphanesi ile hedef csv dosyası okunur.

Section 3

```
df.head()
```

	Cihaz_ID	Cihaz_Uretim_Tarihi	Ariza	Hurda_Durumu
0	0	2019	Batarya	1
1	1	2021	LCD	0
2	2	2019	LCD	0
3	3	2015	LCD	0
4	4	2013	LCD	0

Açıklama: Pandas kütüphanesi ile okutulan csv dosyasını ekrana getirir.

Section 4

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Ariza'] = le.fit_transform(df.Ariza)
df.Hurda_Durumu = le.fit_transform(df.Hurda_Durumu)
```

Açıklama: csv dosyasındaki arıza sütunu yapay zekada “hurda durumu” ile eşleşebilmesi için arıza sütununun dönüştürme işlemi yapılır.

Section 5

```
df = df[['Cihaz_ID', 'Cihaz_Uretim_Tarihi', 'Ariza', 'Hurda_Durumu']]
```

Açıklama: csv dosyası verileri pandas kütüphanesi ile tekrardan düzenlenir.

Section 6

```
df.head(10)
```

	Cihaz_ID	Cihaz_Uretim_Tarihi	Ariza	Hurda_Durumu
0	0	2019	0	1
1	1	2021	1	0
2	2	2019	1	0
3	3	2015	1	0
4	4	2013	1	0
5	5	2016	1	0
6	6	2014	0	1
7	7	2019	0	1
8	8	2017	1	0
9	9	2012	0	1

Açıklama: Düzenlenen csv dosyası df.head komutu ile tekrardan listelenir.

Section 7

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df.iloc[:, :-1], df.iloc[:, -1],
                                                    test_size=0.2, random_state=1)
```

Açıklama: sklearn kütüphanesi içinde bulunan train –test –split modeli kullanılarak test ve eğitim verileri csv dosyası içinde bölümlendirilir.

Section 8

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
((4000, 3), (1001, 3), (4000,), (1001,))
```

Açıklama: Eğitim ve test verileri oluşturulur.

Section 9

```
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier

pipeline = make_pipeline(StandardScaler(), RandomForestClassifier(n_estimators=100, max_depth=4))
```

Açıklama: Pipeline kütüphanesi kullanılarak eğitim ve test verileri kesinlik skoru için senkronize edilir.

Section 10

```
pipeline.fit(X_train, y_train)

Pipeline(steps=[('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(max_depth=4))])
```

Açıklama: Section 9 açıklaması aynı şekilde geçerlidir.

Section 11

```
print('Model Accuracy: %.2f' % pipeline.score(X_test, y_test))
Model Accuracy: 1.00
```

Açıklama: Eğitim ve test verilerini pipeline skorları gösterilir.

Section 12

```
print('Model Accuracy: %.2f' % pipeline.score(X_train, y_train))
Model Accuracy: 1.00
```

Açıklama: Eğitim ve test verilerini pipeline skorları gösterilir.

Section 13

```
from sklearn.model_selection import cross_val_score  
  
scores = cross_val_score(pipeline, X=X_train, y=y_train, cv=10, n_jobs=1)
```

Açıklama: cross validation skoru için sklearn.model_selection kütüphanesinden cross_val_score modeli sisteme çekilir.

Scores= cross_val_score(pipeline,x=x_train,cv=10 n_jobs=1)

Oluşturulan scores değişkeni üzerinden cross validation skor modeli kullanılarak pipeline skoru verileri çekilir. 10 olan cross validation (cv=10) üzerinden her alan için giriş yapmak üzere (n_jobs=1)skor değeri ölçütlendirilir.

Section 14

```
print('CV accuracy scores: %s' % scores)
```

```
CV accuracy scores: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

Açıklama: 10 olan cross validation skorları yansıtılır.

Section 15

```
print('CV accuracy: %.3f +/- %.3f' % (np.mean(scores),np.std(scores)))
```

```
CV accuracy: 1.000 +/- 0.000
```

Açıklama: Np Kütüphanesi kullanılarak kesinlik oranları yansıtılır.

Section 16

```
from sklearn.tree import DecisionTreeClassifier  
from sklearn.linear_model import LogisticRegression  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.svm import SVC  
from sklearn.ensemble import RandomForestClassifier  
import matplotlib.pyplot as plt
```

Açıklama: Makine öğrenim algoritmaları ve grafik tasarımları için gerekli kütüphaneler sisteme aktarılır.

Section 17

```
# Karar Ağaçları
dt_pipeline = make_pipeline(StandardScaler(), DecisionTreeClassifier(random_state=1))
dt_pipeline.fit(X_train, y_train)
dt_accuracy = dt_pipeline.score(X_test, y_test)
print('Decision Tree Model Accuracy: %.2f' % dt_accuracy)

# Lojistik Regresyon
lr_pipeline = make_pipeline(StandardScaler(), LogisticRegression(random_state=1))
lr_pipeline.fit(X_train, y_train)
lr_accuracy = lr_pipeline.score(X_test, y_test)
print('Logistic Regression Model Accuracy: %.2f' % lr_accuracy)

# K-En Yakın Komşu (KNN)
knn_pipeline = make_pipeline(StandardScaler(), KNeighborsClassifier(n_neighbors=5))
knn_pipeline.fit(X_train, y_train)
knn_accuracy = knn_pipeline.score(X_test, y_test)
print('KNN Model Accuracy: %.2f' % knn_accuracy)

# Destek Vektör Makineleri (SVM)
svm_pipeline = make_pipeline(StandardScaler(), SVC(kernel='linear', C=1, random_state=1))
svm_pipeline.fit(X_train, y_train)
svm_accuracy = svm_pipeline.score(X_test, y_test)
print('SVM Model Accuracy: %.2f' % svm_accuracy)

# Rastgele Ormanlar
rf_pipeline = make_pipeline(StandardScaler(), RandomForestClassifier(n_estimators=100, max_depth=4, random_state=1))
rf_pipeline.fit(X_train, y_train)
rf_accuracy = rf_pipeline.score(X_test, y_test)
print('Random Forest Model Accuracy: %.2f' % rf_accuracy)

# Performansları karşılaştırmak için grafik oluşturma
models = ['Decision Tree', 'Logistic Regression', 'KNN', 'SVM', 'Random Forest']
accuracies = [dt_accuracy, lr_accuracy, knn_accuracy, svm_accuracy, rf_accuracy]

plt.bar(models, accuracies, color=['blue', 'orange', 'green', 'red', 'purple'])
plt.ylim(0.0, 1.0)
plt.title('Model Performansları')
plt.xlabel('Modeller')
plt.ylabel('Doğruluk')
plt.show()
```

Açıklama: Makine öğrenim algoritmalarının doğruluk oranları yansıtılır ve mat pilotlip kütüphanesi ile grafiği çıkartılır. Grafik şekli programı çalıştırdığınızda görseli görebilirsiniz.

Section 18

```
# Karar Ağaçları
dt_scores = cross_val_score(dt_pipeline, X_train, y_train, cv=10, n_jobs=1)
print('Decision Tree CV accuracy: %.3f +/- %.3f' % (np.mean(dt_scores), np.std(dt_scores)))

# Lojistik Regresyon
lr_scores = cross_val_score(lr_pipeline, X_train, y_train, cv=10, n_jobs=1)
print('Logistic Regression CV accuracy: %.3f +/- %.3f' % (np.mean(lr_scores), np.std(lr_scores)))

# K-En Yakın Komşu (KNN)
knn_scores = cross_val_score(knn_pipeline, X_train, y_train, cv=10, n_jobs=1)
print('KNN CV accuracy: %.3f +/- %.3f' % (np.mean(knn_scores), np.std(knn_scores)))

# Destek Vektör Makineleri (SVM)
svm_scores = cross_val_score(svm_pipeline, X_train, y_train, cv=10, n_jobs=1)
print('SVM CV accuracy: %.3f +/- %.3f' % (np.mean(svm_scores), np.std(svm_scores)))

# Rastgele Ormanlar
rf_scores = cross_val_score(rf_pipeline, X_train, y_train, cv=10, n_jobs=1)
print('Random Forest CV accuracy: %.3f +/- %.3f' % (np.mean(rf_scores), np.std(rf_scores)))

# Performansları karşılaştırmak için çapraz doğrulama sonuçlarını görselleştirme
cv_models = ['Decision Tree', 'Logistic Regression', 'KNN', 'SVM', 'Random Forest']
cv_accuracies = [dt_scores, lr_scores, knn_scores, svm_scores, rf_scores]

plt.figure(figsize=(10, 6))
plt.boxplot(cv_accuracies, labels=cv_models)
plt.title('Çapraz Doğrulama Sonuçları')
plt.xlabel('Modeller')
plt.ylabel('Doğruluk')
plt.show()
```

Açıklama: Makine öğrenim algoritmalarını doğrulama sonuçları yansıtılır. Doğrulama tablosunu programı çalıştırdığınızda inceleyebilirsiniz.

Sonuç (Conclusion)

Geliştirme: Yukarıda yaptığım çalışmanın ileri ki zaman da daha detaylı ve görsel haline getirmeyi planlıyorum.

Yapılacak Eklmeler: Sorguları arttırmak, yapılan çalışmayı bir ara yüze bağlamak.

Bu çalışma, Makine Öğrenmesi konusunda bir analizi sunuldu. Bu raporun hazırlanmasında bize destek olan Aytuğ ONAN hocam ile birlikte çalışma fırsatı bulduğum için teşekkür ederim.

3.DÖNEM BİTİRME PROJESİ PART II

2.CSV Data Analyzer GUI Modu Bitirme Projesi

Projenin ilk versiyonunda yapılan analiz bir csv dosyası üzerinden yürütülmekteydi. Final versiyonunda ise kolay data erişimi ve anlaşılır data bilgisi için veri kontrolü Database (sqlite) olarak uygulanmıştır. Veriler hurda ise kırmızı, hurda değil ise yeşil renk ile belirtilmektedir (Şekil.2.1)

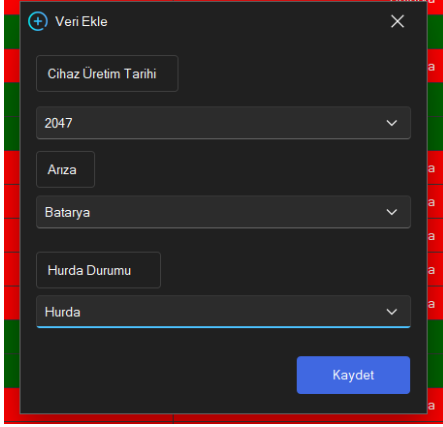
CHAZ LİRETIM TARİHİ	ARIZA	HURDA DURUMU
2012	LCD	0
2013	Batarya	1
2014	LCD	0
2013	Batarya	1
2020	LCD	0
2006	LCD	0
2018	Batarya	1
2009	Batarya	1
2018	Batarya	1
2009	Batarya	1
2018	Batarya	1
2007	LCD	0
2017	LCD	0
2010	Batarya	1
2012	Batarya	1
2013	Batarya	1
2007	Batarya	1
2017	Batarya	1
2019	LCD	0
2022	LCD	0
2008	Batarya	1

Şekil.2.1: Hurda durum renk göstergesi

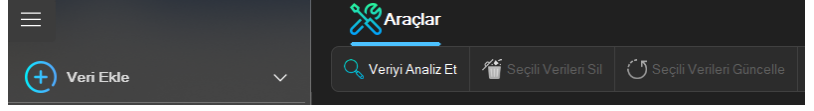
Uygulama içerisinde veri ekleme, veri silme (opsiyonel: çoklu veri silme) ve çoklu veri güncelleme metotları bulunmaktadır.

Veri Ekleme:

Yan tarafta bulunan hamburger menü açıldığında veri ekle butonu karşınıza çıkar. Bu buton ile etkileşimde bulunulduğunda verileri ekleyeceğiniz form karşınıza çıkar(Şekil.2.2.1, Şekil.2.2.2).



Şekil.2.2.1: Menü açılımı



Şekil.2.2.2: Veri ekleme butonu

Veri Silme:

Ana ekranda seçili bir veri olur ise yukarıda bulunan ribbon toolbar da seçili verileri sil butonu aktifleşir ve bu butona basıldığında seçtiğiniz veriler silinir(Şekil.2.3).

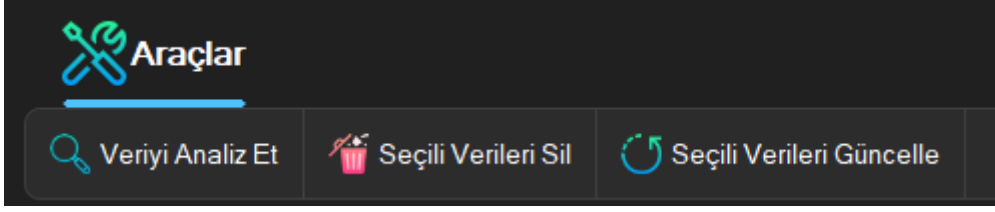


	CIHAZ ÜRETİM TARİHİ	ARIZA	HURDA DURUMU
<input checked="" type="checkbox"/>	2012	LCD	0
<input type="checkbox"/>	2013	Batarya	1
<input type="checkbox"/>	2014	LCD	0
<input type="checkbox"/>	2013	Batarya	1
<input checked="" type="checkbox"/>	2020	LCD	0
<input type="checkbox"/>	2006	LCD	0
<input type="checkbox"/>	2018	Batarya	1
<input type="checkbox"/>	2009	Batarya	1
<input checked="" type="checkbox"/>	2018	Batarya	1

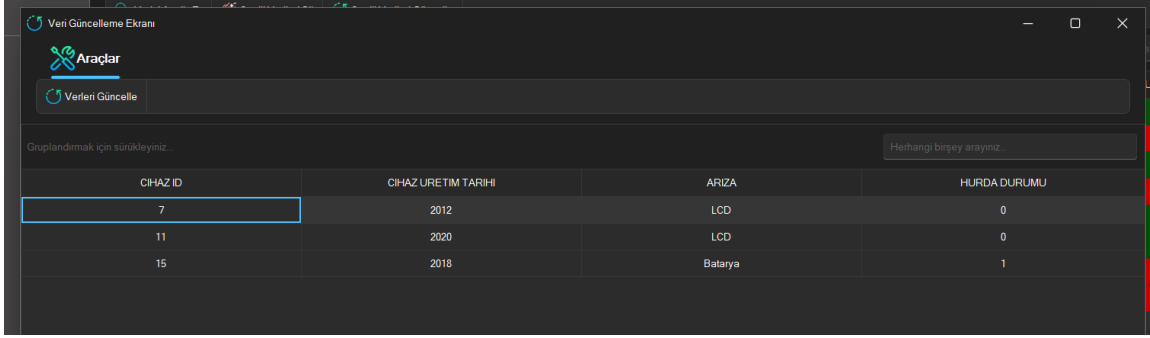
Şekil.2.3: Veri silme

Çoklu Güncelleme:

Ana ekranda bulunan veriler seçildiğinde yukarıda bulunan “Seçili Verileri Güncelle” butonu aktifleşir. Bu buton ile etkileşim bulunulduğunda seçtiğiniz verileri ayrı bir güncelleme işlemi yapacağınız forma yönlendirmektedir. (Şekil.2.4.1, Şekil.2.4.2, Şekil.2.4.3)

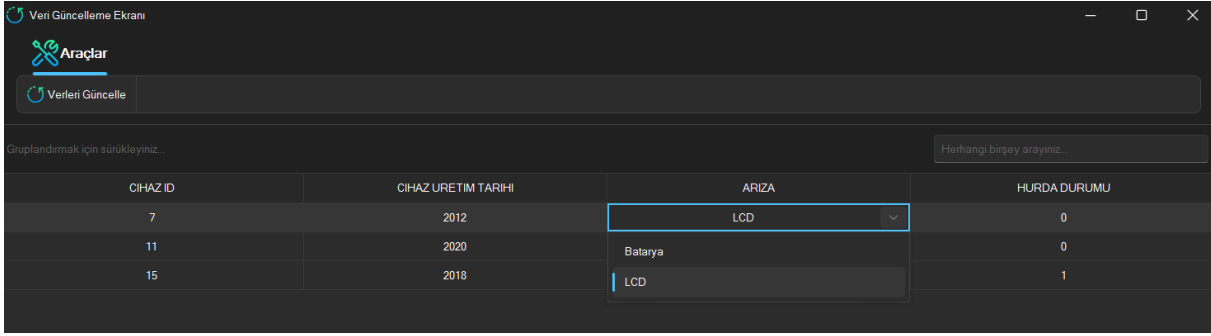


Şekil.2.4.1: Çoklu Güncelleme ekranı



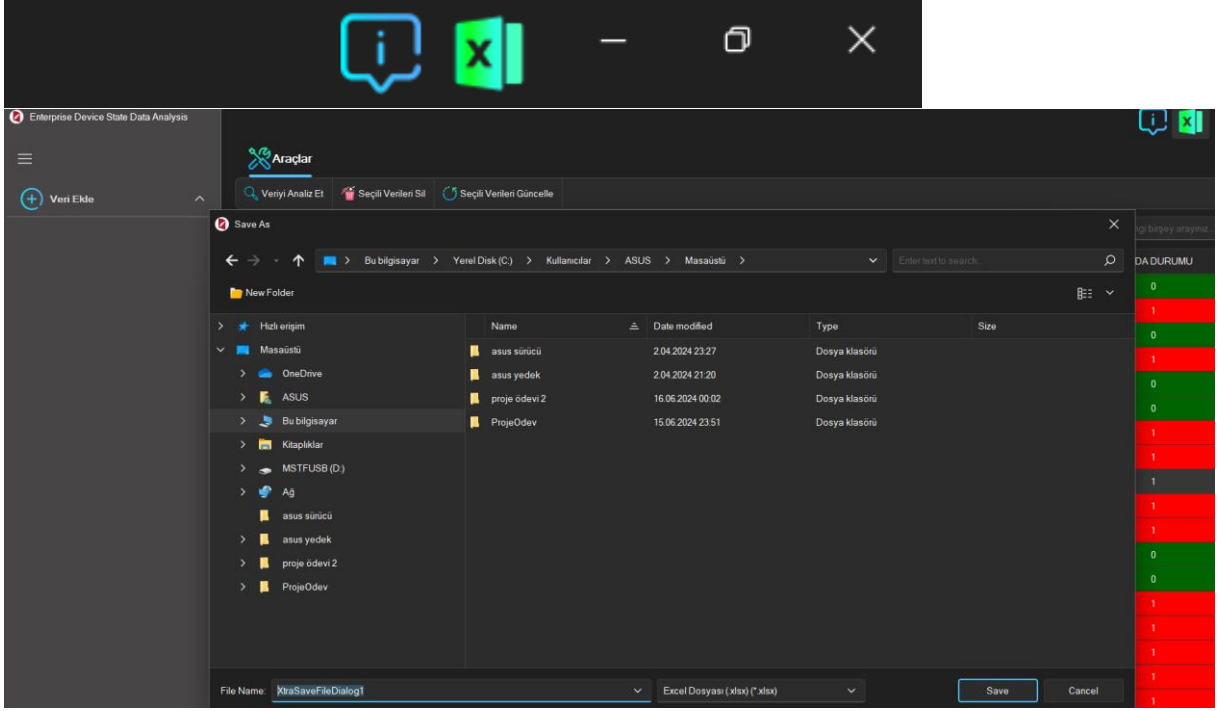
Şekil.2.4.2: Çoklu Güncelleme ekranı

Seçilen veriler karışımıza gelen güncelleme formunda grid içerisinde kolay erişim ile güncellenebilmektedir.



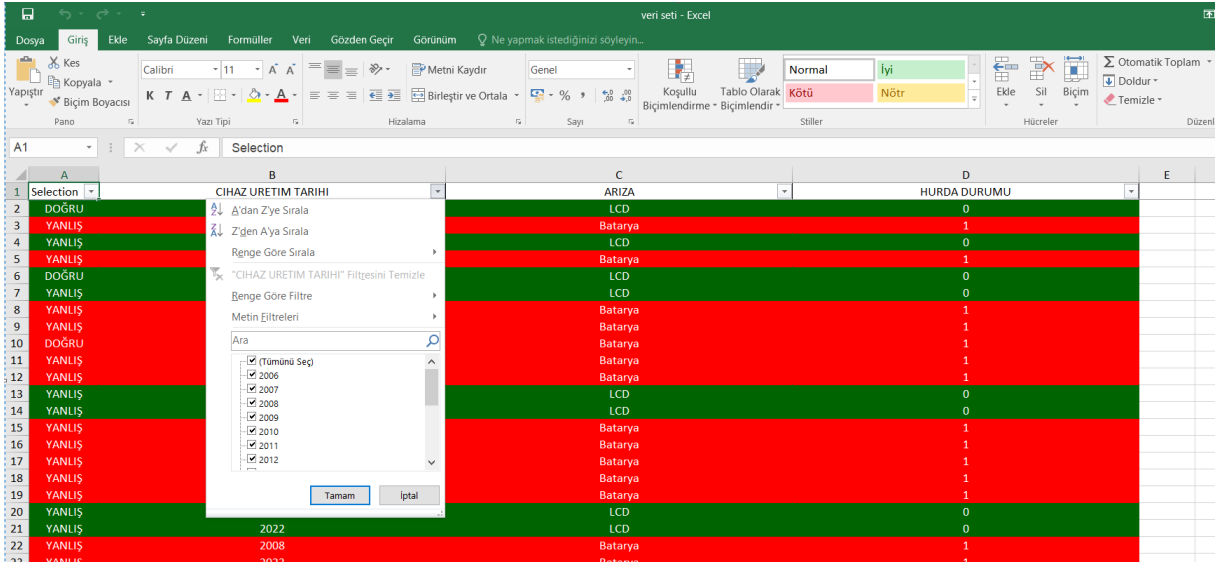
Şekil.2.4.3: Çoklu Güncelleme ekranı

Excel Formatında Dışarı Aktarıma: Ana ekranda sağ üst köşede bulunan excel resmine tıkladığında verilerinizi excel formatında nereye kaydedeceğinizi seçmenizi isteyen diyalog penceresi açılacaktır. Konum seçimini yaptığınızda veriler excel formatında kaydedilecektir(Şekil.2.5.1, Şekil.2.5.2).



Şekil.2.5.1: Excel Aktar Ekranı

Veriler excel formatında dışarıya aktarıldığında uygulama içerisinde bulunan filtreler ve renklendirme kuralları excel dosyasında da mevcut bulunmaktadır.



Şekil.2.5.1: Excel Aktarılmış hali

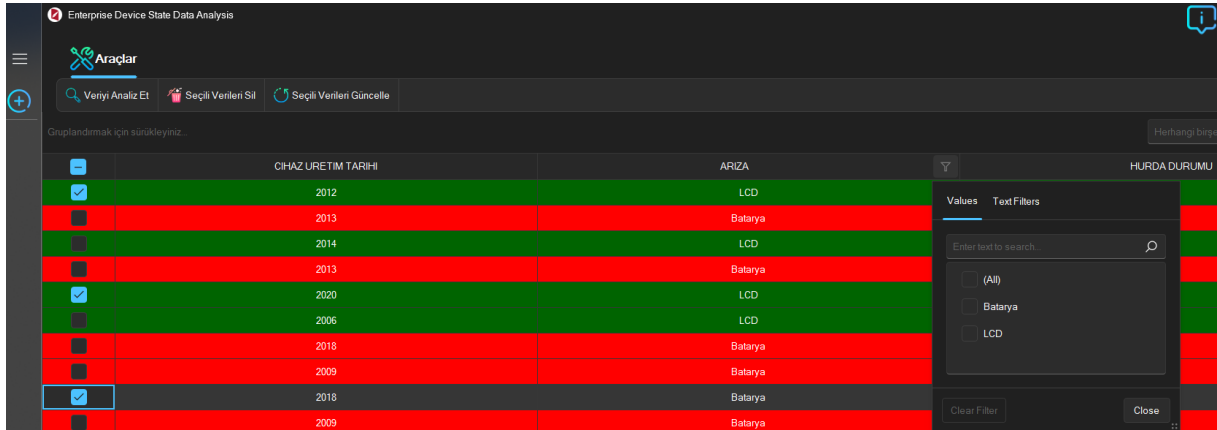
Uygulama İçi Filtreleme Sistemi: Uygulama içerisinde üç adet filtreleme sistemi mevcuttur. Bunlar:

Excel tipi filtreleme

Mixed arama

Jira tipi filtreleme

Excel Tipi Filtreleme: Excel uygulamasında bulunan header filtreleme modu uygulama içerisinde bulunan grid’de mevcuttur(Şekil.2.6.1).



Şekil.2.6.1: Excel tipi filtreleme

Mixed Arama: uygulamanın ana ekranında sağ üstü köşede bulunan arama çubuğu girilen tekli veya çoklu verileri ayrıştıracak arama motordur(Şekil.2.7.1).

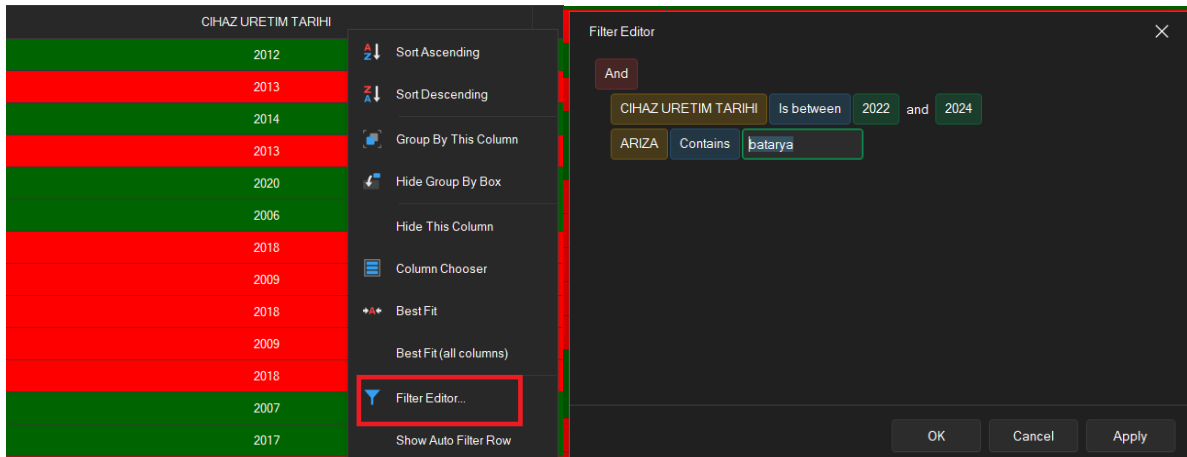


Şekil.2.7.1: Mixed Arama

Jira Tipi Filtreleme: Birden fazla sorguyu tek ekranda kolaylıkla yapabileceğiniz filtreleme sistemidir.

Örnek filtreleme sorgusu aşağıda verilmiştir. Sorgunun sonucu resim olarak aşağıda mevcuttur.

Sorgu: Sağ tarafta bulunan resimde 2022 ile 2024 tarihleri arasında arıza durumu batarya olan verileri getirme sorgusu gösterilmiştir(Şekil.2.8.1, Şekil.2.8.2).



Şekil.2.8.1: Jira Tipi Sorulama

Sonuç:

	HAZURETİM TARİHİ	ARIZA	HURDADURUMU
	2023	Batarya	1
	2023	Batarya	1
	2023	Batarya	1
	2023	Batarya	1
	2022	Batarya	1

Şekil.2.8.2: Jira Tipi Sorulama Sonuç

Veri Analizi

Uygulama içerisinde data tarafı visual basic dili ile kontrol edilir. Verilerin doğruluk ve netlik oranını kontrolünü sağlayacak ve bu oranların grafiğini kullanıcıya sunacak analiz işlemi Python dilinden yönlendirilir. Analiz işlemi uygulama içerisinde uygulama dosyalarının konumunda bulunan PyFunctions dizini içerisinde bulunan python dosyaları ile yapılır(Şekil.2.9.1).



Şekil.2.9.1: Veriyi Analiz Et

Veriyi analiz et butonuna basıldığında ana ekranda sırasıyla model performansları tablosu, çapraz doğrulama sonuçlarını bulunduran grafik formu ve netlik oranlarının çıktısını veren formlar kullanıcıyı karşılamaktadır. Belirtilen formların hangi dil ile yürütüldüğü aşağıda bulunan maddelerde verilmiştir.

Model Performansları Formu: Python

Çapraz Doğrulama Formu: Python

Sonuçlar Formu: Visual Basic

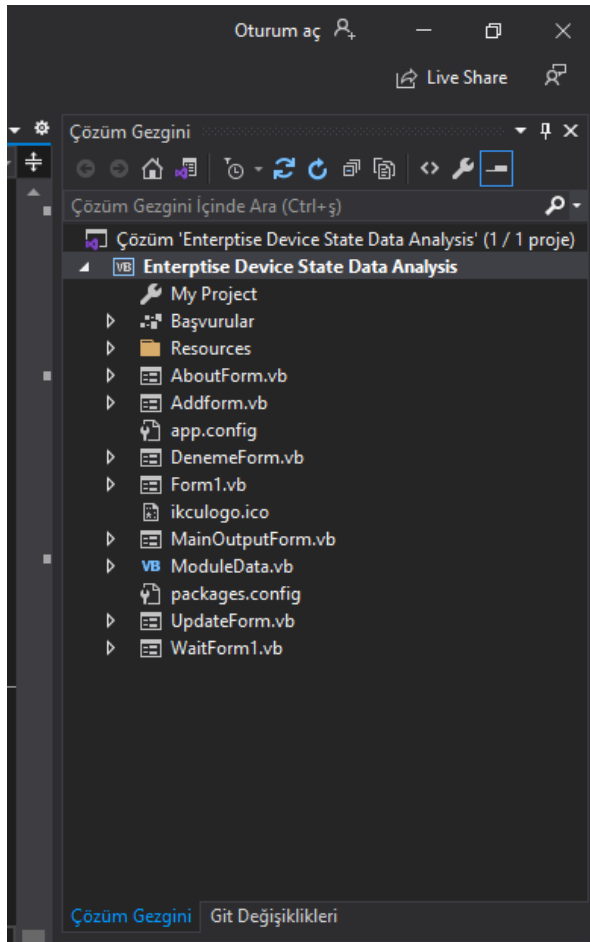
3.Modüller ve Class Kodları

Bu bölümde uygulama içerisinde modüllerde ve classlarda yazılan kodların amacını ve hedefini bölümlendirilerek anlatacağım. Aşağıda verilen maddelerde sınıflandırmalar listelenmiştir.

- Module Data : Uygulamanın çalışma fonksiyonları, data fonksiyonları ve python komutlarını çalıştıracak Windows Batch file komutları mevcuttur.
- Ana dizinde (Form1) yapısı, sub'lar ve class verileri mevcuttur.
- Module ve Sub'larda SQL Transaction kodları anlatılacaktır.

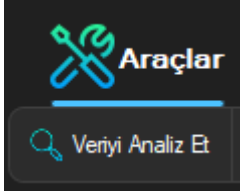
Modüller: ModuleData.vb

Aşağıda verilen resim içerisinde bulunan “ModuleData.vb” modülü uygulamanın SQL, Procces fonksiyonlarını çalıştıran sınıftır. Kodlar ara yüz ile kod bağlantısı yapılarak aşağıda anlatılmıştır(Şekil.3.1).



Şekil.3.1: Modüller

Verileri Analiz Et Butonu: Uygulamanın ana sayfasında bulunan buton uygulama dizinin içerisinde bulunan F1, F2, F3, F4 ve F5 adlı python dosyalarını Batch file üzerinden çalıştırmaktadır(Şekil.3.2).



Şekil.3.2: Verileri Analiz Et Butonu

VB.Net F4 Python Fonksiyonu:

F4 Fonksiyonu vb.net dilinde uygulamanın dizini içerisinde bulunan F4.py dosyasını çalıştıracak bat dosyasını çalıştırır. Bu çalıştırma işlemi yapmadan önce çapraz doğrulama işleminin netlik ve doğrulama printlerini CMD ekranından alıp form uygulamasında görüntülemek için StreamReader fonksiyonu üzerinden output çıktısını uygulama ile birleştirmektedir. Bu işlem sonrasında “Çapraz Doğrulama” grafiğini yansıtmak için F4.py dosyasını çalıştırmaktadır(Şekil.3.3).

```
1 başvuru
Public Sub F4_Py_Function()
    Dim proc As New Process

    proc.StartInfo.FileName = "C:\Enterprise Device State Data Analysis\PyFunctions\F4_Print.bat"
    proc.StartInfo.UseShellExecute = False
    proc.StartInfo.RedirectStandardOutput = True
    proc.StartInfo.CreateNoWindow = True
    proc.Start()
    proc.WaitForExit()

    Dim sOutput As String
    Using oStreamReader As System.IO.StreamReader = proc.StandardOutput
        sOutput = oStreamReader.ReadToEnd()
    End Using

    MainOutputForm.RichTextBox_KNN_F.Text = sOutput

    Dim RLine As List(Of String) = MainOutputForm.RichTextBox_KNN_F.Lines.ToList()
    RLine.RemoveAt(0)
    MainOutputForm.RichTextBox_KNN_F.Lines = RLine.ToArray()
    MainOutputForm.RichTextBox_KNN_F.Refresh()
    RLine.RemoveAt(0)
    MainOutputForm.RichTextBox_KNN_F.Lines = RLine.ToArray()
    MainOutputForm.RichTextBox_KNN_F.Refresh()

    Dim proc2 As New Process

    proc2.StartInfo.FileName = "C:\Enterprise Device State Data Analysis\PyFunctions\F4.bat"
    proc2.StartInfo.UseShellExecute = False
    proc2.StartInfo.RedirectStandardOutput = True
    proc2.StartInfo.CreateNoWindow = True
    proc2.Start()
End Sub
```

Şekil.3.3: VB.Net F4 Python Fonksiyonu

F5 Python Fonksiyonu:

F5 Fonksiyonu vb.net dilinde uygulamanın dizini içerisinde bulunan F5.py dosyasını çalıştıracak bat dosyasını çalıştırır. Bu çalıştırma işlemini yapmadan önce model performansı işleminin netlik ve doğrulama printlerini CMD ekranından alıp form uygulamasında görüntülemek için StreamReader fonksiyonu üzerinden output çıktısını uygulama ile birleştirmektedir. Bu işlem sonrasında “Model Performansı” grafiğini yansıtmak için F5.py dosyasını çalıştırmaktadır(Şekil.3.4).

```
Public Sub F5_Py_Function()  
    Dim proc As New Process  
  
    proc.StartInfo.FileName = "C:\Enterprise Device State Data Analysis\PyFunctions\F5_Print.bat"  
    proc.StartInfo.UseShellExecute = False  
    proc.StartInfo.RedirectStandardOutput = True  
    proc.StartInfo.CreateNoWindow = True  
    proc.Start()  
    proc.WaitForExit()  
  
    Dim sOutput As String  
    Using oStreamReader As System.IO.StreamReader = proc.StandardOutput  
        sOutput = oStreamReader.ReadToEnd()  
    End Using  
  
    MainOutputForm.RichTextBox_Capraz_F.Text = sOutput  
  
    Dim RLine As List(Of String) = MainOutputForm.RichTextBox_Capraz_F.Lines.ToList()  
    RLine.RemoveAt(0)  
    MainOutputForm.RichTextBox_Capraz_F.Lines = RLine.ToArray()  
    MainOutputForm.RichTextBox_Capraz_F.Refresh()  
    RLine.RemoveAt(0)  
    MainOutputForm.RichTextBox_Capraz_F.Lines = RLine.ToArray()  
    MainOutputForm.RichTextBox_Capraz_F.Refresh()  
  
    Dim proc2 As New Process  
  
    proc2.StartInfo.FileName = "C:\Enterprise Device State Data Analysis\PyFunctions\F5.bat"  
    proc2.StartInfo.UseShellExecute = False  
    proc2.StartInfo.RedirectStandardOutput = True  
    proc2.StartInfo.CreateNoWindow = True  
    proc2.Start()  
End Sub
```

Şekil.3.4: F5 Python Fonksiyonu

F3 Python Fonksiyonu :

F3 fonksiyonu eğitim ve test sonuçlarının oranını gösterecek olan uygulama dizini içerisinde bulunan F3.py dosyasını Batch dosyası aracılığıyla çalıştırmaktadır. Çalıştırılan F3.py dosyasının cmd ekranına yansıyan printlerini yakalayıp uygulamanın “Sonuçlar” formuna taşımaktadır(Şekil.3.5).

```
1 başvuru
Public Sub F3_Py_Function()
    Dim proc As New Process
    proc.StartInfo.FileName = "C:\Enterprise Device State Data Analysis\PyFunctions\F3.bat"
    proc.StartInfo.UseShellExecute = False
    proc.StartInfo.RedirectStandardOutput = True
    proc.StartInfo.CreateNoWindow = True
    proc.Start()
    proc.WaitForExit()

    Dim sOutput As String
    Using oStreamReader As System.IO.StreamReader = proc.StandardOutput
        sOutput = oStreamReader.ReadToEnd()
    End Using

    MainForm.RichTextBox_TestOrani.Text = sOutput

    Dim RLine As List(Of String) = MainForm.RichTextBox_TestOrani.Lines.ToList()
    RLine.RemoveAt(0)
    MainForm.RichTextBox_TestOrani.Lines = RLine.ToArray()
    MainForm.RichTextBox_TestOrani.Refresh()
    RLine.RemoveAt(0)
    MainForm.RichTextBox_TestOrani.Lines = RLine.ToArray()
    MainForm.RichTextBox_TestOrani.Refresh()
End Sub
```

Şekil.3.5: F3 Python Fonksiyonu

4.Database İşlemleri

ModuleData.vb içerisinde bulunan database komutları aşağıda verilen maddelerde ayrılmıştır.

- Load Data Kodları : SQLite veritabanında bulunan kayıtları uygulama içerisinde bulunan gridview aracına aktaracak komutlardır.
- Remove Data Kodları : Uygulamada gridview aracında bulunan seçili kayıtları veritabanı üzerinden silinmesini sağlayacak kodlardır.
- Update Data Kodları : Uygulamada gridview aracında bulunan seçili kayıtların veritabanında güncellenmesini sağlayan kodlardır.

Load Data :

Aşağıda gösterilen kod uygulamanın ana sayfasında bulunan gridview aracına cihaz listesini listeleyen kodlardır(Şekil.4.1.1, Şekil.4.1.2).

```
Public Sub Self_Login_Load_Data()  
    Dim sqlite_cmd As SQLiteCommand  
    sqlite_cmd = SQLiteConnectionString.CreateCommand()  
    sqlite_cmd.CommandText = "SELECT * FROM Machines;"  
  
    Dim sqlite_dataadapter As New SQLite.SQLiteDataAdapter  
    sqlite_dataadapter.SelectCommand = sqlite_cmd  
    Dim dt As New DataTable  
  
    sqlite_dataadapter.Fill(dt)  
    SQLiteConnectionString.Open()  
  
    Form1.GridControl1.DataSource = dt  
    SQLiteConnectionString.Close()  
End Sub
```

Şekil.4.1.1: Load Data

Aşağıda verilen kod parçası var olan ana listedeki bulunan seçili cihazlar update etmek için ayrı bir veri tabanı tablosuna ayrıştırdıktan sonra update tablosunu verileri getirme amacı bulunan kod parçasıdır.

```
2 başvuru  
Public Sub Self_Update_List_Load()  
    Dim sqlite_cmd As SQLiteCommand  
    sqlite_cmd = SQLiteConnectionString.CreateCommand()  
    sqlite_cmd.CommandText = "SELECT * FROM Update_List;"  
  
    Dim sqlite_dataadapter As New SQLite.SQLiteDataAdapter  
    sqlite_dataadapter.SelectCommand = sqlite_cmd  
    Dim dt As New DataTable  
  
    sqlite_dataadapter.Fill(dt)  
    SQLiteConnectionString.Open()  
  
    UpdateForm.GridControl1.DataSource = dt  
    SQLiteConnectionString.Close()  
End Sub
```

Şekil.4.1.2: Load Data

Remove Data :

Uygulamanın ana formunda bulunan gridview aracında seçili kayıtların veri tabanı üzerinden silinmesini sağlayan kod parçasıdır.

NOT: En alt kısımda bulunan “Self_Login_Load_Data” kodu belirtilen isime sahip olan sub fonksiyona gidip silme işlemi sonrası yeni data verilerini ana ekrana listeleyen koddur(Şekil.4.2).

```
1 başyuru
Public Sub Remove_Selected_Data()
    Form1.SplashScreenManager1.ShowWaitForm()
    Form1.SplashScreenManager1.SetWaitFormCaption("Lütfen Bekleyiniz.")
    Form1.SplashScreenManager1.SetWaitFormDescription("Seçili veriler siliniyor...")

    Dim sqlite_cmd As SQLiteCommand

    Dim selectedRowHandles As Int32() = Form1.GridView1.GetSelectedRows()
    Dim I As Integer
    For I = 0 To selectedRowHandles.Length - 1
        Dim selectedRowHandle As Int32 = selectedRowHandles(I)
        If (selectedRowHandle >= 0) Then
            Dim Cihaz_ID = CType(Form1.GridView1.GetRowCellValue(selectedRowHandles(I), "Cihaz_ID").ToString(), String)

            SQLiteConnectionString.Open()
            sqlite_cmd = SQLiteConnectionString.CreateCommand()
            sqlite_cmd.CommandText = "DELETE FROM Machines WHERE Cihaz_ID = " & "" & Cihaz_ID & "" & ";"
            sqlite_cmd.ExecuteNonQuery()
            SQLiteConnectionString.Close()
        End If
    Next

    Self_Login_Load_Data()
    Form1.SplashScreenManager1.CloseWaitForm()
End Sub
```

Şekil.4.2: Remove Data

Register Data :

Aşağıda verilen kod parçası uygulamanın ana ekranında hamburger menü içerisinde bulunan “Veri Ekle” butona basıldığında veri tabanına ekleme yapan kod parçasıdır(Şekil.4.3).

```
Public Sub Add_Register()
    Form1.SplashScreenManager1.ShowWaitForm()
    Form1.SplashScreenManager1.SetWaitFormCaption("Lütfen Bekleyiniz.")
    Form1.SplashScreenManager1.SetWaitFormDescription("Veri ekleniyor...")

    Dim Hurda_Durumu_String = ""

    If (Addform.ComboBoxEdit2.Text.ToString() = "Hurda") Then
        Hurda_Durumu_String = "1"
    Else
        Hurda_Durumu_String = "0"
    End If

    Dim sqlite_cmd As SQLiteCommand

    SQLiteConnectionString.Open()
    sqlite_cmd = SQLiteConnectionString.CreateCommand()
    sqlite_cmd.CommandText = "INSERT INTO Machines (Cihaz_Uretim_Tarihi, Ariza, Hurda_Durumu) VALUES (" & "" & Addform.ComboBoxEdit3.Text.ToString() & "" & ", " & "" & Addform.ComboBoxEdit1.Text.
    sqlite_cmd.ExecuteNonQuery()
    SQLiteConnectionString.Close()

    Self_Login_Load_Data()
    Addform.Dispose()
    Addform.Hide()
    Form1.SplashScreenManager1.CloseWaitForm()
End Sub
```

Şekil.4.3: Register Data

Update Data :

Aşağıda verilen kod parçası update işlemi için ana sayfa üzerinde bulunan seçili kayıtların “Update_List” veri tabanı tablosuna aktarılmasını sağlayan kod parçasıdır. Register (Insert Into) transaction kapsamaktadır(Şekil.4.4).

```

Public Sub Transfer_Update_List()
    Dim sqlite_cmd As SQLiteCommand
    Form1.SplashScreenManager1.ShowWaitForm()
    Form1.SplashScreenManager1.SetWaitFormCaption("Lütfen bekleyiniz.")
    Form1.SplashScreenManager1.SetWaitFormDescription("Veriler hazırlanıyor...")

    SQLiteConnectionString.Open()
    sqlite_cmd = SQLiteConnectionString.CreateCommand()
    sqlite_cmd.CommandText = "DELETE FROM Update_List;"
    sqlite_cmd.ExecuteNonQuery()
    SQLiteConnectionString.Close()

    Dim selectedRowHandles As Int32() = Form1.GridView1.GetSelectedRows()
    Dim I As Integer
    For I = 0 To selectedRowHandles.Length - 1
        Dim selectedRowHandle As Int32 = selectedRowHandles(I)
        If (selectedRowHandle >= 0) Then
            Dim Cihaz_ID = CType(Form1.GridView1.GetCellValue(selectedRowHandles(I), "Cihaz_ID").ToString(), String)
            Dim Cihaz_Uretim_Tarihi = CType(Form1.GridView1.GetCellValue(selectedRowHandles(I), "Cihaz_Uretim_Tarihi").ToString(), String)
            Dim Ariza = CType(Form1.GridView1.GetCellValue(selectedRowHandles(I), "Ariza").ToString(), String)
            Dim Hurda_Durumu = CType(Form1.GridView1.GetCellValue(selectedRowHandles(I), "Hurda_Durumu").ToString(), String)

            SQLiteConnectionString.Open()
            sqlite_cmd = SQLiteConnectionString.CreateCommand()
            sqlite_cmd.CommandText = "INSERT INTO Update_List (Cihaz_ID, Cihaz_Uretim_Tarihi, Ariza, Hurda_Durumu) VALUES (" & "" & Cihaz_ID & "" & ", " & "" & Cihaz_Uretim_Tarihi & "" & ", " & "" & Ariza & "" & ", " & "" & Hurda_Durumu & "" & ");"
            sqlite_cmd.ExecuteNonQuery()
            SQLiteConnectionString.Close()
        End If
    Next

    Self.Update_List_Load()
    Form1.SplashScreenManager1.CloseWaitForm()
    UpdateForm.ShowDialog()
End Sub

```

Şekil.4.4.1: Update Data

Aşağıda verilen kod parçası update formunda bulunan verilen yeni verilerini ana veri tabanı tablosunda (Machines) güncelleyen ve yeni verileri update tablosunda göstermek için kendi içerisinde de güncelleyen kod parçasıdır(Şekil.4.4.2).

```

Dim selectedRowHandles As Int32() = UpdateForm.GridView1.GetSelectedRows()
Dim I As Integer
For I = 0 To selectedRowHandles.Length - 1
    Dim selectedRowHandle As Int32 = selectedRowHandles(I)
    If (selectedRowHandle >= 0) Then
        Dim id = CType(UpdateForm.GridView1.GetCellValue(selectedRowHandles(I), "id").ToString(), String)
        Dim Cihaz_ID = CType(UpdateForm.GridView1.GetCellValue(selectedRowHandles(I), "Cihaz_ID").ToString(), String)
        Dim Cihaz_Uretim_Tarihi = CType(UpdateForm.GridView1.GetCellValue(selectedRowHandles(I), "Cihaz_Uretim_Tarihi").ToString(), String)
        Dim Ariza = CType(UpdateForm.GridView1.GetCellValue(selectedRowHandles(I), "Ariza").ToString(), String)
        Dim Hurda_Durumu = CType(UpdateForm.GridView1.GetCellValue(selectedRowHandles(I), "Hurda_Durumu").ToString(), String)

        SQLiteConnectionString.Open()
        sqlite_cmd = SQLiteConnectionString.CreateCommand()
        sqlite_cmd.CommandText = "UPDATE Machines SET Cihaz_Uretim_Tarihi = @Cihaz_Uretim_Tarihi, Ariza = @Ariza, Hurda_Durumu = @Hurda_Durumu WHERE Cihaz_ID = " & "" & Cihaz_ID & "" & ";";
        sqlite_cmd.Parameters.AddWithValue("@Cihaz_Uretim_Tarihi", Cihaz_Uretim_Tarihi)
        sqlite_cmd.Parameters.AddWithValue("@Ariza", Ariza)
        sqlite_cmd.Parameters.AddWithValue("@Hurda_Durumu", Hurda_Durumu)
        sqlite_cmd.ExecuteNonQuery()
        SQLiteConnectionString.Close()

        SQLiteConnectionString.Open()
        sqlite_cmd = SQLiteConnectionString.CreateCommand()
        sqlite_cmd.CommandText = "UPDATE Update_List SET Cihaz_Uretim_Tarihi = @Cihaz_Uretim_Tarihi, Ariza = @Ariza, Hurda_Durumu = @Hurda_Durumu WHERE id = " & "" & id & "" & ";";
        sqlite_cmd.Parameters.AddWithValue("@Cihaz_Uretim_Tarihi", Cihaz_Uretim_Tarihi)
        sqlite_cmd.Parameters.AddWithValue("@Ariza", Ariza)
        sqlite_cmd.Parameters.AddWithValue("@Hurda_Durumu", Hurda_Durumu)
        sqlite_cmd.ExecuteNonQuery()
        SQLiteConnectionString.Close()
    End If
Next

```

Şekil.4.4.2: Update Data

Kaynaklar

- <https://www.kaggle.com/code/alexisbcook/cross-validation>
- <https://www.kaggle.com/code/prashant111/decision-tree-classifier-tutorial/output>
- <https://www.sqlite.org/c3ref/constlist.html>
- <https://www.sqlite.org/lang.html>
- <https://www.sqlite.org/pragma.html#toc>
- <https://stackoverflow.com/questions/181719/how-do-i-start-a-process-from-c>
- <https://stackoverflow.com/questions/2178872/run-batch-file-in-vb-net>

*Belirtilen kaynaklar haricinde yazdığım uygulama tamamen özgüdür.