

T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
İKTİSADİ VE İDARİ BİLİMLER FAKÜLTESİ
YÖNETİM BİLİŞİM SİSTEMLERİ BÖLÜMÜ

YBS 2015 – Programlama II
Final Ödevi

Rasyo Analizi KDS

2017469074
Melih Can BAL

2020, İZMİR

1. Planlama

1.1 Projenin Amacı ve Gerekçesi

İşletmelerin hedeflerini gerçekleştirmesi ve sürekliliğini sağlaması adına performans göstergelerini takip etmesini sağlamak projenin genel amacıdır.

Bunun yanında sadece işletmeye verileri aktarmakla kalmayıp Dünya üzerinde diğer işletmelere göre olan durumunu da izlemesini sağlayacak araçlar, işletmenin daha hızlı karar vermesinde rol oynayacaktır.

İşletmelerin karar verirken daha hızlı olmasına, hatasının nerede olduğunu daha çabuk görmesine olanak sağlayacaktır.

Günümüzde hızın çok önemli bir faktör olduğu düşünülürse projenin gerekçesinin gayet açık olduğu düşünülüyor.

1.2 Projenin Kapsamı

İşletmenin güncel 1 yıllık bilançosu üzerinden alınan verilerle işleyecek olan yazılımın üzerinde duracağı analizler;

- Likidite Rasyoları (Mevcut)
- Ekonomik Yapı Rasyoları (Veri Çekme Aşamasında)
- Finansal Yapı Rasyoları (Veri Çekme Aşamasında)
- Karlılık Rasyoları (Veri Çekme Aşamasında)

2. Analiz

2.1 İlgili Rasyoların Hesaplanma Formülleri

Likidite Rasyoları

- Cari Rasyo= Dönen Varlıklar / KVK

2.2 Veri İhtiyaçları Analizi

Tüm bunları hesaplarken kullanıcıdan girilmesi istenen veriler sırasıyla;

- İşletmenin Adı
- Dönen Varlıklar Değeri
- Duran Varlıklar Değeri
- KVYK Değeri
- UVYK Değeri
- Özkaynak Değeri
- Brüt kar (İlgili Dönem)

2.3 Proje Standartları Yöntem ve Metodolojiler

- Masaüstü Uygulama Arayüzü (PyQt5)
- Numpy ve matplotlib ile Grafik Dökümü
- Beautifulsoup ile rasyo değerlerinin çekilmesi
- Pycharm Community Edition geliştirme ortamıdır.
- Gelişigüzel SDLC Geliştirme Modeli

2.4 İşlevsellik

-İşletme ilgili rasyosunu seçeceği işletme ile kıyaslayabilecek ve grafik arayüzünde veriyi analiz etme şansı bulacak.

2.5 Kullanılabilirlik

- Kullanılabilirliğini ölçmek için kullanıcılardan geri bildirim alınacaktır.

3.Tasarım

3.1 Arayüz Tasarımı

Ana Ekran

Form

Bilgiler

İşletme Adı

Dönen Varlık Değeri

Duran Varlık Değeri

KVYK Varlık Değeri

UVYK Varlık Değeri

Özkaynaklar

Karşılaştırmak İstediğiniz Firma

Apple

Karşılaştır

Firma Karşılaştırma Seçenekleri

Karşılaştırmak İstediğiniz Firma

Apple

Apple

Tesla

Microsoft

Karşılaştır

Örnek hesaplama sonucunda ortaya çıkan bilgilendirme kutusu.

Form

Bilgiler

İşletme Adı
banvit a.ş

Dönen Varlık Değeri
1446

Duran Varlık Değeri
2307

KVYK Varlık Değeri
777

UVYK Varlık Değeri
1218

Özkaynaklar
1089

Karşılaştırmak İstedığınız Firma
Apple

Karşılaştır

Hesap Sonucu

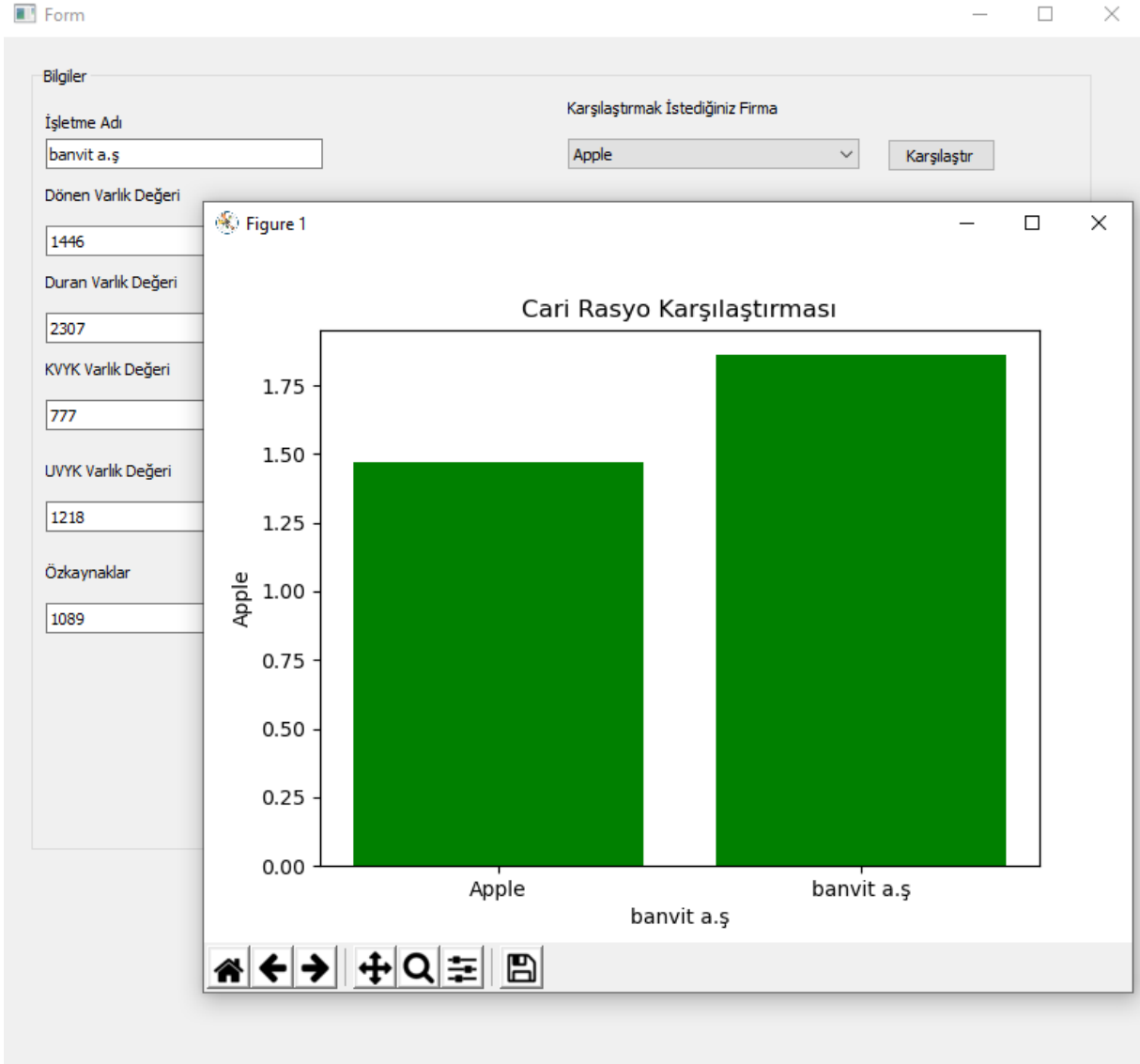
Rasyo Değerleri

OK

Hide Details...

Cari Rasyo: 1.86

“OK” tuşuna bastıktan sonra ortaya çıkan grafik



KOD

Import Kısmı

```
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QMessageBox
import requests
from bs4 import BeautifulSoup
import random
import numpy as np
import matplotlib.pyplot as plt
```

UYGULAMA

```
class Ui_Form(object):
    def setupUi(self, Form):
        Form.setObjectName("Form")
        Form.resize(788, 718)
        self.
        self.groupBox = QtWidgets.QGroupBox(Form)
        self.groupBox.setGeometry(QtCore.QRect(20, 20, 731, 541))
        self.groupBox.setObjectName("groupBox")
        self.label = QtWidgets.QLabel(self.groupBox)
        self.label.setGeometry(QtCore.QRect(10, 30, 61, 16))
        self.label.setObjectName("label")
        self.lineEdit = QtWidgets.QLineEdit(self.groupBox)
        self.lineEdit.setGeometry(QtCore.QRect(10, 50, 191, 21))
        self.lineEdit.setObjectName("lineEdit")
        self.label_2 = QtWidgets.QLabel(self.groupBox)
        self.label_2.setGeometry(QtCore.QRect(10, 80, 101, 16))
        self.label_2.setObjectName("label_2")
        self.lineEdit_2 = QtWidgets.QLineEdit(self.groupBox)
        self.lineEdit_2.setGeometry(QtCore.QRect(10, 110, 191, 21))
        self.lineEdit_2.setObjectName("lineEdit_2")
        self.label_3 = QtWidgets.QLabel(self.groupBox)
        self.label_3.setGeometry(QtCore.QRect(10, 140, 101, 16))
        self.label_3.setObjectName("label_3")
        self.lineEdit_3 = QtWidgets.QLineEdit(self.groupBox)
        self.lineEdit_3.setGeometry(QtCore.QRect(10, 170, 191, 21))
        self.lineEdit_3.setObjectName("lineEdit_3")
        self.label_4 = QtWidgets.QLabel(self.groupBox)
        self.label_4.setGeometry(QtCore.QRect(10, 200, 101, 16))
        self.label_4.setObjectName("label_4")
        self.lineEdit_4 = QtWidgets.QLineEdit(self.groupBox)
        self.lineEdit_4.setGeometry(QtCore.QRect(10, 230, 191, 21))
        self.lineEdit_4.setObjectName("lineEdit_4")
        self.label_5 = QtWidgets.QLabel(self.groupBox)
        self.label_5.setGeometry(QtCore.QRect(10, 270, 101, 16))
        self.label_5.setObjectName("label_5")
        self.lineEdit_5 = QtWidgets.QLineEdit(self.groupBox)
        self.lineEdit_5.setGeometry(QtCore.QRect(10, 300, 191, 21))
        self.lineEdit_5.setObjectName("lineEdit_5")
        self.label_6 = QtWidgets.QLabel(self.groupBox)
        self.label_6.setGeometry(QtCore.QRect(10, 340, 101, 16))
        self.label_6.setObjectName("label_6")
        self.lineEdit_6 = QtWidgets.QLineEdit(self.groupBox)
```

```

self.lineEdit_6.setGeometry(QtCore.QRect(10, 370, 191, 21))
self.lineEdit_6.setObjectName("lineEdit_6")

self.pushButton = QtWidgets.QPushButton(self.groupBox)
self.pushButton.setGeometry(QtCore.QRect(590, 50, 75, 23))
self.pushButton.setObjectName("pushButton")
self.comboBox = QtWidgets.QComboBox(self.groupBox)
self.comboBox.setGeometry(QtCore.QRect(370, 50, 201, 21))
self.comboBox.setObjectName("comboBox")
self.comboBox.addItem("Apple")
self.comboBox.addItem("Tesla")
self.comboBox.addItem("Microsoft")

self.label_8 = QtWidgets.QLabel(self.groupBox)
self.label_8.setGeometry(QtCore.QRect(370, 20, 151, 16))
self.label_8.setObjectName("label_8")

self.retranslateUi(Form)
QtCore.QMetaObject.connectSlotsByName(Form)

self.pushButton.clicked.connect(self.hesapla)

def hesapla(self):
    cnt=self.comboBox.currentText()

    ad=self.lineEdit.text()

    donen = self.lineEdit_2.text()
    donen_int = int(donen)

    duran = self.lineEdit_3.text()
    duran_int = int(duran)

    kvyk = self.lineEdit_4.text()
    kvyk_int = int(kvyk)

    uvyk = self.lineEdit_5.text()
    uvyk_int = int(uvyk)

    ozkaynak = self.lineEdit_6.text()
    ozkaynak_int = int(ozkaynak)

    cari=round((donen_int/kvyk_int),2)

    if cnt == "Apple":
        msg = QMessageBox()
        msg.setWindowTitle("Hesap Sonucu")
        msg.setIcon(QMessageBox.Information)
        msg.setText('Rasyo Değerleri')
        msg.setDetailedText('Cari Rasyo: '+str(cari))
        msg.exec()

        #Veri Çekme Kısmı
        r =
requests.get("https://www.gurufocus.com/term/current_ratio/NAS:AAPL/Current
-Ratio/Apple#:~:text=The%20current%20ratio%20is%20a,2020%20was%201.47.")
        soup = BeautifulSoup(r.content)
        linkler = soup.find_all("strong")

```



```

list = []
for link in linkler:
    list.append(link.text)
print((list[4]))

#matplotlib ile tablo oluşturma
dgr = list[4]
dgr_int=float(dgr)
x = [cnt, ad]
energy = [dgr_int, cari]
x_pos = [i for i, _ in enumerate(x)]
plt.bar(x_pos, energy, color='green')
plt.xlabel(ad)
plt.ylabel(cnt)
plt.title("Cari Rasyo Karşılaştırması")
plt.xticks(x_pos, x)
plt.show()

elif cnt == "Tesla":
    print("Tesla")
    r =
requests.get("https://www.gurufocus.com/term/current_ratio/NAS:TSLA/Current
-Ratio/Tesla")
    soup = BeautifulSoup(r.content)
    linkler = soup.find_all("strong")
    list = []
    for link in linkler:
        list.append(link.text)
    print((list[4]))

    dgr = list[4]
    dgr_int = float(dgr)
    x = [cnt, ad]
    energy = [dgr_int, cari]
    x_pos = [i for i, _ in enumerate(x)]
    plt.bar(x_pos, energy, color='green')
    plt.xlabel(ad)
    plt.ylabel(cnt)
    plt.title("Cari Rasyo Karşılaştırması")
    plt.xticks(x_pos, x)
    plt.show()
else:
    print("Microsoft")
    r =
requests.get("https://www.gurufocus.com/term/current_ratio/NAS:MSFT/Current
-Ratio/Microsoft")
    soup = BeautifulSoup(r.content)
    linkler = soup.find_all("strong")
    list = []
    for link in linkler:
        list.append(link.text)
    print((list[4]))

    dgr = list[4]
    dgr_int = float(dgr)
    x = [cnt, ad]
    energy = [dgr_int, cari]
    x_pos = [i for i, _ in enumerate(x)]
    plt.bar(x_pos, energy, color='green')

```

```

        plt.xlabel(ad)
        plt.ylabel(cnt)
        plt.title("Cari Rasyo Karşılaştırması")
        plt.xticks(x_pos, x)
        plt.show()

def retranslateUi(self, Form):
    _translate = QtCore.QCoreApplication.translate
    Form.setWindowTitle(_translate("Form", "Form"))
    self.groupBox.setTitle(_translate("Form", "Bilgiler"))
    self.label.setText(_translate("Form", "İşletme Adı"))
    self.label_2.setText(_translate("Form", "Dönen Varlık Değeri"))
    self.label_3.setText(_translate("Form", "Duran Varlık Değeri"))
    self.label_4.setText(_translate("Form", "KVYK Varlık Değeri"))
    self.label_5.setText(_translate("Form", "UVYK Varlık Değeri"))
    self.label_6.setText(_translate("Form", "Özkaynaklar"))
    self.pushButton.setText(_translate("Form", "Karşılaştır"))
    self.label_8.setText(_translate("Form", "Karşılaştırmak İsteddiğiniz
Firma"))

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Form = QtWidgets.QWidget()
    ui = Ui_Form()
    ui.setupUi(Form)
    Form.show()
    sys.exit(app.exec_())

```