

# Sabancı Universitesi

PROJ302

INTERNSHIP REPORT

MACHINE LEARNING & MICROPROCESSOR BASED SYSTEM DESIGN

BARIŞCAN KÖSE - 00025413

11.06.2021 - 13.08.2021

IDEAKTİF DANIŞMANLIK & YAZILIM SAN. TİC. LTD. ŞTİ.

SUPERVISOR : İBRAHİM TEKİN

03.10.2021

FACULTY OF ENGINEERING AND NATURAL SCIENCES

SABANCI UNIVERSITY

ELECTRONICS ENGINEERING PROGRAM

## ABSTRACT

Throughout my PROJ302 Internship course which I completed during the summer months of 2021 in a software consultancy company “Ideaktif Danışmanlık & Yazılım Ltd. Şti.” which is situated in the Depark Technology Development Building of Dokuz Eylül University, Izmir, Turkey, I completed 2 internship projects related to machine learning and microprocessor based system design. Firstly, I gathered the sales data from an e-commerce solutions provider company selling L’Oreal products in Trendyol called Saldos, then preprocessed it, created a model of machine learning for it, trained and tested it using mostly Tensorflow and Pandas libraries. Then I developed the code that allowed the said company to make sales number predictions thanks to the deep neural network model I trained, using just an excel file on their part by inputting various criterias of time, discount rate, buyboxness, keyword rank etc. For my second project there, I assembled a “SOSCode panic button” for alerting the security personnel in emergency situations in hospitals using ESP-32S Wifi + Bluetooth Dual Mode Development Board, CR2477 iBeacon-type beacons, a FC-75 battery charger, a PX402050 3.7 V battery and a 4 pin push button. Utilizing the Arduino IDE for the code part, I made the ESP-VROOM-32 chip connect to the local wifi network during the setup part and triggered by the button push, I made the BLE part scan for the nearest beacon and sent its location data to a server via an API. I’m very satisfied with my experience as I have completed both my projects.

## TABLE OF CONTENTS

1. INTRODUCTION	4
2. COMPANY INFORMATION	4
3. PROJECT BACKGROUND	5
3.1 Department Information	5
3.2 Status of the project and/or the problem at the beginning	6
3.3 Motivation and/or problem definition	6
3.4 Related Literature	7
4. INTERNSHIP PROJECT	7
4.1 Project objective	7
4.2 My responsibilities	8
4.3 Methodology / tools	9
4.4 Expected outcome and deliverables	11
4.5 Details	11
4.6 Results	13
5. INTERNSHIP EXPERIENCE	14
5.1 Learning	14
5.2 Relation to undergraduate education	14
5.3 Difficulties	15
5.4 A typical day	15
6. CONCLUSIONS	16
7. RECOMMENDATIONS	16
8. REFERENCES	17
9. APPENDIX	18

## 1. INTRODUCTION

This internship report is written in order to convey the experience of my PROJ302 internship class which, for me, was centered around 2 projects about microprocessor based system design and creating machine learning models for a company situated in DEPARK, the technology development HQ for the Dokuz Eylül University in Izmir. This report is structured in a way as to provide details of my 2 month long internship. Information about the company that I worked at, project's background, internship project, internship experience, conclusions for the overall experience, recommendations for future PROJ302 students and references are given in this order.

## 2. COMPANY INFORMATION

The company that I interned at is called “Ideaktif Danışmanlık & Yazılım San. Tic. Ltd. Şti.”. It's situated in “Adatepe Neighb. Doğu St. 207/AG D.E.U. Tınaztepe Campus Depark Technology Development Zone, Beta Building Floor:2 No:203 Buca,Izmir,Turkey”. You can reach the office via telephone through the number “+90 (232) 453 1 433”. The company has quite a pretty web page which can be accessed in “ideaktif.com.tr”.

The company is quite a bit old, having passed the age of 5. It is a small company that was founded in 2015 after a few friends in a different company came under a different leadership. The founder's name is Volkan Kuyucuoğlu which has multiple collegiate level educational achievements which includes a doctorate and is currently trying for a PhD. The company had 5 people in total, including 2 co-founders. They are mostly specialised in Industry 4.0 & ERP solutions, consulting, hardware products, mobile applications, netsis support and software development. The company is also supported by the Science, Industry

and Technology Ministry of Turkey . The name “Ideaktif” comes from the combination of the words “ide” and “aktif (active)”.

The company has only one facility or what we might call an office which is in the address above. Everybody works there as it is not that big of a company.

There is no parent or partner companies.

Main competitors are “Sisbim”, “Egebimtes”, “DNA Proje” and “Apeks Teknoloji”.

It produces software applications and Industry 4.0 & ERP solutions and gives Netsis support.

Major customers by name are “Mayr-Melnhof Graphia”, “İzmir Fırça”.

Major suppliers are “LOGO” and “Microsoft”.

There are 5 people who work in the company.

Mr.Volkan is the co-founder and the leader. Mr. Anıl (Aykutluğ) is the other co-founder. Mr. Anıl (Karahan) is the manager and has a hand in almost everything the company does. Mr. Alican handles the netsis support mainly. Mr. Emre developes software applications mainly focused in backend. And I did the machine learning and microprocessor based system design for the time being.

### 3. PROJECT BACKGROUND

#### 3.1 Department Information

The department I worked at was managed under Mr. Anıl Karahan. He is the Tech Lead of the company. His email is “ anilkarahan@ideaktif.com.tr”. He was the only person I worked closely with and he also sort of became my mentor. Our department created a machine learning model for Saldos, an e-commerce solution provider, and a microprocessor based system design for hospital emergency situations needed by Depark.

### 3.2 Status of the project and/or the problem at the beginning

I had 2 projects in my internship. Saldos needed a machine learning model to predict how much sales it would accrue from particular products at any time of the day. The project was non-existent before I began building it. Mr. Selim, the head of Saldos, wanted me to do this project. On the other hand, Ideaktif was supposed to complete some projects approved by Depark in order to keep existing in Depark's building. So Mr. Volkan, wanted to build an emergency detection and support system for hospitals in cases of child abduction or assaults on doctors. This project was also not started yet and I started building it when I got there.

### 3.3 Motivation and/or problem definition

For my machine learning project, I had to predict how much the top 10 products would be sold at any hour, of any day, of any month, its keyword rank, if they had a proper thumbnail or not, if it was a buybox product or not, with different prices and view numbers and discount percentages. All this data had to be taken, sifted through and applied to the model in order to save money on the advertisement spending and to employ strategical purchases.

For my microprocessor based system design project, I had to put iBeacon beacons in every floor of hospitals in order to triangulate the position of a beacon transmitter inside the hospital walls whenever a panic button was pressed. The panic button would be embedded in the ID cards of the holder and when the panic button on the ID card was pressed, it would send the location of the holder to a server where it would alert the security personnels.

### 3.4 Related Literature

For the machine learning project, Tensorflow , Keras and ScikitLearn libraries are very helpful. Tensorflow has very good documentation (1) which allowed me to quickly dive into machine learning and especially deep learning because I used deep neural networks in order to predict the sales numbers. I wanted to use deep neural networks instead of multiple linear regression models because it provided better predictions and estimates and because many things in life are not linearly dependent. I also did sensitivity analysis in order to find out what factor contributed the most to the sales numbers in order to make sure the company paid extra attention to it. (2)

For the microprocessor based system design project, I used many good documentations that explains how to connect ESP32 Dev Module boards to iBeacon-type beacons and how to use Arduino IDE while doing so (3) . I also sifted through datasheets in order to find out how much voltage should be applied to the system (4) .

## 4 INTERNSHIP PROJECT

### 4.1 Project objective

For the machine learning project, I will just be focusing on top 10 best seller products. The respective data of each products sales will be gathered together. What is important about the sales numbers is its corresponding hour, day, month, views, keyword rank, discount rate, sales price, if it was buybox or not, if it had correct thumbnail or not. Saldos already tracks this data so there is no need for web scraping. Other factors are not important. Some of the features are not independent but they give better prediction so we keep them. If we can predict how much a product will sell at any point in time and with how many views using Tensorflow libraries which include deep neural network model implementations in Python, we can thus increase the price of the product to get better

profits. We could spend more money on advertisement by improving the description or the thumbnail to increase views, if it means in the end it will be more profitable.

For the microprocessor based system design project, any person that is employed by the hospitals also carry a risk of being assaulted or a vulnerable person can be abducted. These people will be given ID cards that hang from their necks. In the ID card, we will have embedded our ESP32-Vroom-32 which has both bluetooth and wifi capabilities. This motherboard will talk to the iBeacon-type beacons placed in the middle height of certain rooms in order to not to scramble the triangulation with the higher or lower floors. The beacons have limited ranges and their transmission strength decreases as we get further away from the beacon. So, we will calculate the nearest beacon and get its inherent data and cross-reference it with the locations and send the location to the server that alerts security personnel in the event of a panic. This will all be done any time a card holder presses the “SOSCode” panic button. One thing out of the scope of the project is the amount of time the card has to stay operational, so battery power management is not very crucial for the completion of this project.

#### 4.2 My responsibilities

For the machine learning project, I will be taking data from 4 different excel spreadsheets which are exported from PowerBI. I have to put all the data with the same timestamp to one complete SQL table. Then I will take all the data from the SQL Server to a Pandas dataframe in python. I will then feature engineer the table and normalize the values in order to get better prediction. I will partition the data to test and training set then I will create a deep learning model, for example by 64x64 nodes, in order get accurate predictions. I am supposed to train the model from the data I get from the training set and test it from the data I get from the test set and display the results of the test set predictions in order to see if I have correctly predicted the sales number values. Then in a different excel sheet



which I directly upload to a Pandas dataframe later in the code, I am supposed to be able to display the sales numbers for different changing criteria like sales price or views.

For the microprocessor based system design project, first of all, I'm supposed to code the ESP32-Vroom-32 chip in order to process the data coming from the surrounding beacons. I'm supposed add a battery to the system and a button to the system. The button should scan the beacons everytime it is pressed but it shouldn't repeatedly connect to the wifi. iBeacon-type beacons' major and minor numbers should correctly be identified because later they will be important in determining which room has what major and minor numbers. I should also only scan the iBeacon-type beacons and not any other type beacons. There is also a need for a server which will receive JSON data and display the emergency location to the security personnel, so we will need to build some API. I'm supposed to solder all the wires necessary to create the system without any harm to the components and to pick proper battery and buttons because the voltage levels are very important not to burn the components.

#### 4.3 Methodology / tools

For the machine learning project, I used pyodbc to connect to the SQL Server from Python in VS Code. Sent the data to a Pandas dataframe in numpy format. I used Tensorflow for the machine learning part. I used Keras to implement the layers and preprocess the data. I used seaborn library to represent the end results.



**Figure 1. Solder Station.**

For the microprocessor based system design project, I used the ESP-32 bluetooth-low-energy libraries developed under the espressif umbrella by the author Neil Kolban. I utilized Arduino IDE and imported the necessary ESP32 Dev Module board library in order to work with my specific hardware which I soldered together (see Figure 1). I connected the system to the wifi in the setup part of the code and implemented a button-triggered scan and sending the necessary result to the server in the loop part. I used Serial Monitor of the Arduino IDE during this whole time to check for proper augmentation as Arduino IDE uses a bit complicated language. I also tested the system by placing 2 different beacons in very far and very near locations but very close to each other. I read the button pin's voltage value for a trigger and if triggered, scanned, cross-referenced the location with the major and minor values and used a HTTP POST request to send the data to the API.

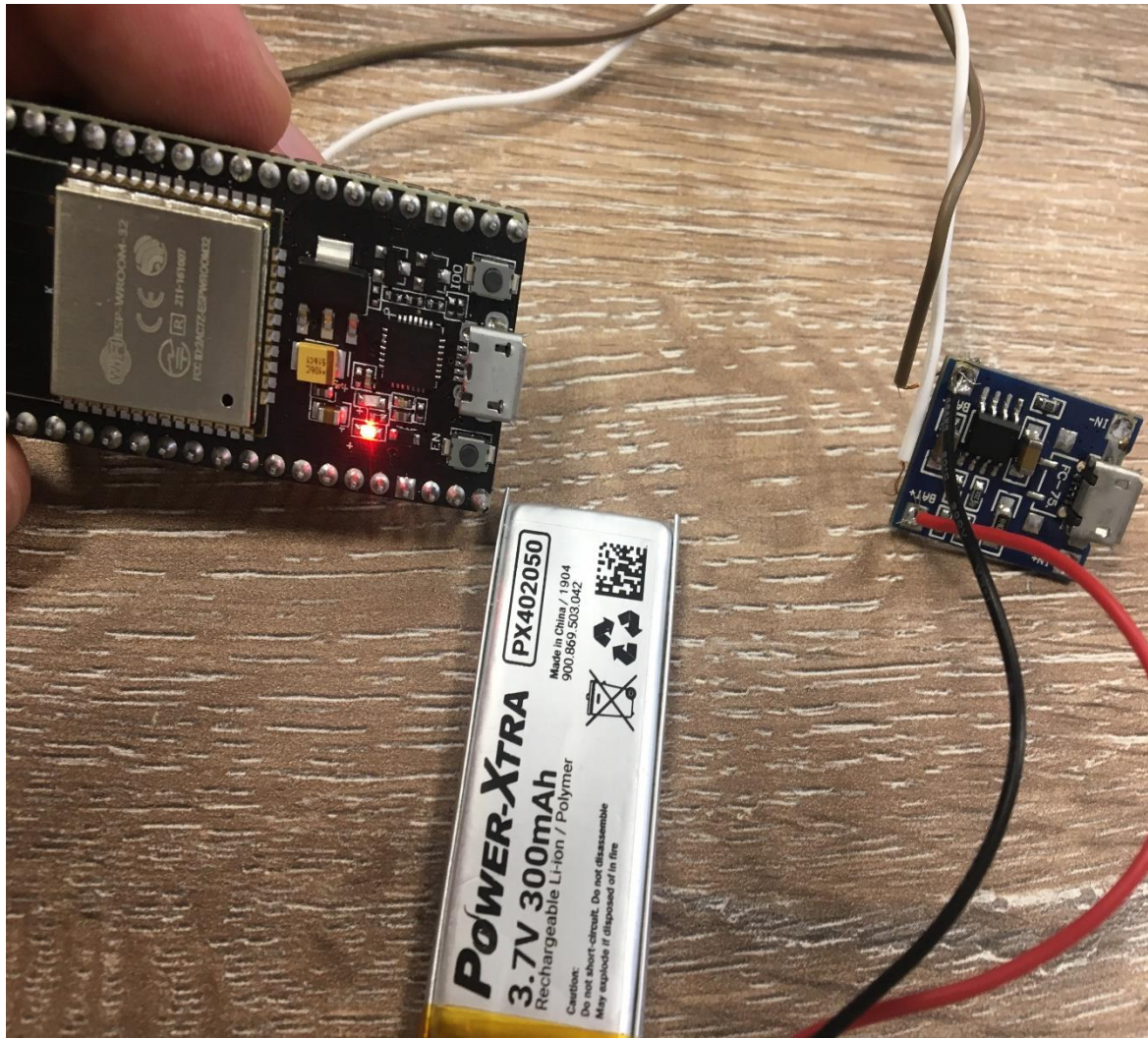
#### 4.4 Expected outcome and deliverables

For the machine learning project, I needed to be able to predict sales numbers with varying criterias which included the hour, views, sales price, discount rate and keyword rank mostly. And I had to show this all properly in a graph.

For the microprocessor based system design project, whenever the button was pushed, the system had to cross-referenced the location of the closest beacon's minor and major values and send that data to the API where we could finally see it on the server.

#### 4.5 Details

For the machine learning project, I first started talking with Mr.Selim who is the head of Saldos' İstanbul headquarters. He gave me 4 excel spreadsheet files. Pyodbc library allowed me to create a table and create specific columns with specific data types for those columns which took a lot of time. I had to execute SQL queries on Python. So I first sent all the data to their own SQL table and then took them back to Python, combined them with the corresponding timestamp and then added the data back to a complete table row by row by using "CREATE TABLE" and "INSERT INTO" query command. Then I got back the complete dataset, dropped the unnecessary features, partitioned the dataset to training and test set. I then normalized the numerical values using "tf.keras.utils.normalize" method and hotboxed the categorical columns. I also added the numerical columns to the feature columns set. I then created an input function in order to feed the data to the machine learning model by creating training, evaluation and prediction input functions. By using the "tf.estimator.DNNRegressor" method I created the model and then trained and evaluated it. I used matplotlib subplots to display the results in 4 different subplots in one figure. I used violin plot to show how much a feature was influencing the outcome and used a scatter plot to show how accurate my model was by displaying the predictions over the true values.



**Figure 2. The soldered system.**

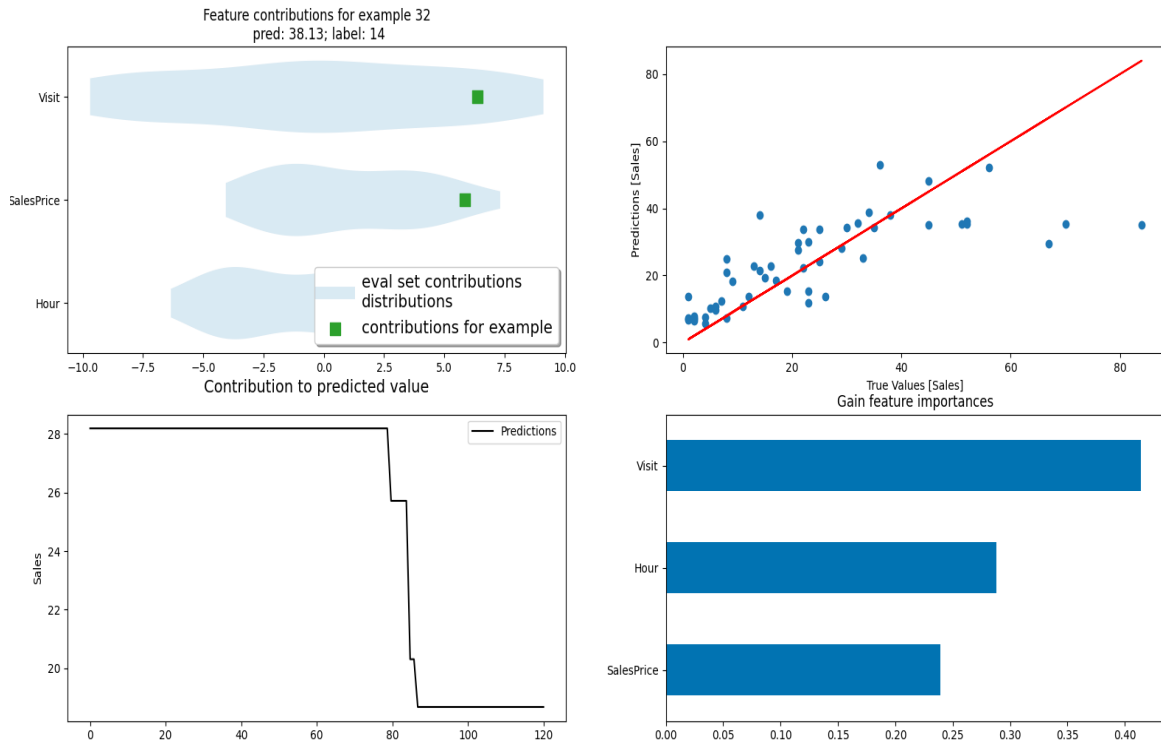
For the microprocessor based system design project, I soldered a “3.7 V 300 mAh POWER-XTRA PX402050” battery to the BAT- and BAT+ pins of the “FC-75” battery charger component(see Figure 2). This component, by the help of the micro-USB interface that it provides, allowed us to make the battery rechargeable. Then I soldered the IN- and IN+ pins of the “FC-75” to GND and 3V3 pins of the “ESP-32S Wifi + Bluetooth Dual Mode Development Board” which has 38 pins. I then soldered one pin of the “4 pin push button” to another GND pin in the ESP32 board and then soldered the other pin of the button to the pin number 33 but I used the pinMode method with the INPUT\_PULLUP parameter in the Arduino IDE to specify to use the internal pullup resistor in order make the pin not float.

For the code, I used Neil Kolban's ESP-32 BLE libraries and necessary wifi libraries. They included "BLEDevice", "BLEUtils", "BLEScan", "BLEAdvertisedDevice", "BLEBeacon", "HTTPClient", "Arduino" and "Wifi" libraries for ESP32. I defined the wifi SSID, wifi password, pin number for the button, server address beforehand. I used the setup part to specify the button pin number and connect to the local wifi. Then in the loop part, triggered by the button press which leads digital reading of the pin number 33 to become LOW, I start scanning for BLE devices which fit the characteristics of iBeacon beacons. In the "BLEAdvertisedDeviceCallbacks" function, I repeatedly looked for the device that was the closest to the ESP32 by checking if the advertised devices' RSSI was higher than the currently closest advertised device's RSSI since higher RSSI means closer device. Inside the loop part, I cross-referenced the major and minor numbers of the beacon with the pre-determined "location, major and minor number" tuples and sent the location data using a HTTP POST request with content-type of "application/json". I tested the results with iBeacon beacons which had the battery code of "CR2477".

#### 4.6 Results

Both the projects have been completed. I created the 4 graphs necessary to complete my machine learning project (see Figure 3) and I tested my SOSCode panic button project with 2 beacons many times. The machine learning project has been implemented and is a factor in guiding SALDOS buying and marketing strategies in order to create more profit. SOSCode panic button project hasn't yet been implemented as its casing is not done. The company bought a 3D printer and they are working on embedding it into an ID card and prolonging its operational time. But for this project, my responsibilities were done.





**Figure 3. ML Model Results.**

## 5 INTERNSHIP EXPERIENCE

### 5.1 Learning

I have learned that I can learn anything I need to if I keep on pushing and never giving up. I also learned how to work and communicate with co-workers. I have realized I liked machine learning and small-scale microprocessor based system designs and I might be looking to work on these topics.

### 5.2 Relation to undergraduate education

Understanding electricity in electronic components and how the inside of the chips with billions of transistors worked helped in making me feel more confident. I had learned coding beforehand thanks to the university's freedom to choose most of my classes and that also helped in creating a machine learning model. Working in groups in the university also helped me work harmoniously with co-workers.

There is no particular skill I wish I had learned at the university.

### 5.3 Difficulties

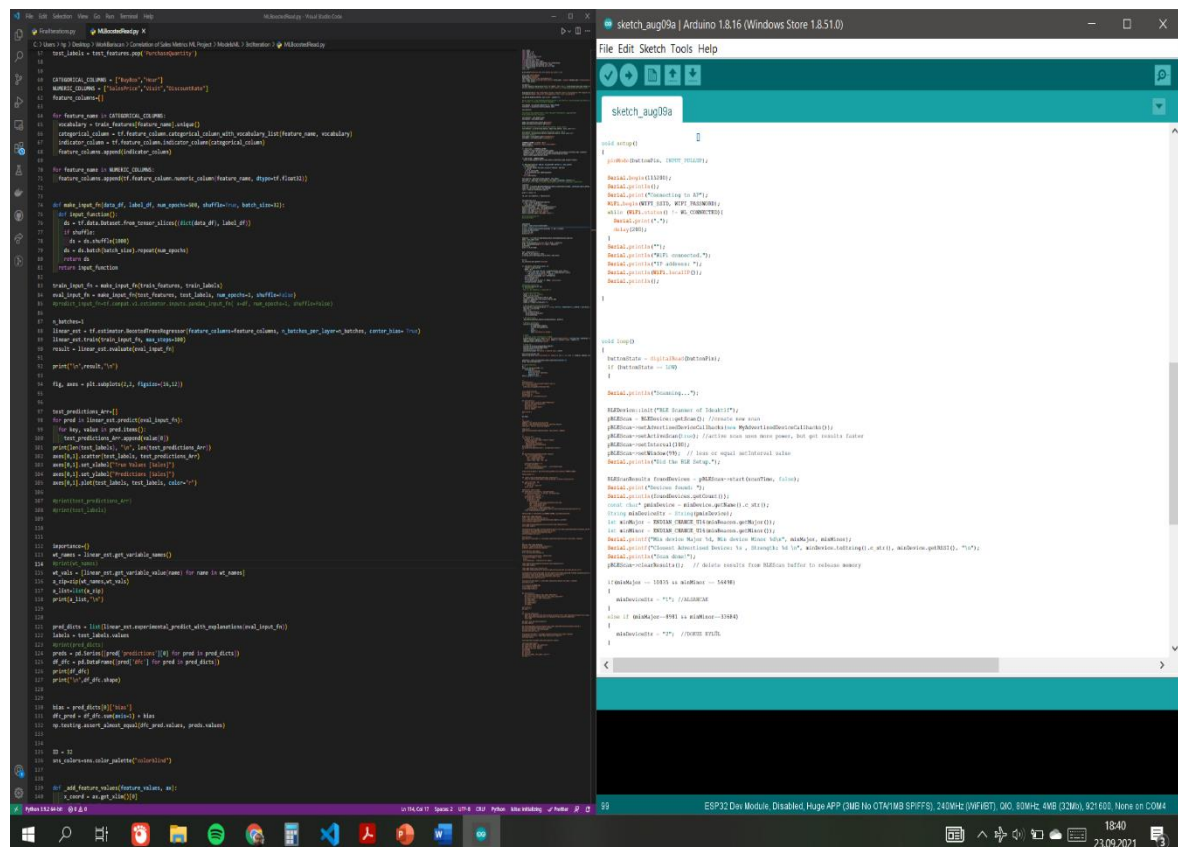
I kept feeling that I was out of my depth, but keeping on asking questions and being able to process them allowed me to do what I needed to do. I had help from my mentor in connecting all the electronic components without whom I couldn't have succeeded. I was almost not going to be paid, but a professional attitude and some communication skills I believe got me a minor pay which was nice.

### 5.4 A typical day

A typical day starts at 9 am. I say hi to everyone and get on my desk. Maybe make coffee for everyone and chat a bit. Then I take out everything related to my projects and start working. We have 2 monitors so I can see a lot of code and read a lot of text easily. I constantly googled the problematic aspects of my projects. Around 12, we order take-out food to the office. Until 4, I work and try to keep my focus but after around 4, it is harder to stay focused so we chat a bit and also work on the side. At 6 pm, we leave and I tuck everything away nicely before in order not to break anything.

## 6 CONCLUSIONS

In conclusion, it was a relief to know that I could hold my own under pressure of deadlines. I did 2 projects (see Figure 4) and also other side projects and was able to make connections and friends. It was nice and I felt sad when I had to leave. I am happy I could help SALDOS make profits and I am happy maybe someday I can help save someone's life when they are threatened at a hospital. I did both of these projects almost completely on my own. I'm more experienced and happier because I could exercise my knowledge which is the reason I believe we all learn things. Things in action are always more dynamic than just abstract ideas and they create significant memories.



**Figure 4. Narrow view of the codes I have written in VS Code and Arduino IDE.**

## 7 RECOMMENDATIONS

People shouldn't ignore their communication skills. And I believe people should actually do real projects in their internships. To me, it felt better than doing a fake internship.



## 8 REFERENCES

- 1 Core, Tensorflow (2021). “tf.estimator.DNNRegressor”. Tensorflow Core. Retrieved June 17,2021, from [https://www.tensorflow.org/api\\_docs/python/tf/estimator/DNNRegressor](https://www.tensorflow.org/api_docs/python/tf/estimator/DNNRegressor)
- 2 Core, Tensorflow (2021). “Gradient Boosted Trees: Model Understanding”. Tensorflow Core. Retrieved June 17, 2021, from [https://www.tensorflow.org/tutorials/estimator/boosted\\_trees\\_model\\_understanding](https://www.tensorflow.org/tutorials/estimator/boosted_trees_model_understanding)
- 3 Inc, Espressif (2016). “ESP32 Technical Reference Manual”. Espressif Inc. Retrieved July 26,2021, from [https://cdn-shop.adafruit.com/product-files/3269/esp32\\_technical\\_reference\\_manual\\_en\\_0.pdf](https://cdn-shop.adafruit.com/product-files/3269/esp32_technical_reference_manual_en_0.pdf)
- 4 Santos, R. (2016). “Getting Started with ESP32 Bluetooth Low Energy (BLE) on Arduino IDE”. Random Nerd Tutorials. Retrieved July 10,2021, from <https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/>

## 9 APPENDIX

Some of my machine learning Code for my first project. I was told not to disclose too much of the code at the company :

”

...

```
drop_features=["Month","Day","ProductName","Barcode","ContentId","Seller","VisualIntegrity","PSF","Keyword","KeywordRank"]
corr_features=["BuyBox","SalesPrice","PurchaseQuantity","Hour","Visit","DiscountRate"]
```

```
sql_query[0].drop(drop_features, axis='columns', inplace=True)
```

```
#df=pd.read_excel(r'C:\Users\hp\Desktop\Work\Correlation of Sales Metrics ML Project\SalDOSData\Sky_Deneme.xlsx')
#df = df.loc[:, ~df.columns.str.contains('^Unnamed')]
```

```
train_dataset = sql_query[0].sample(frac=0.8, random_state=0)
test_dataset = sql_query[0].drop(train_dataset.index)
```

```
test_results={}
```

```
#sns.pairplot(train_dataset[["Sales","Visit","Discount","SatisFiyati"]], diag_kind="kde")
#train_dataset.describe().transpose()
```

```
train_features = train_dataset.copy()
test_features = test_dataset.copy()
```

```
buybox_train_feature=train_features.pop("BuyBox")
buybox_test_feature=test_features.pop("BuyBox")
```

```
#train_features=tf.keras.utils.normalize(train_features, axis=0, order=2)
train_features = pd.concat([train_features, buybox_train_feature], axis=1, join='inner')
```

```
#test_features=tf.keras.utils.normalize(test_features, axis=0, order=2)
test_features = pd.concat([test_features, buybox_test_feature], axis=1, join='inner')
#print(train_features)
train_labels = train_features.pop('PurchaseQuantity')
test_labels = test_features.pop('PurchaseQuantity')
```

```
CATEGORICAL_COLUMNS = ["BuyBox","Hour"]
NUMERIC_COLUMNS = ["SalesPrice","Visit","DiscountRate"]
feature_columns=[]
```

```
for feature_name in CATEGORICAL_COLUMNS:
    vocabulary = train_features[feature_name].unique()
```

```

    categorical_column = tf.feature_column.categorical_column_with_vocabulary_list(feature_name, vocabulary)
    indicator_column = tf.feature_column.indicator_column(categorical_column)
    feature_columns.append(indicator_column)

for feature_name in NUMERIC_COLUMNS:
    feature_columns.append(tf.feature_column.numeric_column(feature_name, dtype=tf.float32))

def make_input_fn(data_df, label_df, num_epochs=500, shuffle=True, batch_size=32):
    def input_function():
        ds = tf.data.Dataset.from_tensor_slices((dict(data_df), label_df))
        if shuffle:
            ds = ds.shuffle(1000)
        ds = ds.batch(batch_size).repeat(num_epochs)
        return ds
    return input_function

train_input_fn = make_input_fn(train_features, train_labels)
eval_input_fn = make_input_fn(test_features, test_labels, num_epochs=1, shuffle=False)
#predict_input_fn=tf.compat.v1.estimator.inputs.pandas_input_fn( x=df, num_epochs=1,
shuffle=False)

n_batches=1
linear_est = tf.estimator.BoostedTreesRegressor(feature_columns=feature_columns, n_batches_per_layer=n_batches, center_bias= True)
linear_est.train(train_input_fn, max_steps=100)
result = linear_est.evaluate(eval_input_fn)

print("\n",result,"\n")

fig, axes = plt.subplots(2,2, figsize=(16,12))

test_predictions_Arr=[]
for pred in linear_est.predict(eval_input_fn):
    for key, value in pred.items():

...

```

Some of my microprocessor-based system design code for my second project:

```

...
#include <BLEAdvertisedDevice.h>
#include <BLEBeacon.h>

#include <HTTPClient.h>

```

```

#include <Arduio.h>
#if defined(ESP32)
    #include <WiFi.h>
#endif

#define ENDIAN_CHANGE_U16(x) (((x)&0xFF00) >> 8) + (((x)&0xFF) << 8))

#define WIFI_SSID "ideaktif"
#define WIFI_PASSWORD "2017idea2017";

const int buttonPin = 33;
int buttonState = 1;
const char* serverName = "https://192.168.3.249:89/default";
int scanTime = 3; //In seconds
BLEScan* pBLEScan;
BLEBeacon minBeacon;
BLEAdvertisedDevice minDevice;

class MyAdvertisedDeviceCallbacks:
public BLEAdvertisedDeviceCallbacks {
    void onResult(BLEAdvertisedDevice advertisedDevice)
    {

        if (advertisedDevice.haveName())
        {
            Serial.print("Device name: ");
            Serial.println(advertisedDevice.getName().c_str());
            Serial.println("");
        }

        if (advertisedDevice.haveServiceUUID())
        {
            BLEUUID devUUID = advertisedDevice.getServiceUUID();
            Serial.print("Found ServiceUUID: ");
            Serial.println(devUUID.toString().c_str());
            Serial.println("");
        }
        else
        {
            if (advertisedDevice.haveManufacturerData() == true)
            {
                std::string strManufacturerData = advertisedDevice.getManufacturerData();
                uint8_t cManufacturerData[100];
                strManufacturerData.copy((char *)cManufacturerData,
strManufacturerData.length(), 0);

```

```

        if (strManufacturerData.length() == 25 && cManufacturerData[0] == 0x4C &&
cManufacturerData[1] == 0x00)
        {
            Serial.println("Found an iBeacon!");
            BLEBeacon oBeacon = BLEBeacon();
            oBeacon.setData(strManufacturerData);
            Serial.printf("iBeacon Frame\n");
            Serial.printf("ID: %04X Major: %d Minor: %d UUID: %s Power: %d\n",
oBeacon.getManufacturerId(), ENDIAN_CHANGE_U16(oBeacon.getMajor()),
ENDIAN_CHANGE_U16(oBeacon.getMinor()),
oBeacon.getProximityUUID().toString().c_str(), advertisedDevice.getRSSI());
            if( minDevice.getRSSI() < advertisedDevice.getRSSI() )
...
    ”

```