Project Assignment

Umuthan Ercan
March 17, 2023

Abstract

This project report provides information on how I fulfilled our term project's requirements, which mainly focused on working on a business/real-life/research problem, obtaining a related data set, and performing datarelated operations on the way to a possible solution. I used Python packages and libraries I deemed necessary, tried to keep our code clean and readable, and commented on it to make it more understandable.

1 Introduction

In this term project, I were expected to work on a possible real-life concern that can be aided by machine learning techniques. I explored, discussed, and evaluated potential topics that can be used for our project, ended up with 5 prospects, and finally concluded on investigating obesity among adults. For this subject, I attained a data set called "Obesity among adults by country, 1975-2016" [1] and expected to observe relationships between years, countries, and obesity levels. I imported the data set into a Python environment, analyzed and cleaned data, and performed certain data operations. I also used OOP principles and visualized the acquisitions I obtained from processed data. The following sections will provide further details of our study.

2 Methodology

I used Anaconda as a main suite and Jupyter Notebook during code implementation. I also used the random, numpy, pandas, and matplotlib.pyplot, math, and sklearn libraries.

1. Retrieving the data
2. Object Oriented Programming
3. Data Pre-processing
4. Machine Learning Algorithms

2.1 Retrieving the data

After I decided on our subject, I started searching the web for a comprehensive data set based on obesity over years. I found one on Kaggle, but after exploring I thought it wasn't clean enough so I kept on searching on Kaggle. After some research, I found "Obesity among adults by country, 1975-2016" data set was already in CSV format. So firstly, I downloaded and examined the data set with the naked eye, to see if anything in particular will catch our attention. I checked if all columns are necessary, or are there any missing data, etc.

2.2 Object Oriented Programming

To generalize the data operations, I created a class named "Dataset" and gathered all the necessary functions in this class. This is the part of the project in which I implement object oriented programming and these functions should be usable for any Dataset that I would want to work with. Maybe some changes would be required in our functions to adapt to a certain Dataset, or that Dataset would need to go through some pre-processing to be fully compatible with the class methods but generally speaking I think I came up with some nice methods that provide adaptability and would be able to work properly with almost any given Dataset
In the init constructor, which is where I initialize our class instances I do some data preprocessing for each Dataset that I pass as an argument to the class and I prepare it so that our methods can work with it. Later on I defined the following methods inside the class to calculate relevant values (mostly statistical values):

- The "make number" method is an auxiliary method that tries to convert a certain value in a number and in case it fails it just returns False, this is done due to Datasets often providing numerical values as strings, which I need to change to a numerical type in order to be able to work with them for the rest of the statistical values, this method is called in the data preprocesing in the init function.
- "mean()" for calculating the mean of a specified column
- "var()" for calculating the variance of a specified column

- ”max()” for calculating the maximum element of a specified column
- ”min()” for calculating the minimum element of a specified colum I can see in the following figure the UML diagram of our class:

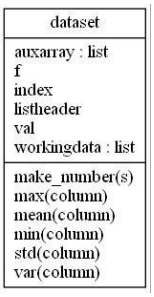

Figure 1: UML Diagram of the Dataset class.

2.3 Data Pre-processing

Later on, I started working with our specific data set, ”Obesity among adults by country, 1975-2016.csv”. These are the operations I performed for data pre-processing.

- I read the file, split, and stored all the elements in an empty list called ”workingdataLIST”.
- I extracted header elements from the list, and saved them into ”listheader”.
- While investigating the data set, I observed some values in the Obesity column for certain countries are missing, and they are saved as a string, ”No Data”. I generated a loop to check every element in the Obesity column, remove the rows that have a ”No Data” value, and numerate the remaining rows accordingly.
- While investigating the data set, I observed the first column is redundant so I removed the first column.
- I created a dictionary called ”CountryIndexes” and I created a loop to check and enumerate every unique country name.
- Later, I formed a new column in our ”workingdataLIST” with these unique numbers for each country.
- At this point, I pre-processed the data successfully, so I converted it to a pandas data frame and exported this data frame into a file named ”export dataframe.csv”.

2.4 Machine Learning Algorithms

For the Machine Learning Algorithms I have to come back to our main hypotheses and our project aims. To revise our hypotheses:

I want to prove that the fast paced society that I currently live in, comes with a very alarming side effect as an increase in obesity over the years. I agree that looking at our past, and especially at older generations such as our parents or even grandparents, there were not as many people with obesity, and it was much more normal for everyone to have a proper diet with home made food rather than eating junk food as often as I do nowadays. It is not that people from before had more time than I have today for cooking, but I think it is more related to how easy it is nowadays to go to a fast food restaurant, or to buy something pre-cooked.

The machine learning algorithms that I wanted to make use of initially were:

- A correlation matrix, to see how related obesity and the years are, I expect a high positive relation as the later the year the larger the obesity values are.
- A multiple linear regression to relate both the country and the years to predicted obesity so that I can predict the obesity value in a certain country on a certain year if the tendency of obesity increase over the years continues being the same.

Other than the algorithms above, I implemented three other algorithms with the same purpose:

- A normal linear regression function that would take a year and a country name as arguments and return the predicted value, isolating the values for a certain country creating a new set of data with the obesity values of just that country. This seemed a better option than the multiple linear regression taking into account the whole Dataset as it will be shown in the results part.
- With the exact same purposes as the above mentioned linear regression, I defined a polynomial regression function.

- Finally, I repeated once again both algorithms but this time including another argument to pass to the functions, it being ”sex” which indicates the gender of the wanted prediction. The same way that before I isolated the obesity values of just one certain country, with this constrain now I also will just take the values not only from the requested country but also from the requested gender, for example females in Spain. It is worth mentioning, for the polynomial regression in this case I wanted to do it as finely as possible, so I added a hyperparameter tuning process for the degree of the polynomial needed for the regression: whenever the function is called it will first calculate which is the optimal degree for the polynomial by calculating the mean squared error for the different degrees of the polynomial from 1 to 15. Another option that I implemented in case that the optimal degree selected by the optimizer would return strange results is that I can manually select the degree of the wanted polynomial as well. Below find included a snippet of this part of the code since it is the most relevant of our project.

```
def PolRegCountryGender (countryname, year, gender, ploterror = True, Manual = False):
    #in order to make better the polynomial regression I not only add the gender selection but
    #I add a hyperparameter optimization algorithm, in which I check the mean squared error of the
    #polynomial regression and I keep the polynomial degree that returns the minimum error

    #in any case I also add a manual selector for the polynomial degree just in case I want to try
    #with other degrees polynomials
    ValuesObesity = [] ValuesYear = [] for i in
    range(len(FinalWorkingData)):
        if FinalWorkingData['Country'][i] == countryname and FinalWorkingData['Sex'][i] == gender:
            ValuesObesity.append(FinalWorkingData['Obesity_(%)'][i])
            ValuesYear.append(FinalWorkingData['Year'][i])
    np_arrayYears = np.array(ValuesYear) np_Obesity =
    np.array(ValuesObesity)

    #fit polynomial regression model poly_reg_model =
    LinearRegression()


    #I implement a degree selector which is going to give us the most accurate
    #degree for our polynomial regression

    minmse=9999999 error=[] for
    i in range (1, 15):
        poly_features = PolynomialFeatures(degree=i, include_bias=False) #I iterate the features of the polynomial
            for each degree that I want to test
        X_poly = poly_features.fit_transform(np_arrayYears.reshape(-1,1)) #I fit the x data so it's suitable for the
            polynomial fitting
        poly_reg_model.fit(X_poly, np_Obesity) mse = 0
        for j in range(1975, 2017):
            mse = (mse + (np_Obesity[j-1975] - (#I calculate the mean squared error for each polynomial degree
                poly_reg_model.predict(poly_features.fit_transform(np.array([j]).reshape(1,
                    -1)))))**2) mse = mse /41 error.append(mse)#I append that calculated error to an error array
        so I can sitck with the minimum of the array

        if mse < minmse: minmse =
            mse optimaldegree = i

    #in case I want to plot the error, I can use ploterror = True which is the default value
    if ploterror == True:
        plt.plot(list(range(1,15)),error) plt.xlabel('Degree of the
        polynomial') plt.ylabel('Average mean squared error') plt.show()

    #here I set the degree of the polynomial based on whether I decided a manual degree or not
    if Manual == False:
        degree = optimaldegree if
        ploterror == True:
            print("\nthe optimal degree of the polynomial regression is:", optimaldegree)
```

```
else:
    degree = Manual if
    ploterror == True:
        print("\nthe degree of the polynomial regression is the manually selected:",
            Manual)

poly = PolynomialFeatures(degree, include_bias=False) #reshape data to
work properly with sklearn x_poly =
poly.fit_transform(np_arrayYears.reshape(-1,1))
poly_reg_model.fit(x_poly, np_Obesity) if ploterror == True:
    print("\nEstimated model parameters for polynomial regression", poly_reg_model.intercept_,
        poly_reg_model.coef_)

#use model to make predictions on response variable np_year =
poly.fit_transform(np.array(year).reshape(1,-1)) y_predicted =
poly_reg_model.predict(np_year)

#create scatterplot of x vs. y #plt.scatter(x, y)

#add line to show fitted polynomial regression model #plt.plot(x,
y_predicted, color='purple') return y_predicted, np_Obesity
```

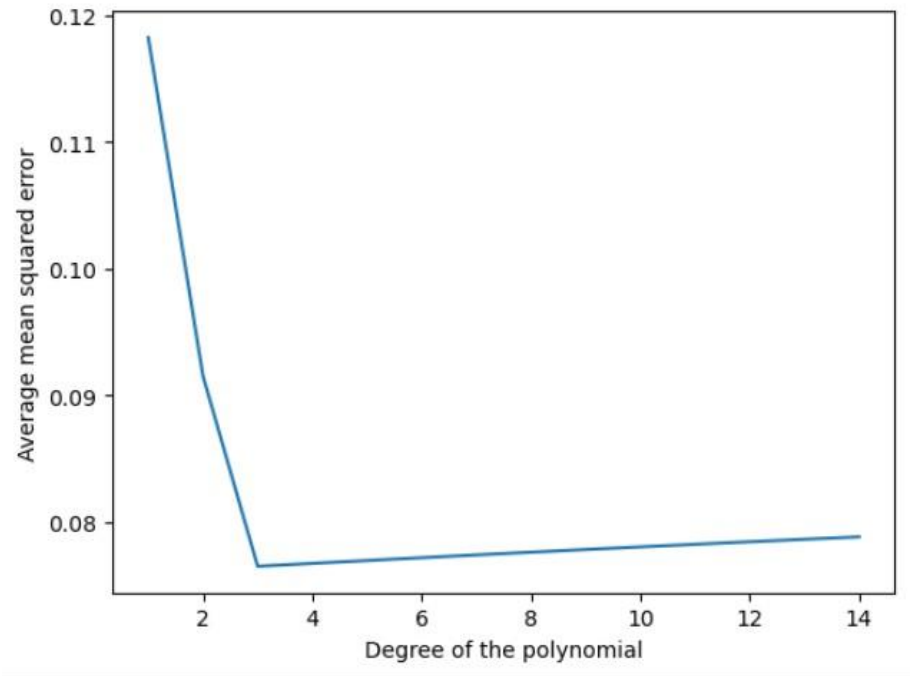

Figure 2: The mean squared error change for Females in Turkey in the year 1993.

3 Results

Coming to the results part, firstly I had the objective of checking with a correlation matrix if the variables *"year"* and *"Obesity*(%)", and as I can see in figure 2, I have a strong positive relationship between them.

	Year	Obesity_(%)	Country_ID
Year	1.000000	0.378862	0.000243
Obesity_(%)	0.378862	1.000000	0.058968
Country_ID	0.000243	0.058968	1.000000

Figure 3: This figure shows the Correlation Matrix results I obtained.

Moving on to the other project objective, I can observe that our first approach even if it gives a result, it does not really correlate to the real values in the Dataset that I encountered for each country. I understand that that is due to the formatting of the data and how for every country the years reset. The previously mentioned succession is making the multi-linear regression be only dependent on the year that I calculate it and the order in which the countries appear (alphabetic). For example if I would ask our algorithm to predict a value for Andorra and Afghanistan for the same year, just because they appear in a similar order in the Dataset, they would have very similar results.

Given that, I decided to isolate the countries that I were calculating the regressions for and that is how I came up with the other three algorithms that I will be appending results from:

- For the normal linear regression isolating the countries and taking all obesity values, I can see that the values gotten are already looking better than the previously gotten with the first approach. Although, as I were taking all the possible values for the ”sex” attribute it was returning an average of ”male”, ”female” and ”both sexes”, being ”both sexes” already an average of ”male” and ”female”, so that was not fully right.

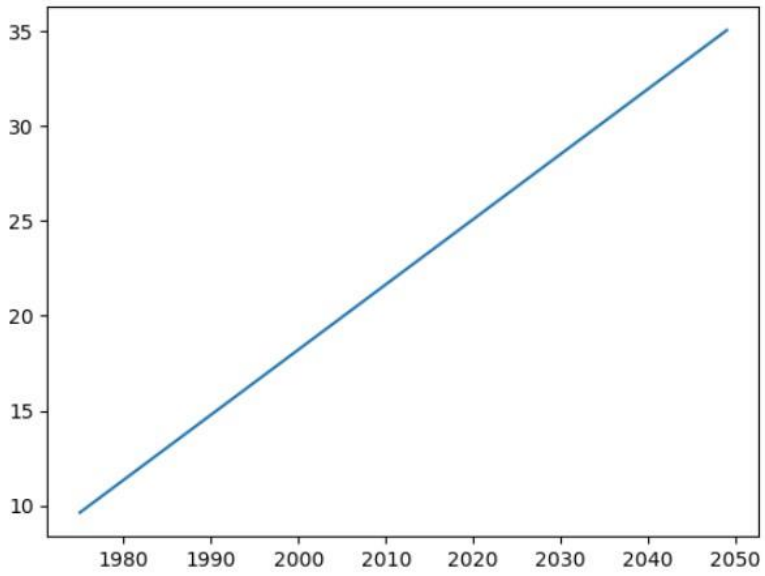

Figure 4: This figure shows the linear regression prediction for Spain until the year 2050.

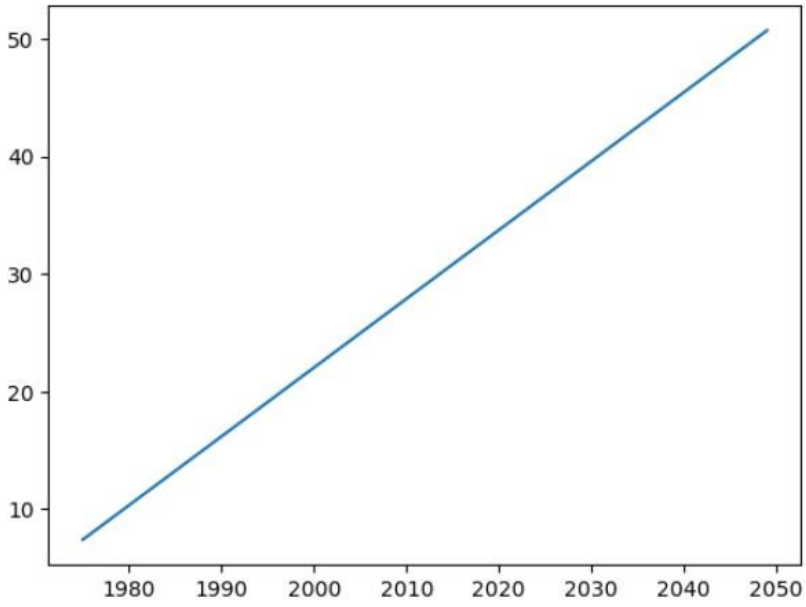

Figure 5: This figure shows the linear regression prediction for Turkey until the year 2050.

- With the exact same purposes as the above-mentioned linear regression, I defined a polynomial regression function. It is true that the data is quite linear but for some countries, I saw that there is some polynomial relation and the graphs have some curvature, so I wanted to be able to properly fit a curve to them. This is also a good way to see how clearly linear the tendency is since for example I can see that for Spain even using the polynomial regression, the relation shown is clearly linear.

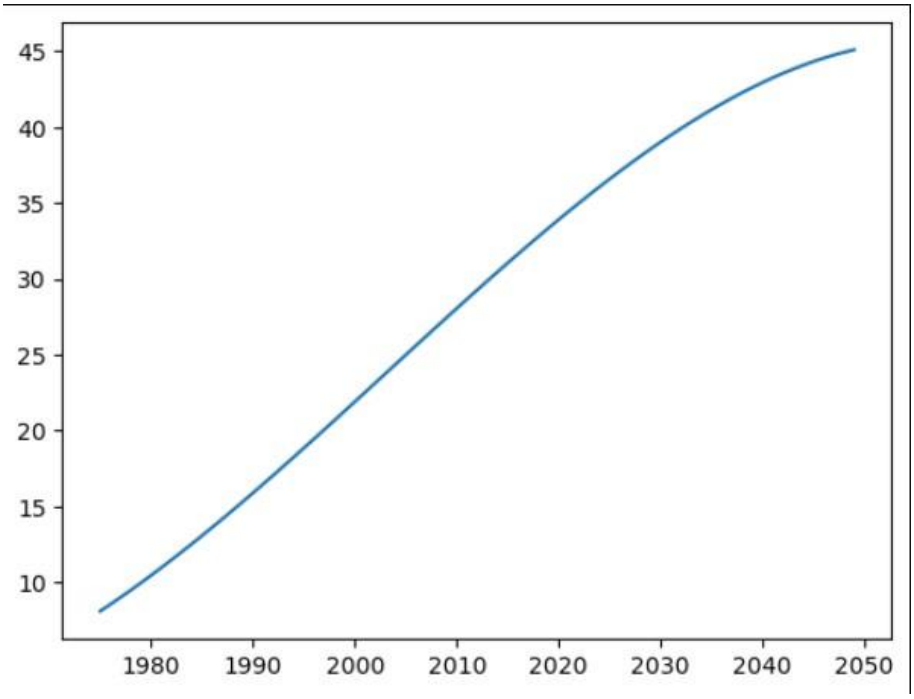

Figure 6: This figure shows the polynomial regression prediction for Turkey until the year 2050.

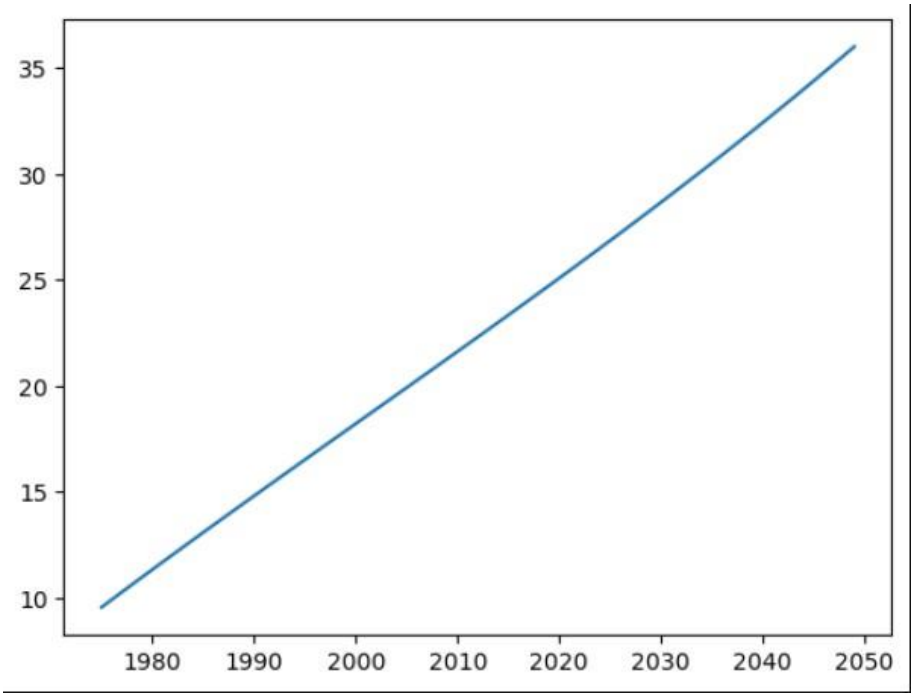

Figure 7: This figure shows the polynomial regression prediction for Spain until the year 2050.

- I now decided to improve both our previous linear and polynomial regression with the addition of being able to select the gender that I want to predict, since it is not the exact same procedure for both algorithms I will talk separately for each of them:

- For the linear regression as I said, I are now able to select which gender I want to predict, as well as the year and the country that I want the prediction from.

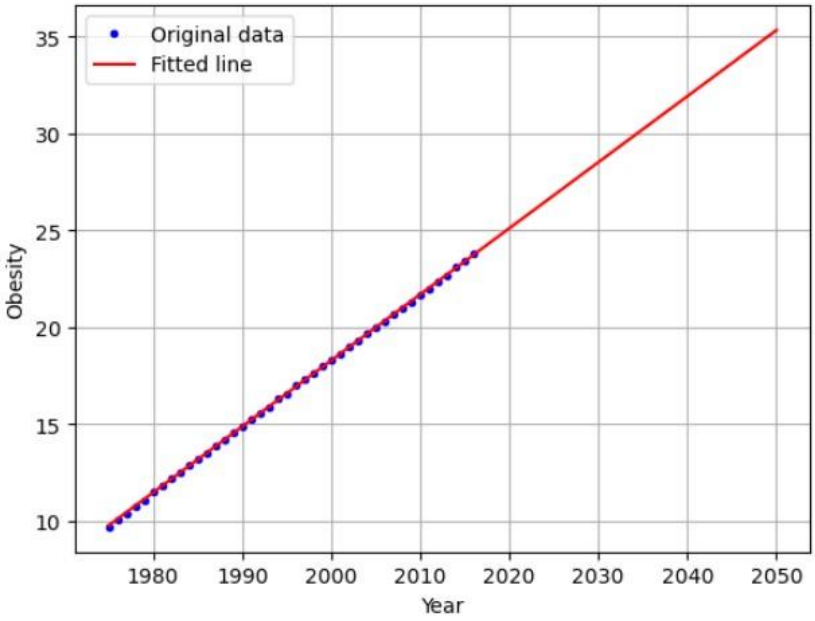

Figure 8: This figure shows the Linear regression prediction for both genders in Spain until the year 2050.

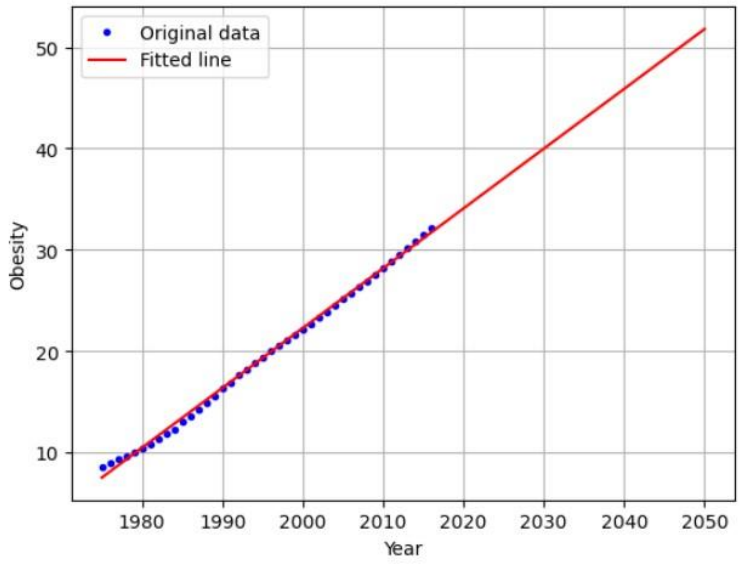

Figure 9: This figure shows the Linear regression prediction for both genders in Turkey until the year 2050.

- For the polynomial regression not only I implemented the gender selection for the prediction but I also added an optimization algorithm for the degree of the polynomial, this sometimes tends to overfit if it selects a very high degree or provide strange results (as you will be able to see with the prediction for females in Turkey after 2045 as the plot starts decreasing instead of increasing), thus I also added the possibility of manually selecting the degree of the polynomial.

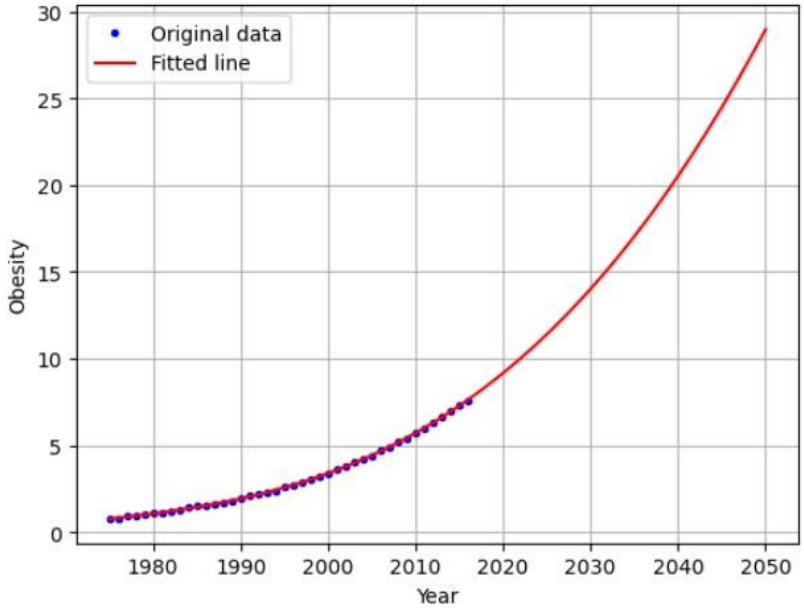

Figure 10: This figure shows the polynomial regression prediction for females in Afghanistan until the year 2050.

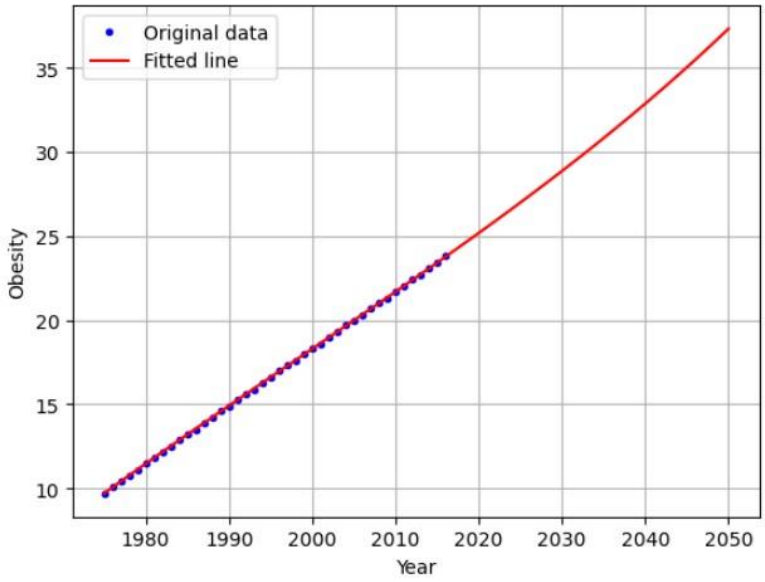

Figure 11: This figure shows the polynomial regression prediction for both sexes in Spain until the year 2050.

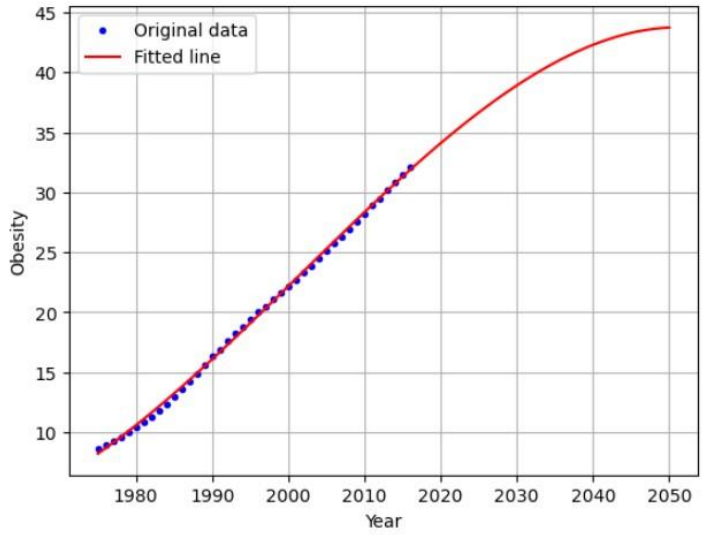

Figure 12: This figure shows the polynomial regression prediction for both sexes in Turkey until the year 2050.

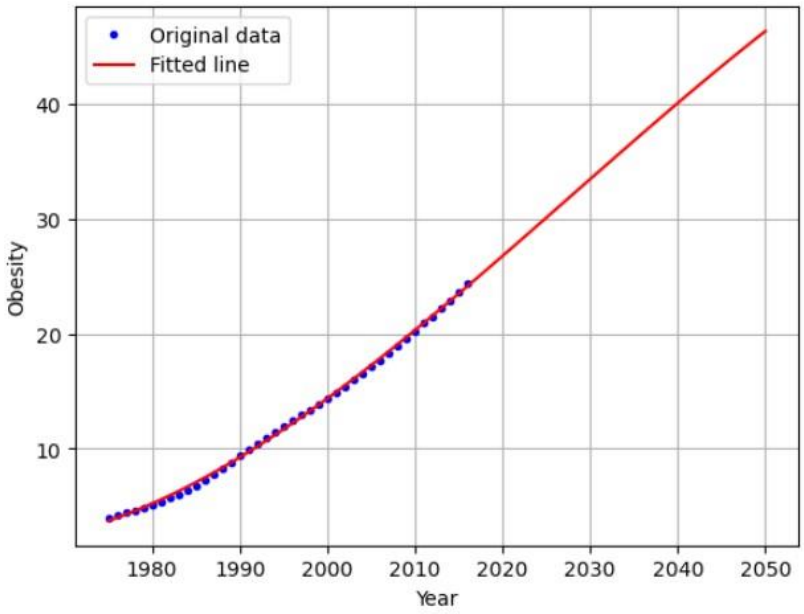

Figure 13: This figure shows the polynomial regression prediction for males in Turkey until the year 2050.

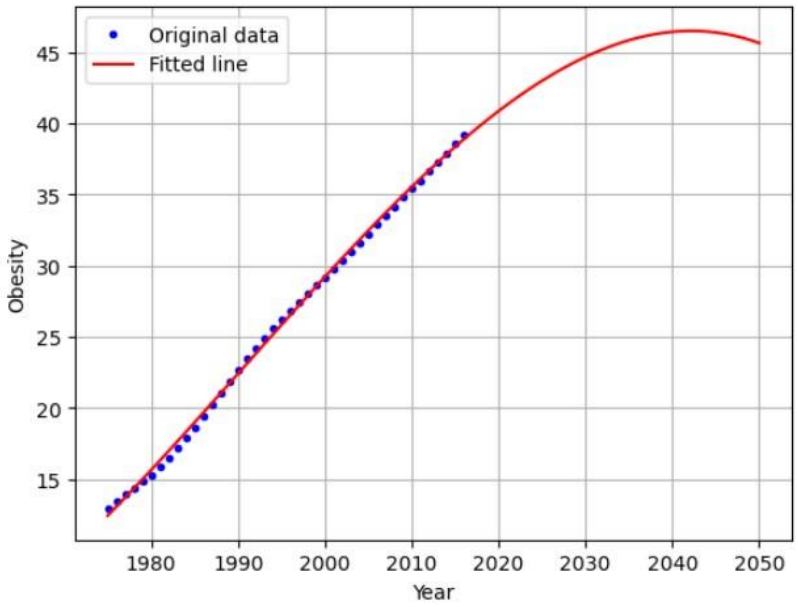

Figure 14: This figure shows the polynomial regression prediction for females in Turkey until the year 2050.

4 Discussion

As I have seen by our predictions in the results, they are quite worrying in case that the tendency continues being the same, but now our question is, will it really be the same?

With this discussion I want to encourage the readers to think about the reasons why this tendency kept going up and what can be done to stop this almost linear increase in the obesity over the years, I do not want to be unable to move by 2035 and I do believe that this tendency will stop being the way it is right now at some point.

I believe that our algorithms even if they do not have the most extensive amount of data to be based on clearly show a worrying trend, and even if they might not be the most accurate machine learning implementation to ever exist, they are still a model based on what has been happening in the last years and that is something that if continues going the same way will definetiley have a negative impact on society.

Even though I think that this tendency will stop at some point, I have no certainty that it will happen, and it is more a thought based on the fact that it seems illogical for us that the governments will not do anything about this, so what will the future bring?

Regarding data ethics, some might argue that our predictions can be offensive, but I intend no harm whatsoever with our project, instead I just intend to spread awareness that this is a real problem and I need to do something to stop it. I do not know how the data was collected but I would assume it was collected either from hospitals when they take measurements from people or from a survey.

In any case I would not say that this is violation of GDPR since it does not include the names of the people who are taken into account for this Dataset, and even if it would include them I believe it would be for the better since this is an important matter that concerns the overall health and well-being of our population.

References

[1] ”obesity among adults by country, 1975-2016”. https://www.kaggle.com/datasets/amanarora/ obesity-among-adults-by-country-19752016?select=obesity-cleaned.csv. Accessed: 02.03.2023.