**- Linear Regression**

**Prediction of custom variables by using Linear Regression**

1. **Introduction**

In this study, the main purpose is to find out the linear function which represents the relationship between dependent and independent variables. We are going to try to find the estimated values associated with minimum error values. Hence, the second purpose will be modeling the relationship between variables. Machine Learning modeling means a description of a system using mathematical concepts and language. The model will help to explain the system and to study the effects of different components, and to make predictions about variables.

1. **Material and Method**
2. **Covid-19 –Turkey Dataset**

Result of my Google researches, I found an opportunity to find **Covid-19 –Turkey Dataset (** which I attend into referances ) belong to “ozanerturk” on Github. Covid-19 –Turkey is a dataset corresponds to 11 columns which contains within itself: date, totalTests, totalCases, totalDeaths ,totalIntensiveCare, totalIntubated ,totalRecovered, tests ,cases, deaths ,recovered and 53 rows until now which represent days.Dataset updates by pulling data from the website of the Turkish Ministry of Health every 5 minutes.

1. **Linear Regression Theory**

Linearity in algebra refers to a linear relationship between two or more variables.If we draw this relationship in a two dimensional space between two variables, we get a straight line.In this study we progress by “**Yi=B0+B1.Xi**” formula. B0 and B1 are coefficients inside of the dataset. According to that equation, B0 and B1 coefficients are the section that what we say about “intelligence”.Under favour of B0 and B1 we minimalize the error value. Error equation: “Ei =Yi-Ýi” so that final equation of our study is “**Yi=B0+B1.Xi+Ei**”.

1. **Python Codes**

**c.1- Linear Regression Model For TotalCases-TotalDeaths**

#I import the pandas library here

import pandas as pd

df=pd.read\_csv("../CoronaReg/timeline.csv")

df.head() #and I used head to display limited my dataset below.

df.info() #by using info I wanted to recognise dataset here

# I called seaborn library here and created a jointplot to visualise relationship between Total Cases and Total Deaths.I chose which variables I want to joint and specified

#which database and what kind->regression

import seaborn as sns

sns.jointplot(x="totalCases",y="totalDeaths",data=df,kind="reg")

#what the graph says to us ? It says while cases are increasing deaths also going to increase

#then for modelling from sklearn I implement linear model then chose Linear #Regression

from sklearn.linear\_model import LinearRegression

#then I indicate my independent value

X=df[["totalCases"]]

X.head()

#I used head function to see what do I have on that column

#then I indicate my dependent value

#I used head function to see what do I have on that column

y=df[["totalDeaths"]]

y.head()

#I iniciate the Linear function

reg=LinearRegression()

#and I created a model object fit means set up here

model=reg.fit(X,y)

model

# crosscheck

str(model)

#to see which command can I use

dir(model)

model.intercept\_#according to my linear regression formula Yi=B0+B1.Xi I need B0 #and B1 coefficient so I use intercept and coef

model.coef\_#we will multiply that coefficients with Yi=B0+B1.Xi formula

model.score(X,y) # I took expression variable and its important

#Prediction part

#under favour of seaborn we virtualise the data I created a graph and

#put the dependent value to y and independent value to x

#our model multiplied the coefficients via our equation

import seaborn as sns

import matplotlib.pyplot as plt

g=sns.regplot(df["totalCases"],df["totalDeaths"],ci=None,scatter\_kws={'color':'green','s':9})

g.set\_title("Case-Death Equation : Deaths =-71.39+ Cases\*0.02")

#I put the label for x and y

g.set\_ylabel("TotalDeaths")

g.set\_xlabel("TotalCase")

#here I managed the start points x

plt.xlim(-2000,80000)

plt.ylim(-1000,2000)

#our model predicted the deaths number out of 85 million and found 2.145.398 people

model.predict([[85000000]])

**c.2- Linear Regression Model For TotalTests-TotalCases**

import pandas as pd

df=pd.read\_csv("../CoronaTrReg/timeline.csv")

df.head()

df.info()

import seaborn as sns

#this time I changed my independent value to totalTests and dependent value to #totalCases

sns.jointplot(x="totalTests",y="totalCases",data=df,kind="reg")

from sklearn.linear\_model import LinearRegression

X=df[["totalTests"]]

X.head()

y=df[["totalCases"]]

y.head()

reg=LinearRegression()

model=reg.fit(X,y)

model

str(model)

dir(model)

model.intercept\_

model.coef\_

model.score(X,y)

import seaborn as sns

import matplotlib.pyplot as plt

g=sns.regplot(df["totalTests"],df["totalCases"],ci=None,scatter\_kws={'color':'green','s':9})

g.set\_title("Test-Case Equation: Tests =1568.79+Cases\*8.05")

g.set\_ylabel("TotalCase")

g.set\_xlabel("TotalTest")

plt.xlim(0,155000)

plt.ylim(-1000,20000)

model.predict([[1000000]])

model.predict([[85000000]])

#in the end our model predicted 10 million 523 thousands 706 case out of 85 #million people

**c.3- Linear Regression Model For TotalCases-TotalRecovered**

import pandas as pd

df=pd.read\_csv("../CoronaTrReg/timeline.csv")

df.head()

df.info()

import seaborn as sns

#on third model I changed my independent value to totalCases and dependent value #to totalRecovered

import seaborn as sns

sns.jointplot(x="totalCases",y="totalRecovered",data=df,kind="reg")from sklearn.linear\_model import LinearRegression

X=df[["totalCases "]]

X.head()

y=df[["totalRecovered "]]

y.head()

reg=LinearRegression()

model=reg.fit(X,y)

model

str(model)

dir(model)

model.intercept\_

model.coef\_

model.score(X,y)

import seaborn as sns

import matplotlib.pyplot as plt

g=sns.regplot(df["totalCases"],df["totalRecovered"],ci=None,scatter\_kws={'color':'green','s':9})

g.set\_title("Case-Recover Equation : Recovered =-3655+ Cases\*0.272")

g.set\_ylabel("Recover")

g.set\_xlabel("Case")

plt.xlim(0,150000)

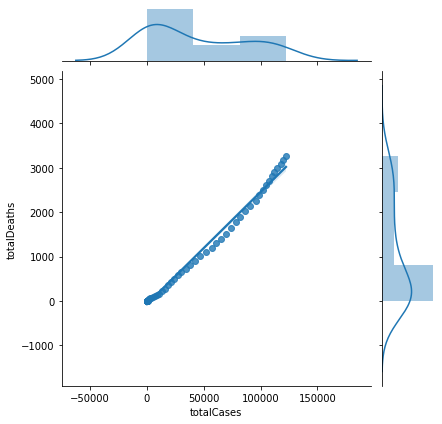
plt.ylim(-10000,80000) model.predict([[1000000]])

model.predict([[85000000]])

#finally our model predict 23.149.635 recovered out of 85.000.000

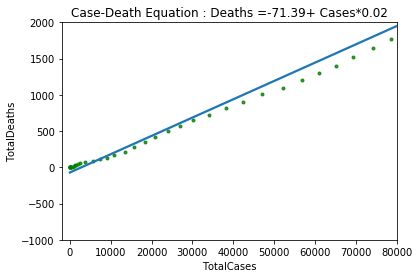
1. **Results and Discussions**

* **Linear Regression Model Result For TotalCases-TotalDeaths**



We made a model and virtualise relationship between totalCases and totalDeaths here.What do we get that is while total cases are increasing, total deaths are increasing also.It has linear and positive way relationship.

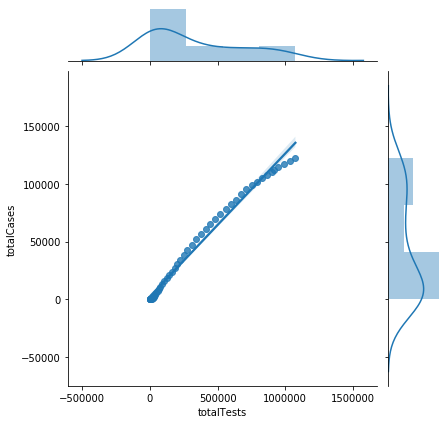
* **Linear Regression Prediction Result For TotalCases-TotalDeaths**



Our model result virtualise the prediction of total case and death relationship.In here our independent variable is TotalCases and dependent variable is TotalDeaths. When we calculate the model.predict function it gives much more near results of relation. Linear model tries to calculate almost no error .To calculate the prediction Linear Regression formule used (Yi=B0+B1\*X1) as a result, algorithm found coefficients coef\_ = 0.02524082 and intercept\_ = -71.39597159 then according the formula algorithm put the variables.

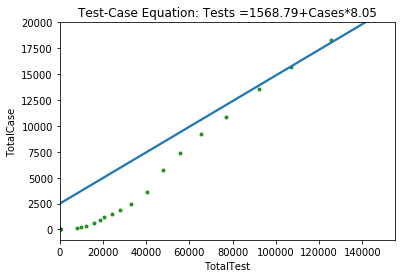
So formula was: Case-Death Equation : Deaths =-71.39+ Cases\*0.02

* **Linear Regression Model Result For TotalTests-TotalCases**



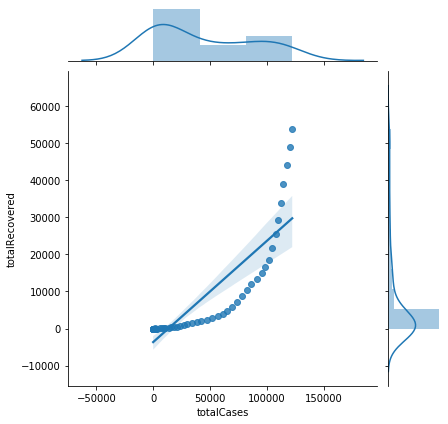
Second model that I made was for total case and total death relationship. In this model I put the dependent value totalCases and independent value totalTests and the result was like preceding.

* **Linear Regression Prediction Result For TotalTests-TotalCases**



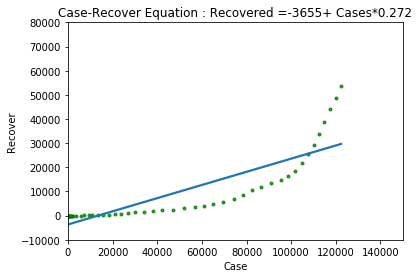
On prediction model Linear Regression algorithm calculated the values of coeficients as coef\_ = 0.12377894 and intercept\_ = 2496.517. According to the coefficients equation was Test-Case Equation: Tests =1568.79+Cases\*8.05. As you see through the tests that made cases were extremely increasing again on that prediction model .Then I calculated the model.predict([[85000000]]) result was 10.523.706 with mininum error condition.

* **Linear Regression Model Result For TotalCases-TotalRecovered**



Final model that I created was for totalCases and totalRecovered.In this model I put dependent value to totalRecovered and independent value to totalCases. Model result was like above.

* **Linear Regression Prediction Result For TotalCases-TotalRecovered**



On final model Linear Regression algorithm found coef\_= 0.27239165 and intercept\_ =-3655.0399 . Accourding the Linear Regression formula algorithm put the variables as Case-Recover Equation : Recovered =-3655+ Cases\*0.272

1. **References**

* (Covid Turkey Dataset) <https://github.com/ozanerturk/covid19-turkey-api/blob/master/dataset/timeline.csv>
* <https://towardsdatascience.com/a-beginners-guide-to-linear-regression-in-python-with-scikit-learn-83a8f7ae2b4f>
* <https://www.youtube.com/watch?v=E5RjzSK0fvY>
* <https://www.youtube.com/watch?v=wSyd6cvyDao>
* Machine Learning Slides