ALEXANDRU IOAN CUZA UNIVERSITY

FACULTY OF COMPUTER SCIENCE

BACHELOR'S THESIS

Machine Learning techniques for predicting COVID-19 cases

proposed by

Cătălin-Alexandru Sîrcu

Session: June, 2022

Supervisor

Conf. Dr. Răschip Mădălina

ALEXANDRU IOAN CUZA UNIVERSITY

FACULTY OF COMPUTER SCIENCE

Machine Learning techniques for predicting COVID-19 cases

Cătălin-Alexandru Sîrcu

Session: June, 2022

Supervisor

Conf. Dr. Răschip Mădălina

Declarație de consimțământ

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul **Machine Learning techniques for predicting COVID-19 cases** , codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test, etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea Alexandru Ioan Cuza, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Absolvent: **Cătălin-Alexandru Sîrcu**

Data: ..... Semnătura: .....

Contents

Motivation **2**

Introduction **4**

1 Related work **7**

2 Supervised Learning **10**

2.0.1 Over-fitting 11

2.0.2 Under-fitting 12

2.1 Support Vector Machines 14

2.2 Linear models 17

2.2.1 Linear Regression 18

2.2.2 Regularization 18

2.2.3 Polynomial regression 19

3 Deep Learning **21**

3.1 Long Short-Term-Memory 22

4 Data Analysis **26**

4.1 Metrics and Evaluation 37

4.2 Results of Support Vector Machines 38

4.3 Results of the polynomial regression 42

4.4 Results of Long short-term-memory 45

Conclusions **52**

Bibliography **54**

Motivation

Our lives are heavily involved technology. Now imagine going back in time to 1918, after the World War One we are struck by a global pandemic known as the Spanish Flu, where the access to the data of the confirmed, recovered, deaths is highly limited to an ordinary person that wants to see the weekly evolution of the cases and also is limited for a certain cities or countries.

Now in our modern times there are websites or applications that are tracking the cases and are updating day-by-day worldwide to see how the evolution of a virus is going to affect us in the near future. Data visualization was limited at that time, mostly because of the absence of modern day technology and also medical staff and statisticians were the only ones to be able to access this kind of data. After statisticians finished their work by tracking the cases day by day, applying formulas, and drawing the charts like in Figure 1, followed the information to be printed on the newspapers. Times have changed, and now, data scientist and analysts collect data and plot them in a variety of ways to illustrate them visually in many shapes: bar charts, line plots, scatter plots, histograms, pie charts, heat maps, stacked column charts etc. shown in Figure 2, which offer information to be easily understandable for a given topic.

As people we tend to check these websites or applications every day to see how the virus is performing by viewing graphs and statistic models, but something that was not available at that time were predictions made by machine learning models. Having access to them we can choose from the many machine learning algorithm that resembles our data set specifications in order to make a prediction and to visualize the results to see if the situation will settle down in the future or it will get worse.

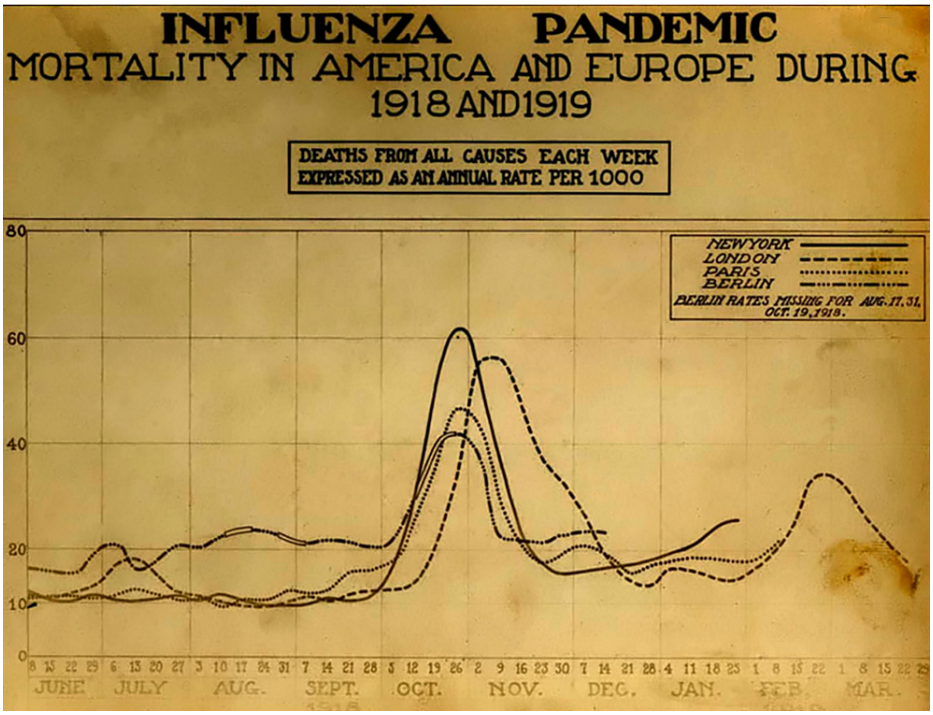


Figure 1: Newspaper depicting a chart of deaths from major cities, showing a peak in October and November 1918. Source: Wikipedia.

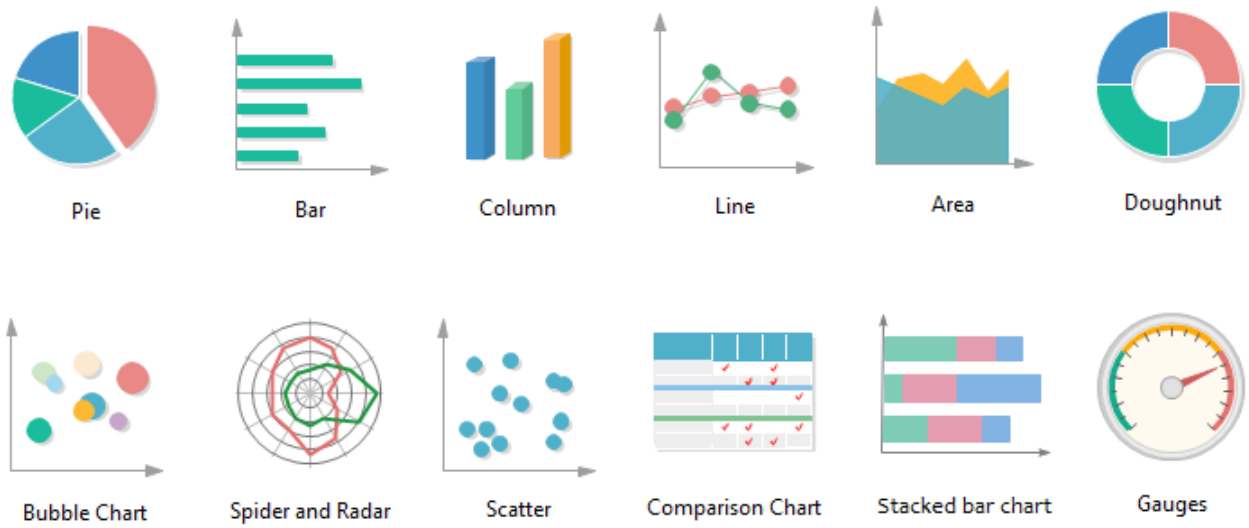


Figure 2: Types of charts.

Introduction

SARS-CoV-2 or commonly referred as COVID-19 is a contagious disease from a family of viruses named Coronaviruses (CoV). The first known case was identified in December 2019 in the Hubei province of China in the city of Wuhan, being home to 12.3 million people. China being a densely populated country, the virus started to spread as quickly as it could. This is considered to be the fifth pandemic in the history of mankind, the fourth one was in 1918 commonly referred as *"The Spanish Flu"*.

The COVID-19 pandemic changed and affected many aspects of our day to day lives. Working has been affected by not going for a period of time to the jobs and working remotely which changed the way we can manage our work; education was affected by doing it in an online manner, with professors which hardly adapted to this style of teaching and poor families from the rural side of the country which couldn't afford a laptop or a smartphone for their kids, which they were highly dependent of going to the school; access to medical services was hard, because the doctors were mostly concentrating on the patients suffering from this infectious disease, other patients with other medical affections have been the bottom priority; in terms of economy, the economy decreased and many people lost their jobs due to the pandemic.

Also there was a research and scientific race for the development of a cure as soon as possible. To bring the end of a pandemic, a large part of the population has to get vaccinated in order to minimize the spreading of the virus and to lower the death rate. By minimizing the spreading, there is a low chance that a virus might evolve in another variant. A variant is a branch of the virus and comes with at least one new change to the original virus which can be more deadlier or infectious.

I would like to mention in this thesis the novel entitled *"The Plague"* written by french philosopher Albert Camus. I realized that what he wrote in this novel, were strangely similar to the pandemic of COVID-19 from start to end. The author affirms that throughout the history there have been many plagues as wars, and yet both take

people by surprise. In the novel the action takes place in an Algerian city called Oran which is mysteriously hit by an epidemic. When the cases begin, a hysteria occur on the local newspapers, the protagonist tries to alert the local authorities regarding his theory, but is denied. Once the death tolls rise quickly, the local authorities and the Prefect hardly accept the situation and take weak measures. When the daily death toll reaches 30, the city is closed and is officially declared a plague outbreak. The city now is in quarantine and the only way to communicate between human beings is the use of short telegrams. The isolation affects the population day-by-day making them depressed. The situation continues to worsen on the summer, and many people try to escape the quarantined city, but they are shot by sentinels. In the end, in winter the epidemic is on the verge of ending, and the people are celebrating the opening of gates. Many aspects of the novel Albert Camus wrote in 1947 are similar with the COVID-19 pandemic, from the way authorities manage to handle the situation, to the impact that it had on its people.

Machine learning focuses on extracting the knowledge from data. It represents a vast and large research field, being the crossroad between statistics, computer science and artificial intelligence. In our times, the applications of machine learning are present in our day-to-day life in many fields such as: digital media and entertainment, medicine, automotive industry, scientific research, finances and many more domains of expertise. Applications in the digital media and entertainment field could be: face and fingerprint recognition, ranking the documents made by the search engines, recommending food and movies. In automotive industry we can find: self-driving cars, collision detection. When you are browsing complex websites such as: Netflix, Google, Amazon and Facebook, it is likely that we have encountered machine learning models.

The problem approached in this thesis is a regression problem. Regression is a statistical method that models the relationship between dependent or target variables and independent or predictor variables with one or more independent variables. It helps us in understanding how the value of the target variable is changing corresponding to a predictor variable. So the goal for regression is to predict a continuous number.

Considering that COVID-19 cases have labels such as: the number of confirmed, death, recovered and active cases, they need a regression model in order to be predicted for a certain number of days.

The most well known machine learning algorithms for a regression problem are:

the Linear Regression family which consists of Lasso and Ridge Regressions, Polynomial Regression, Support Vector Regression, Decision Tree Regression, Random Forest Regression.

Chapter 1

Related work

Being aware that there are many studies on this field of COVID-19, mostly because it's a recent event that occurred, I've been researching scientific articles to see what types of learning have been used, what type of problems have been approached and how well they performed by their scores on their data set.

First we'll take a look at the research entitled *"Artificial intelligence and machine learning for COVID-19"* in which the study has been made on a COVID-19 data set of India to find out the association of various independent variables such as 'State/Union Territory', 'date', 'month', 'cured' and 'death' with dependent variable 'confirmed'. The performance of the algorithms was measured using the coefficient of determination metric.

Ensemble learners provide more accurate results than single learner by building many learners and merging them for better accuracy score. The ensemble learners used for this study and their scores are as it follows: gradient-boosting regressor with a result of 0.99, extra-trees regressor which got a score of 1.0, random-forest regressor with a result of 0.99 and last but not least ada-boost regressor with a score of 0.93. The author concludes that: *doing this analytical study on India of COVID-19 outbreak shows that the cases are spreading rapidly, despite all the various phases of lockdown, thus showing the need for the Indian Government to take drastically measures in order to have the situation under control*.[2]

In another article entitled *"Supervised Machine Learning Models for Prediction of COVID-19 Infection using Epidemiology Dataset"* there is a different approach because

it's a classification problem. The data set used for this study contains positive and negative COVID-19 cases from Mexico having 263,007 instances and 41 features in which the next features were selected: two demographic features including age and gender and eight clinical features which include *pneumonia, diabetes, asthma, hypertension, cardiovascular diseases, obesity, chronic kidney disease and one high-risk factor which is tobacco and RT-PCR*.[3] The author applied algorithms such as: decision trees, logistic regression, naive Bayes, support vector machines and artificial neural networks. In terms of performance their accuracy scores are as it follows: decision trees obtained the best score of all the algorithms used which is 95% accuracy, followed by logistic regression with a score of 94%, naive Bayes with 94%, support vector machines with 92% and at the bottom the artificial neural networks with 89%.

Decision trees pointed out that the age feature is the main factor, being more important than the clinical features. *The model indicated that most of the people above the age of 45 years are more likely to be infected when compared to people with a lower age*.[3] Regarding gender, males are more vulnerable to COVID-19 infection than females, and smokers are in danger of being infected easily by the virus compared with a non-smoker person.

Last article that I am going to cover it briefly for the scientific research that has been done in this field is entitled *"Outbreak Prediction of COVID-19 for Dense and Populated Countries Using Machine Learning"*. The researcher considered the top 10 countries with high population and high density for the outbreak prediction. The data set of the countries, namely, Bangladesh, India, China, Pakistan, Germany, Nigeria, Ethiopia, Democratic Republic of Congo, the Philippines, and Indonesia have been used. The percentage of test data was 6%, 94% for the training data respectively. The algorithms predicted the rise in the number of cases in the next 5 days. Some algorithms that the researcher stands out with are: ARMA (Auto-regressive Moving Average), ARIMA (Auto Regressive Integrated Moving Average) and XGBoost Regressor.

The ARMA model is the combination between Auto Regressive (AR) and Moving Average (MA) models. The AR model, tries to explain the momentum and mean reversing effects often observed in trading markets and the MA model, which tries to capture the shock effects observed in thermal noise.[6]

ARIMA is a predictive model that predicts future time series based on its past values. An ARIMA model is represented by 3 terms: p, d, and q, where p is the order of the auto regressive term, q is the order of the moving average term and d is the number of differences required.[6]

ARIMA model was applied on a training data set and the values of p, q, and d changed depending on what the model fitted the best. *Values of p and q are normally taken up to 6 and d varied between 0 and 1. The values of p, d, and q varied for different training data sets depending on the best fit.*[6]

Extreme Gradient Boosting is commonly referred as XGBoost and is an upgrade of gradient boosting trees algorithm. *The researcher set the estimators to 1000 and fitted the model. This model is characterized by a diversified mix of training losses and regularization measures.*[6] The results of these three algorithms are as it follows: the highest accuracy achieved was with ARMA on Ethiopia data set with a whopping score of 100%. ARIMA gave an accuracy of 85% most of the time on all countries and XGBoost had a 82% accuracy on the China data set.

Chapter 2

Supervised Learning

Machine learning comes into multiple types of algorithms, but the most well known of them are: Supervised Learning and Unsupervised Learning.

Supervised Learning comes into two main shapes and these are: classification and regression. Classification is the process where computers group data together based on predetermined characteristics and it comes into 2 main categories: binary and multiple classification. Binary Classification classifies data into two classes such as Yes/No, good/bad, high/low. In the medical field determining whether a tumor is benign or malign based on images, for this types of problems the programmer requires a database populated with medical images on this issue, also there is needed a professional opinion, so the doctor has to look at all images to see which tumors are benign/malign or not.

Multiple Classification classifies data into three or even more classes, some examples could be classifying a document, malware or product.

Unsupervised learning represents the second family of machine learning algorithms; they are characterized by the fact that they are given only the input data and no output data is given to the algorithm. Mostly there are many algorithms that offer satisfactory results, but they come with the cost that they are harder to evaluate and understand. The main algorithms corresponding to this family are: k-means clustering, hierarchical clustering, anomaly detection and PCA (Principal Component Analysis).

Clustering task is to partition the data set into groups that are called clusters. Mainly clustering algorithms predict a number to each data point, to indicate which point belongs to a given cluster.

2.0.1 Over-fitting

The main goal in Supervised Learning is to construct a model on the training data that we give it, and to be able to make accurate predictions on unseen data that has similar properties as the training data. If a model is able to accurate predictions on unseen data we can conclude that it was able to generalize from the two sets (training and test).

However if the training and test set have enough in common, our expectation is that the model will be able to generalize, right? Not really, there are cases where this does not go according to the plan and this happens when we want to construct very complex models, which in many situations, on the training set will have a high percentage scoring. The only way to determine the performance of an algorithm on unseen data, is to evaluate the test set on classification computing different metrics like F1 score and R^2 for regression problems.

Over-fitting in machine learning is the phenomenon in which the model performs better on the training set, but performs poorly on the test set (unseen data). It occurs when the model takes too long to train and the presence of ”noises” in the training set leads to negative consequences on the performance on unseen data, the reason being that if a model is exposed to noises, it will become more complex.

Data ”noises” in machine learning causes problems, since the algorithm interprets the ”noises” as a pattern in the data set and it tries to generalize it. They appear from mistakes made by individuals in collecting data from forms, excels, databases etc., they are characterized by missing values or extra features. A real life scenario of over-fitting could be as follows: a high school student is preparing for the Baccalaureate exam at literature, but he is learning topics that are not included in the syllabus. Over-fitting generally appears on simple models, such as Linear Regression and Decision Trees.
Some methods for solving this issues could be:

- using K-fold cross validation
- using regularization techniques
- using ensemble techniques
- training the model with sufficient data

2.0.2 Under-fitting

Under-fitting is the case in machine learning and data science in which the model is unable to catch the relationship between the input and output variables precisely, thus generating a high error rate on both the training and unseen data. Under-fitting happens usually when a model is too simple, thus needing more time to train, less regularization and more input features.

A real life example of under-fitting could be like the above one discussed on over-fitting, for example: a high school students prepares for the Baccalaureate exam at literature, and he learns one to two poems; in order to succeed the exam, he needs more materials from the syllabus.
Methods for solving under-fitting issues could be:

- more training data for the model
- pre-processing the data to reduce noise
- increasing the model complexity
- less regularization

To summarize all these three concepts we have to remember:

- For an over-fitting model we have high percentage of accuracy score on the training set and low accuracy score on test set. An over-fitting model could be represented as in the image down below Figure 2.1:

$$Trainingset \uparrow Testset \downarrow$$

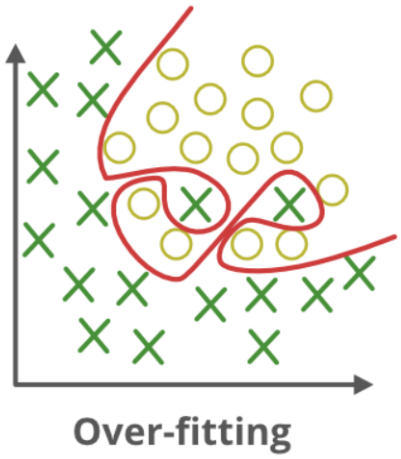


Figure 2.1: A demonstration on depicting over-fitting. Source: GeeksForGeeks.

- For an under-fitting model we have low percentage of accuracy score on both sets. An under-fitting model could be represented as in the image down below Figure 2.2:

$$Trainingset \downarrow Testset \downarrow$$

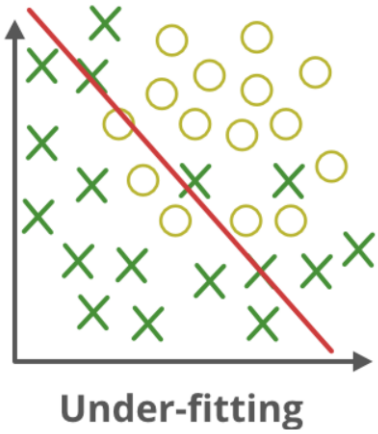


Figure 2.2: A demonstration on depicting under-fitting. Source: GeeksForGeeks.

- For a model to fit the data, we have on both sets high accuracy scores. A generalized model could look something like in the image down below Figure 2.3:

$$Trainingset \uparrow Testset \uparrow$$

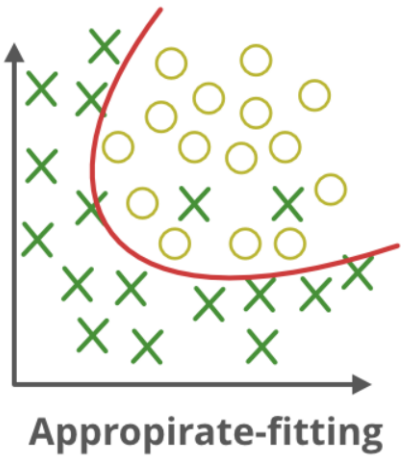


Figure 2.3: A demonstration on depicting a generalized model. Source: GeeksForGeeks.

2.1 Support Vector Machines

Support Vector Machines or commonly referred to SVM are a powerful and flexible class using classification and regression. For classification they are referred to Support Vector Classification or simply SVC and for Regression: Support Vector Regressor or SVR.

In the article entitled *"Analysis on Novel Coronavirus (COVID-19) Using Machine Learning Methods"* this information about how SVRs work could be found: *SVR has the same principles and concepts as SVC, an example being the searching for a hyperplane in a d-dimensional space that uniquely classifies the data points. SVR makes use of a non-parametric technique, thus the output from SVR model does not depend on the distributions between the target and predictor variables. SVR technique is essentially determined by the kernel functions which allow the construction of a non-linear model without changing the predictor variables, thus helping in better interpretation of the resultant model. The model produced by the SVM does not depend on the training set points that are outside the margins, but instead they are dependent on a subset of the training data as the cost function. Similarly, in SVR, support vectors find the closest data points and the actual function is represented by them.* [4]

Hyperplane is a function that classifies the points in a higher or lower dimension, in other words a hyperplane is a boundary. If the margin for any hyperplane is maximum, then that hyperplane is the optimal hyperplane. The points which are closest to hyperplane are called support vector points and the distance of the vectors from the hyperplane are called the margins as shown in the Figure 2.4:

Hyperplane's formula in d-dimensions can be given as:

$$z = l_0 + l_1 x_1 + l_2 x_2 + l_3 x_3 + \cdots + l_n x_n$$

$$z = l_0 + \sum_{i=1}^{n} l_i x_i$$

$$z = l_0 + l_1^T x$$

$$z = b + l_1^T x$$

where $l_i = l_0, l_1, \cdots, l_n$*, b or* l_0 *is the biased term and x are variables.* Kernel is a key component to the SVR, because it computes the dot product of two given vectors x and y in a high dimensional feature space. The kernel trick is maneuver that helps Support Vector Machines to work with non-linear data.

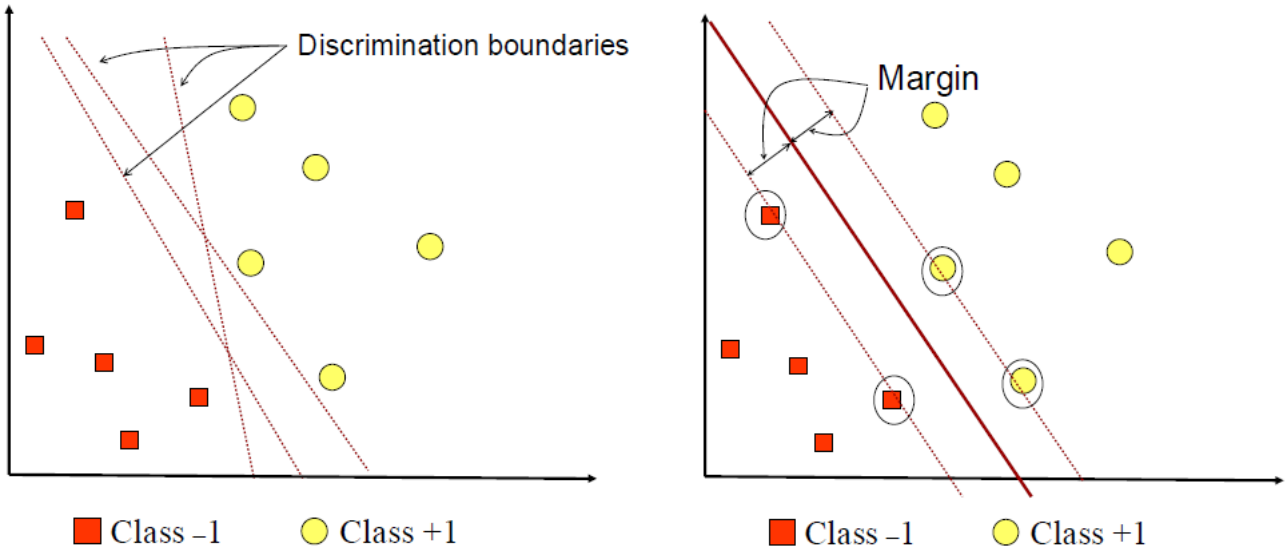


Figure 2.4: SVM Model Maximum-margin Hyperplane. Source: Medium.

Linear SVM is used for low dimensional spaces where data points can be linearly separated. Furthermore the hyperplane is obtained by measuring the distance from it and the closest point in each class like in Figure2.5.

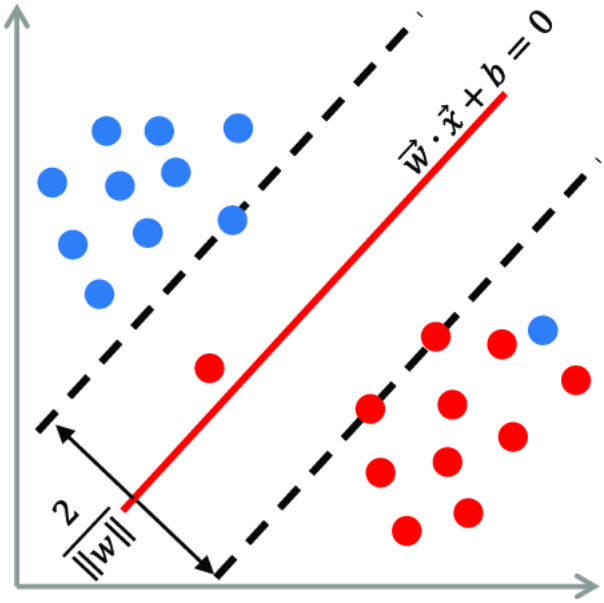


Figure 2.5: Linear Support Vector Machine. Source: ResearchGate.

Non-linear SVM is used for high dimensional (more than two features) spaces where data points can't be linearly separated. For this kind of cases the functions of the kernel trick are applied in order to get the job done. The accuracy of a non-linear SVM model is highly dependent on the choice of the kernel function and the parameters

used for tuning. A visualization of a non-linear SVM could be seen in the image down below in Figure 2.6:

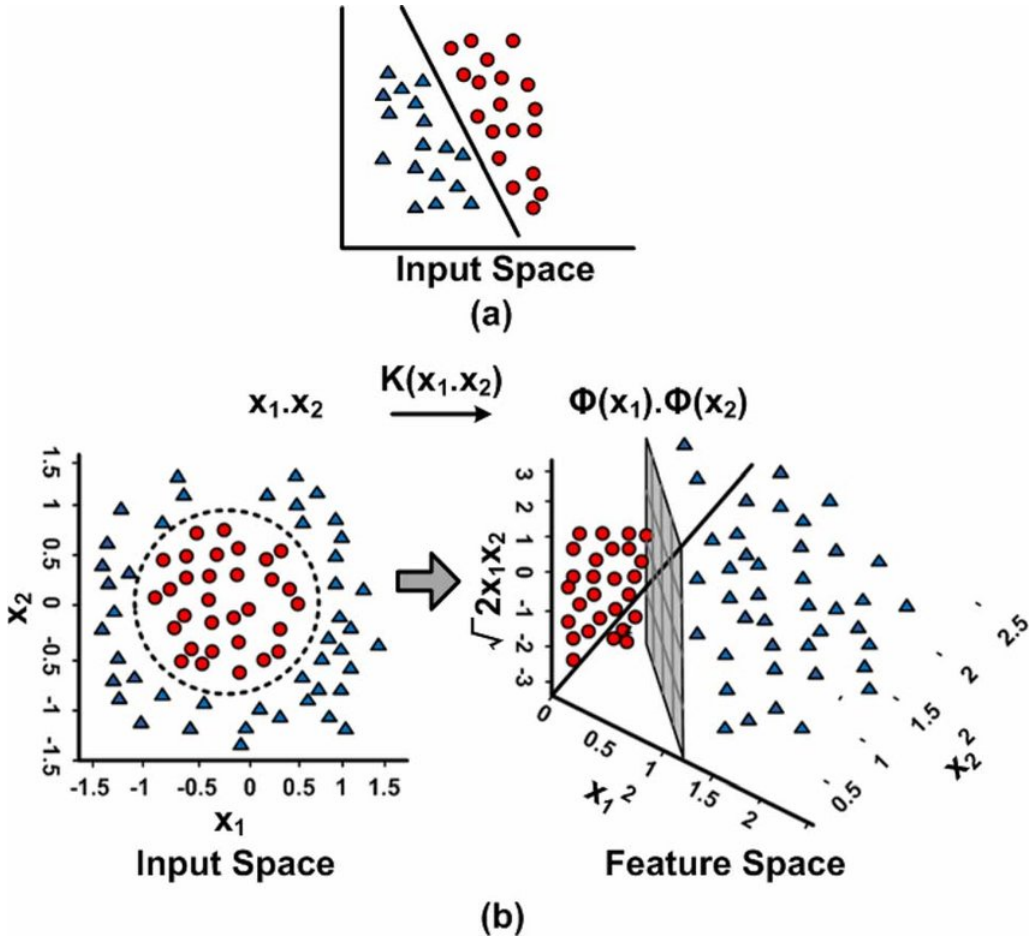


Figure 2.6: Non-linear Support Vector Machine. Source: ResearchGate.

Here are the types of kernel functions for the non-linear Support Vector Machine models:

- Radial-Basis Function Kernel (RBF Kernel)

  The RBF kernel is the most widely used kernel concept, because it is highly performant and offers great results in both classification and regression problems. The equation formula for the RBF kernel function is:

  $$k_{\text{rbf}}(x, x') = \exp\left(-\gamma \left\|x - x'\right\|^2\right)$$

  Here, x and x' are data points, $\|x - x'\|$ denotes the Euclidean distance, and γ is a parameter that controls the width of the kernel.

- Polynomial Kernel

  Represents an upgraded form of the regular linear kernel. The equation of the

polynomial kernel is as denoted down below:

$$k_{\text{poly}}(x, x') = (\gamma \langle x, x' \rangle + r)^d$$

Polynomial kernel it is as used as the RBF kernel and the *main task is to separate non-linear data. It contains a d parameter being the degree of the polynomial, the default value for the parameter is $d = 2$, the bigger the degree the more the accuracy will fluctuate and will become less stable.*[5]

- Sigmoid Kernel
  It originates from neural networks, and is equivalent to a two layer perceptron model and they are not usually used in SVMs, *having a complex structure and being difficult to interpret and understand how it makes it's decisions.*[5]
  The formula for the sigmoid kernel function is:

$$k_{\text{sigm}}(x, x') = \tanh(\gamma \langle x, x' \rangle + r)$$

- Linear kernel
  The formula for the linear kernel is:

$$k_{\text{linear}}(x, x') = \langle x, x' \rangle$$

2.2 Linear models

In the book entitled "Introduction to Machine Learning - A Guide for Data Scientists" there can be found a description about the linear models as: *Linear models are a class of models that are widely used in practice and dominate the field of data science and have been studied extensively over the last decades. Linear models make a prediction using a linear function of the input features.*[1]
Down below, there is the general formula for prediction for a linear model:

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \ldots + w[p] * x[p] + b$$

Here $x[0]$ to $x[p]$ denotes the features (in this example the number of features is equal $p+1$), w and b are parameters of the model that are learned, and $\hat{y}$ is the prediction the model. For a data set of a single feature, the formula looks like this:

$$\hat{y} = w[0] * x[0] + b$$

which is the equation for a line.
W parameters are called weights or coefficients or slope and b is the y-axis offset or intercept.

There are a lot of model types for regression, which we'll cover in this chapter. The difference between these models of linear regression lies in how the the parameters w and b are learned from the training data, and how model complexity is controlled.

2.2.1 Linear Regression

This model is the simplest and the most classic form of regression. Linear regression or sometimes called "ordinary least squares" finds the parameters w and b that minimize the mean squared error between predictions and true regression targets, y, on the training set. A benefit of this form of regression is that it requires no parameters to tune, but the downside is that there is no way you can control the model's complexity.

2.2.2 Regularization

Regularization is one of the many important concepts of machine learning. It represents a method for counter-attacking the over-fitting problem by adding a constraint to the complex model.
The most common regularization techniques for linear regression are: Ridge and Lasso regressions.

Ridge

The first technique is ridge regression, in which the equation is the same as linear regression, but the coefficients are chosen not only so that they predict well on the training data, but also to fit an additional constraint. The entries of the coefficients should be close to zero, this means each feature should have as little effect on the outcome as possible (having a small slope), while still predicting well. Ridge regression is often called L2 regularization.

A less complex model means worse performance on the training, but better generalization of the model. Ridge model makes an exchange between the simplicity of the model (near-zero

coefficients) and its performance on the training set.[1]

Lasso

The second technique for regularization is the Lasso regression, also called L1 regularization. The difference between Ridge and Lasso is that in Lasso some coefficients are exactly zero, while at Ridge there are coefficients at near-zero.

2.2.3 Polynomial regression

Simple linear regression assumes that the relationship between the independent and dependent variables is linear, but in most cases it happens that this relationship between dependent and independent variables to be non-linear, thus resulting a poorly fit model. A way of counter-attacking this kind of problem is to use polynomial regression which is an upgraded linear model, where p is the degree of the polynomial equation down below:

$$\hat{y} = w\left[0\right] * x\left[0\right] + w\left[1\right] * x\left[1\right] + w\left[2\right] * x\left[2\right]^2 + \ldots + w\left[p\right] * x\left[p\right]^p + b$$

When the value of the degree is increased, the model is able to fit non-linear data, but in practice the degree is chosen to be 2, 3 or 4. If the degree goes past 5 or higher, there is a high chance the model will become too complex and over-fitting will occur, resulting in a lower value of R^2 on the test set. As I mentioned above even though polynomial regression is able to fit non-linear data, is considered to be a a upgraded form of linear regression, because it is linear in his coefficients: $w[0], w[1], \cdots, w[p]$.

There are two types of reducible errors in machine learning which are bias and variance, that can be reduced in order to improve the model accuracy and there exists *a bias-variance trade-off when utilizing polynomial regression, if the bias decreases the variance rises which means the model becomes complex and over-fits, if the bias increases and the variance is decreasing the model is simpler and tends to under-fit.*[10]

Our task is to find an optimal trade-off between these two, meaning that the model has to generalize like in the Figure 2.7:

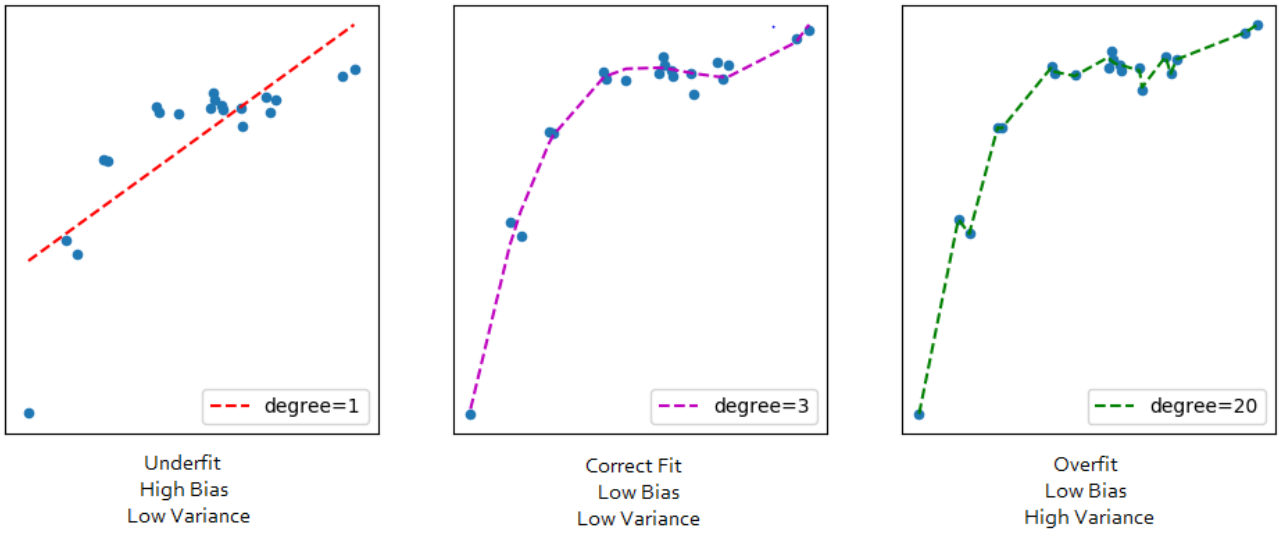


Figure 2.7: Polynomial Regression with types of variance-bias relationships. Source: Medium.

In some cases it helps to iterate through the degrees of the polynomial, but to a certain point, because the model will over-fit. To ensure that the model is fitted k-fold cross validation is a method to go with.

Chapter 3

Deep Learning

Deep learning is a subset of machine learning, and is based on artificial neural networks which they were inspired by the biological structure of the neurons in the brain. Deep learning incorporates algorithms such as recurrent neural networks and convolutional neural networks and they are used for data representation.

In the Figure 3.1 there is a comparison between a biological neuron and an artificial neural network.

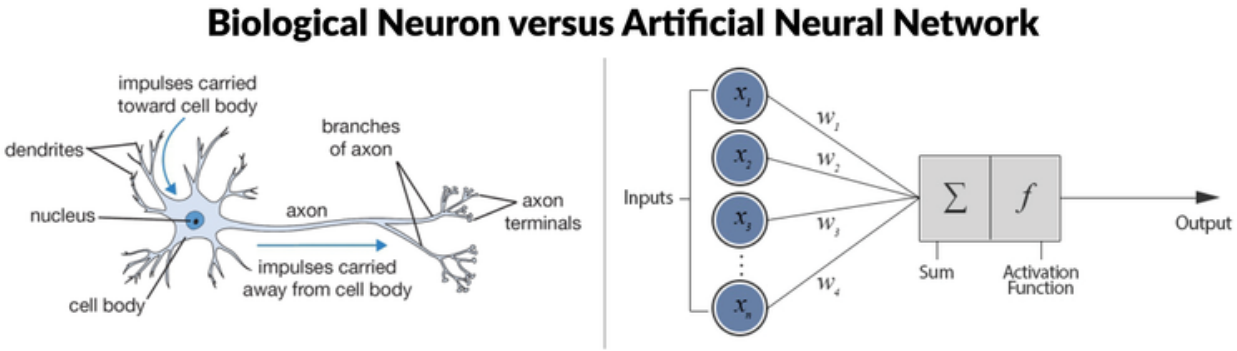


Figure 3.1: The structure of a biological neuron compared to a neural network. Source: ResearchGate.

Deep learning is present in many types of applications showing great promise. Often these kind of algorithms are carefully tailored in order to obtain great results. *They manage to improve automation, perform analytical and physical tasks without a human to be involved. Their technology lies in everyday products such as digital assistants, voice-enabled TV remotes and card fraud detection.*[8]

Recurrent neural networks for short "RNN" are one of the many neural networks algo-

rithms in Deep learning which uses sequential data or time series. Their use is involved in problems such as: natural language processing (nlp), speech recognition, and the most famous application in which they are used is Google Translate. Recurrent neural networks use training data, same as the convolutional neural networks, but they are characterized by their "memory" as they take information from the prior inputs to influence the current input and output. While traditional neural networks suppose that the inputs and the outputs are free of one and another, the output of a RNN depends on the prior elements within the sequence.[7]

Here in Figure3.2 we can see a simple architecture of the recurrent neural network in which it takes input from the previous step and current input. Here in the image $\tanh$ is the activation function, but it also could be ReLu or sigmoid.

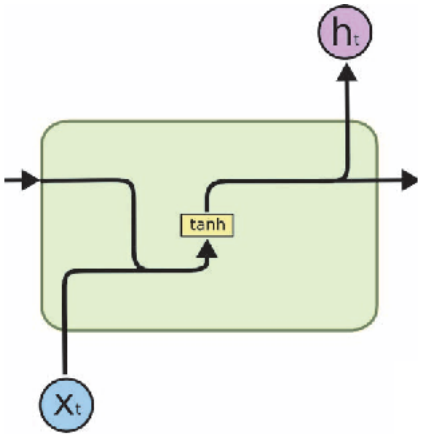


Figure 3.2: A recurrent neural network cell. Source: Medium.

Recurrent neural networks has it's pros and cons like any other algorithm in general, a main con of this kind of neural architecture is the short-term memory problem characterized by the vanishing gradient problem. In the vanishing gradient problem, the gradients of the loss function reach values close to 0 and they become harder to train. A way to combat this kind of problem is to use Long Short-Term-Memory for short "LSTM" which are an extension of recurrent neural networks, introduced for handling situations where RNN fails.

3.1 Long Short-Term-Memory

LSTM as I mentioned above are an extension of RNN, a sequential or time series, that allows information to persist. In short LSTMs extends the "memory" of RNN, and fights against the vanishing gradient problem, which is where the neural network stops

learning because the updates to the various weights within a given neural network become smaller and smaller.

In the article entitled "Long Short-Term Memory Neural Networks for Online Disturbance Detection in Satellite Image Time Series" is stated that: *Recurrent neural networks come with an improvement compared to a regular neural networks which that the hidden layers are treated as successive recurrent layers. This kind of construction is unfolded to produce the results of a neuron component sequence at discrete time steps corresponding to the input time series shown in the Figure 3.3:*[9]

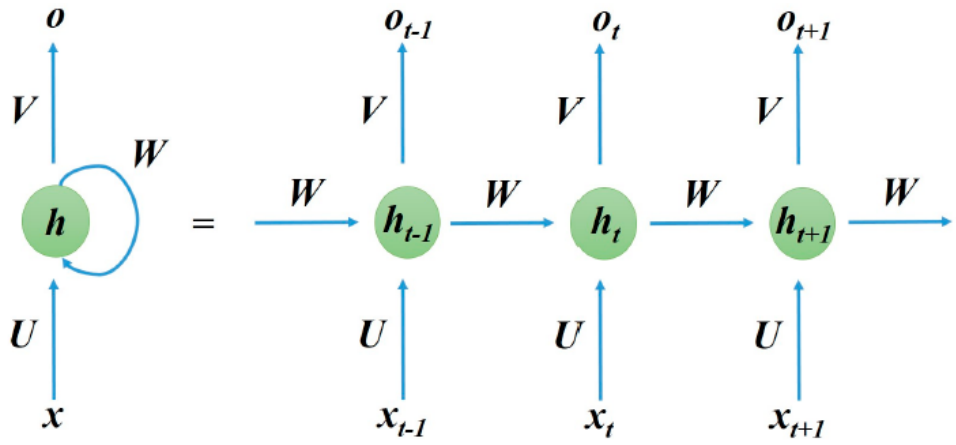


Figure 3.3: A recurrent neural network unfolded structure. Source: [9].

Mathematically the process of carrying memory forward can be described using this formulas:

$$h_t = \sigma(Wx_t + Uh_{t-1} + b^{(h)})$$

$$o_t = \sigma(Vh_t + b^{(o)})$$

x_t is an input vector, h_{t-1} represents the previous hidden state, and h_t is the hidden state for x_t, o_t is the output vector, and W, U, V are the weight matrices, σ is the activation function and b is the bias.[9]

Long short-term memory allows the network to capture information from the inputs for a longer period of time by using a hidden unit, called the LSTM cell, compared to a normal RNN unit in the hidden layer. The structure of a LSTM cell is shown in the Figure 3.4 down below:

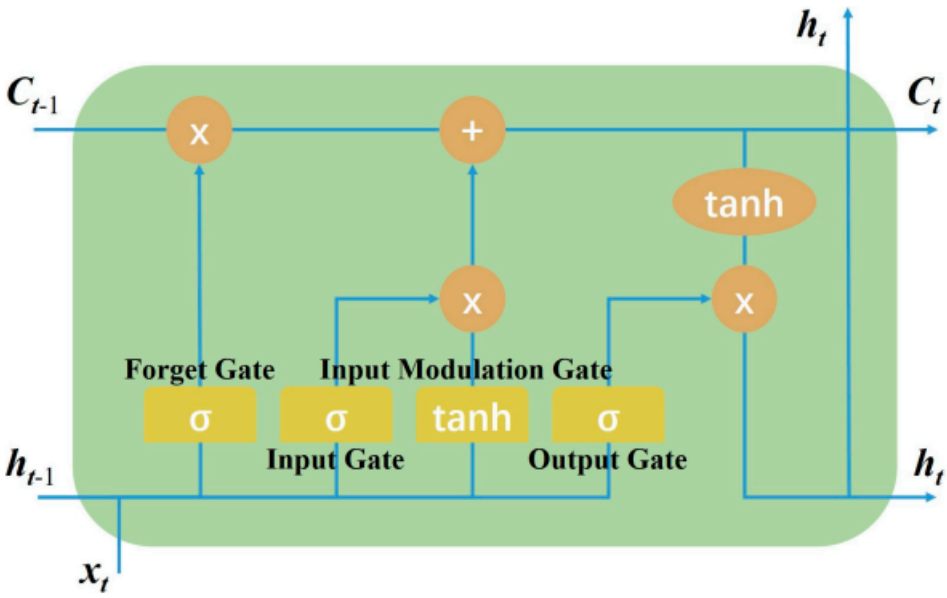


Figure 3.4: A LSTM cell. Source: [9].

The structure contains as it follows: the forget gate (f_t)*, the input gate* (i_t)*, the input modulation gate* (m_t)*, the output gate* (o_t)*, the memory cell* (c_t) *and the hidden state* (h_t)*. The gates are computed by:*

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)})$$

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)})$$

$$m_t = \tanh(W^{(m)}x_t + U^{(m)}h_{t-1} + b^{(m)})$$

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)})$$

where x_t *is an input vector,* h_{t-1} *is the previous hidden state in LSTM network,* W *and* U *are the weight matrices,* σ *is the logistic sigmoid function,* $\tanh$ *is the hyperbolic tangent function and* b *is the bias vector. From these gates, the memory cell* (c_t) *and hidden state* (h_t) *are calculated as:*

$$c_t = i_t \cdot m_t + f_t \cdot c_{t-1}$$

$$h_t = o_t \cdot \tanh(c_t)$$

where "$\cdot$*" is the multiplication between two vectors. The memory cell* (c_t) *contains the information of the previous memory cell* (c_{t-1}) *modulated by the forget gate* (f_t)*, the information of the current input* (x_i) *and previous hidden state* (h_{t-1}) *modulated by the in input modulation gate* (m_t)*. The forget gate* (f_t) *allows the LSTM to selectively forgets its previous memory* (c_{t-1}) *The input gate* (i_t) *controls the LSTM to consider the current input. The input modulation*

gate (m_t) *modulates the information of the input gate* (i_t)*. The output gate* (o_t) *controls the memory cell* (c_t) *to transfer to the hidden state* (h_t).[9]

Chapter 4

Data Analysis

Data analysis is an important concept to take in consideration, before working your way up with machine learning. Is a process that is involving the inspection, cleaning and transforming of the data collected by the researcher with the ultimate goal of finding useful information. Data analysis has numerous approaches and techniques and is commonly used in domains like: finance, business and science.

After collecting the data you are interested in analyzing it, it's mandatory for the data to go through a process called "preprocessing". Preprocessing is an important step, because after collecting the raw data, next have to be applied methods to transform it in a format in such way that can be understood by the computer in order to perform machine learning task and data visualizations. Raw data in general is a mess, in most cases it contains errors and inconsistencies which lead to "noises" when applying a machine learning model, meaning that if the environment of the data is not cleaned, the model will perform poorly on the data set leading to bad results. According to data scientists, the task that is usually performed the most in their day to day job is data cleaning. Data cleaning is a process of repairing, adding missing data and deleting irrelevant data. By doing that, you ensure that your data set is ready for analysis. The main problems are missing data and noisy data. Missing data is a dull process, and usually you have to fill manually. Noisy data is characterized by useless data or hard to group, in case of wanting to perform a clustering method.

The environment used for data analysis and science is Jupyter Notebooks and is an open source web-based application for sharing documents that contain live code, visualization an texts. Next for extracting data and manipulating data through preprocessing, is mandatory the use of the Pandas library.

The data sets used for this thesis project were: one for the confirmed cases, one for recovered cases, and one for death toll cases of the countries around the globe which were found on the *Kaggle* platform. Kaggle is a platform filled with machine learning and data science projects and data sets. The data set contains the corresponding values from their category and 498 columns consisting of: country, region, latitude, longitude and dates from 22/01/2020 to 29/05/2021. All the three data sets were merged in a single data frame.

My study focuses on the analysis of the European continent and its subregions and the South American continent. As we know, Europe was powerfully struck by the pandemic of COVID-19, Italy being one of the most affected European countries. The safest countries in Europe were by far the island nations such as: Cyprus, Malta and Iceland. The subregions of the European continent are: eastern, northern, western, southern, southeastern and central Europe. The next figures 4.1, 4.2, 4.3, 4.4,4.5, 4.6 show the confirmed, death, recovered and active cases of every subregion of Europe.

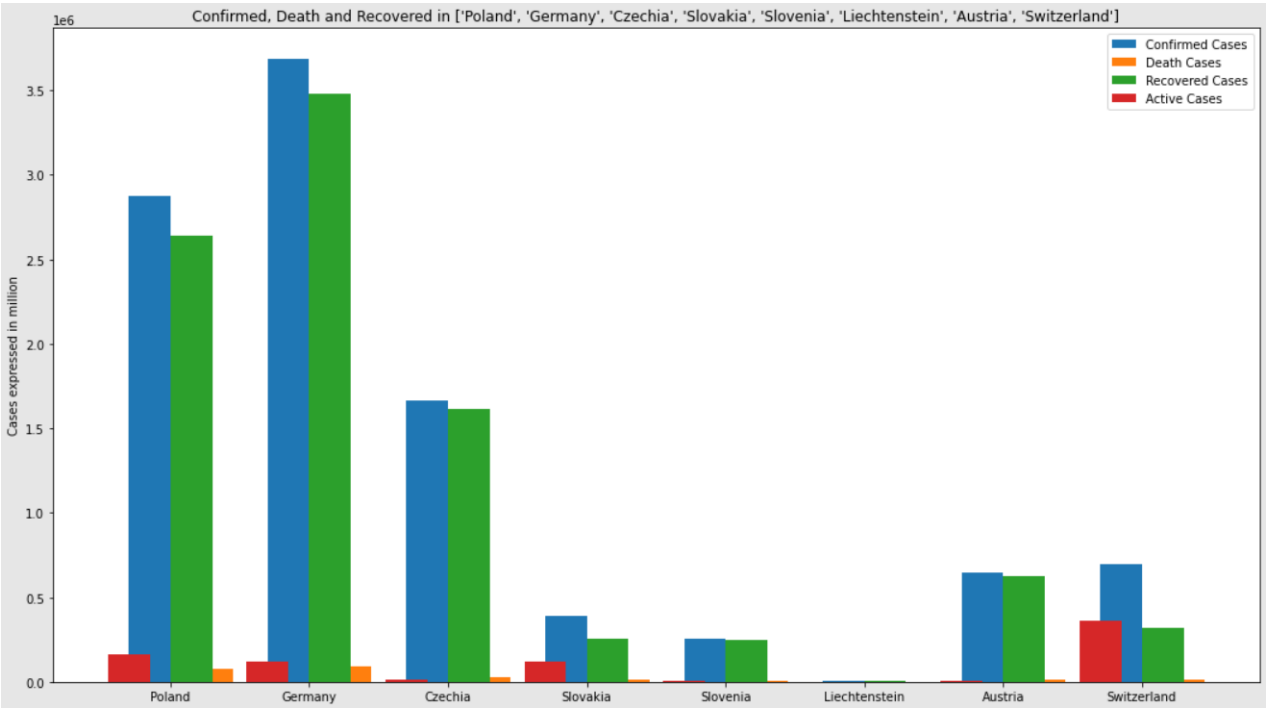


Figure 4.1: Central Europe cases.

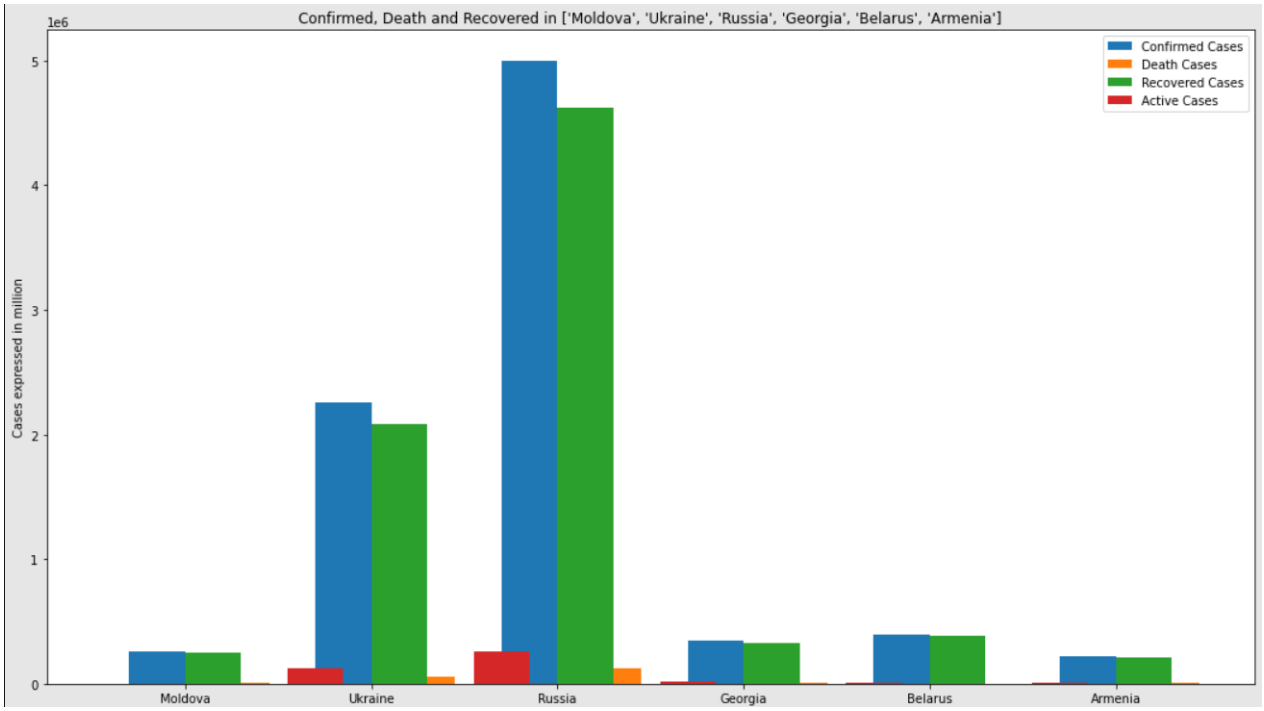


Figure 4.2: Eastern Europe cases.

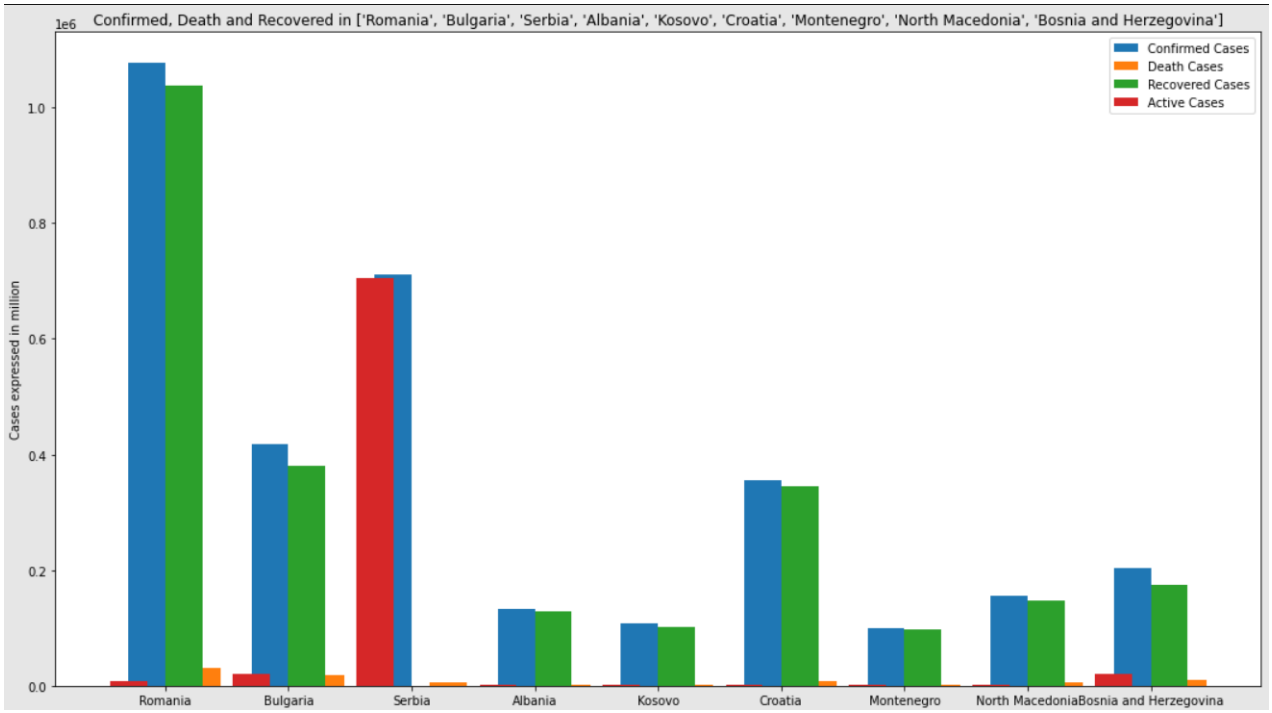


Figure 4.3: South-Eastern Europe cases.

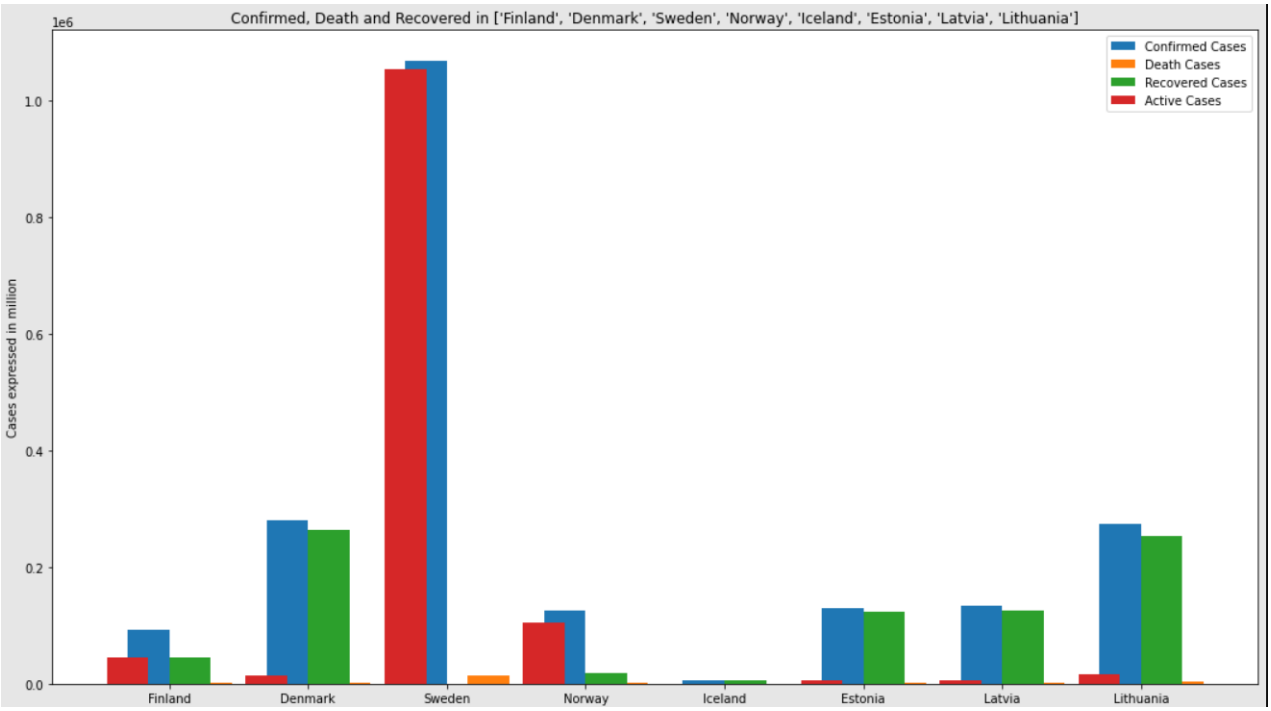


Figure 4.4: Northern Europe cases.

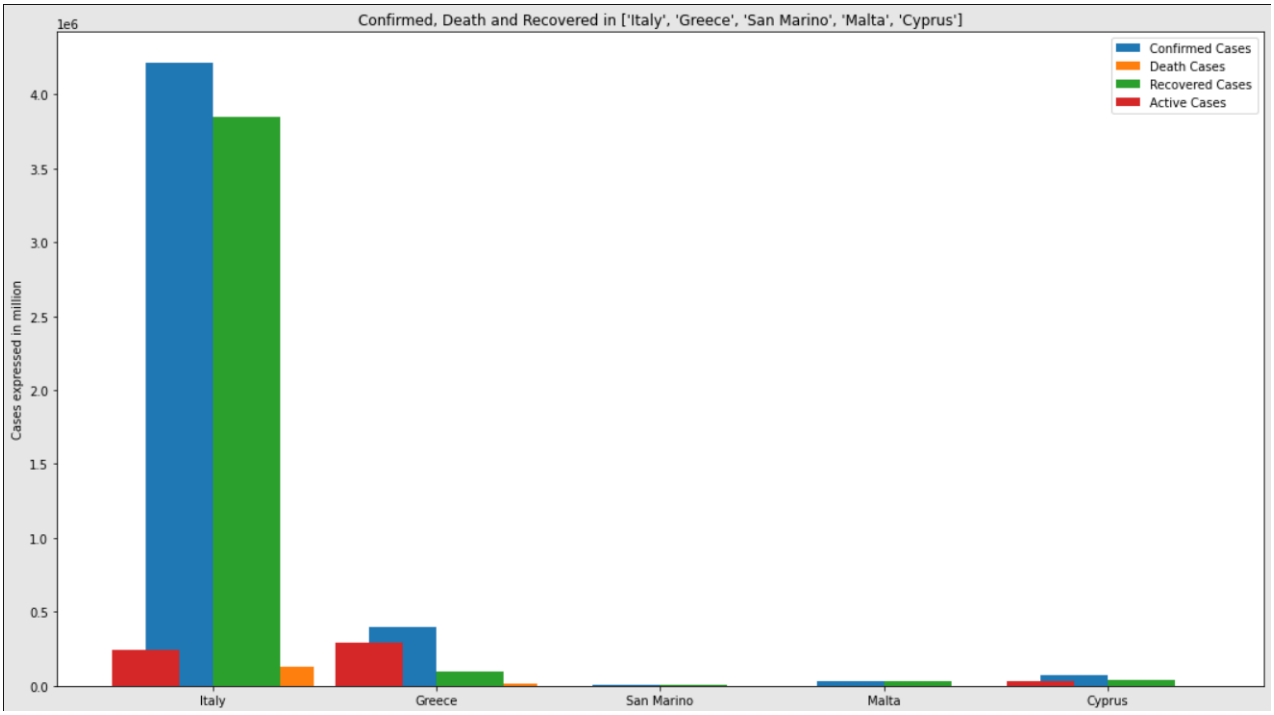


Figure 4.5: Southern Europe cases.

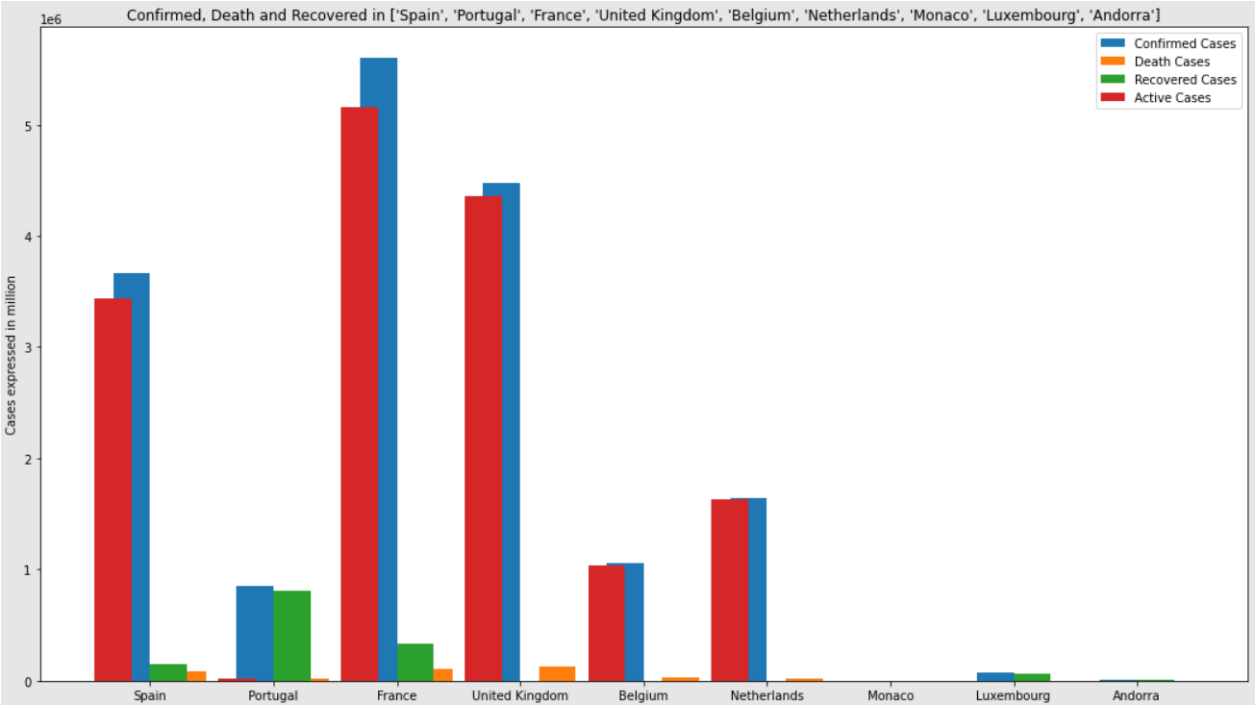


Figure 4.6: Western Europe cases.

From the analysis of every subregion in Europe we can see that Western European countries are the most affected like: The Netherlands, The United Kingdom, France, Spain. Another affected countries are Sweden and Serbia. Now let's see a top of countries with the highest mortality rate in Europe in the Figure 4.7:

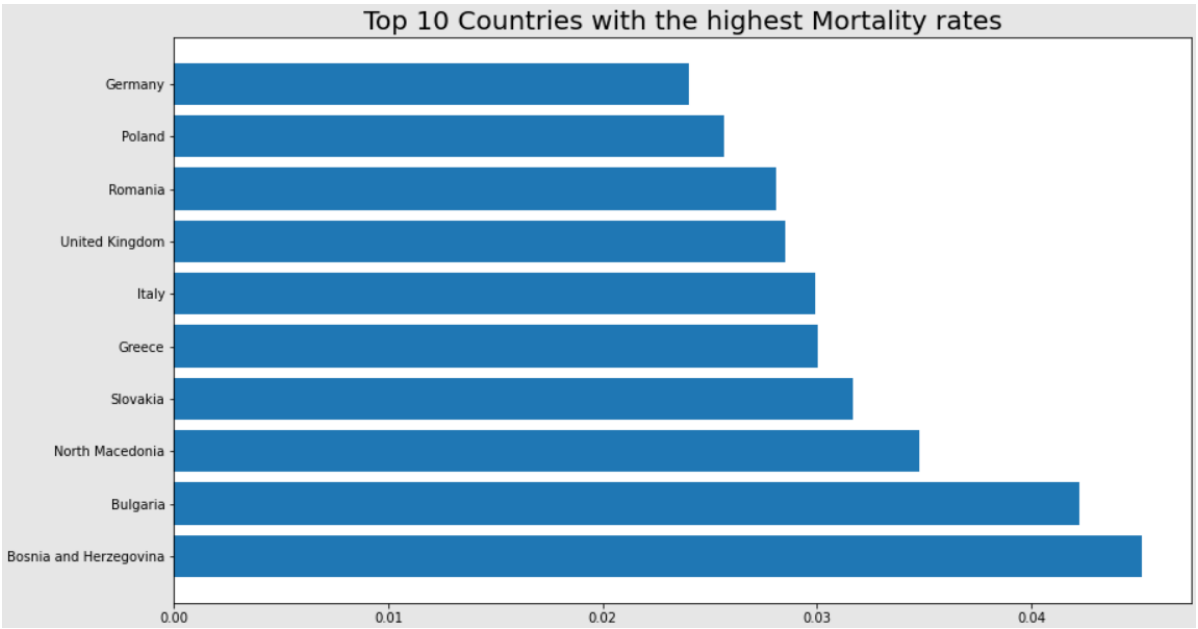


Figure 4.7: Top 10 mortality rates in Europe.

In South America the situation regarding the cases is shown in the Figure 4.8:

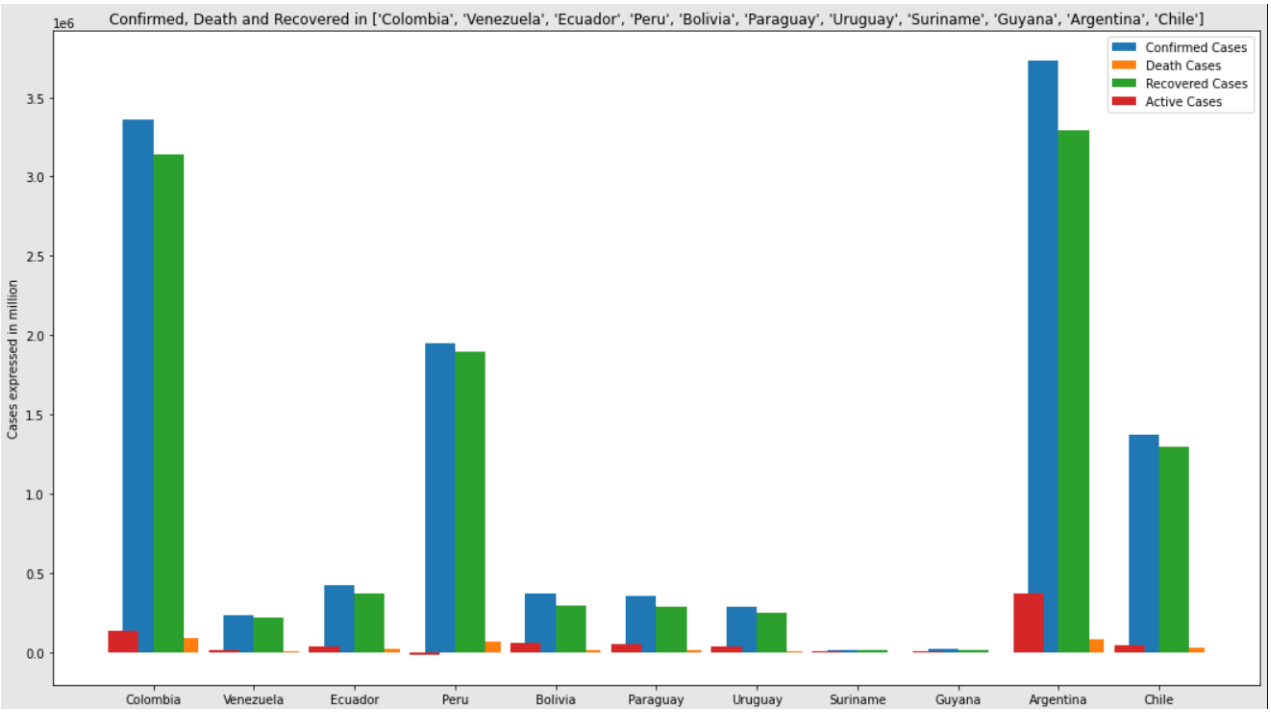


Figure 4.8: South America cases.

For the last part of this analytical study, the only thing to do is to cover the countries that I have selected for implementing machine learning models are: Ukraine, Bulgaria and Japan. For visualization, the blue line represents Ukraine, the yellow one represents Japan and the green one is Bulgaria. We will plot graphs to make a comparison between these three countries in the Figures: 4.9, 4.10, 4.11, 4.12, 4.13.

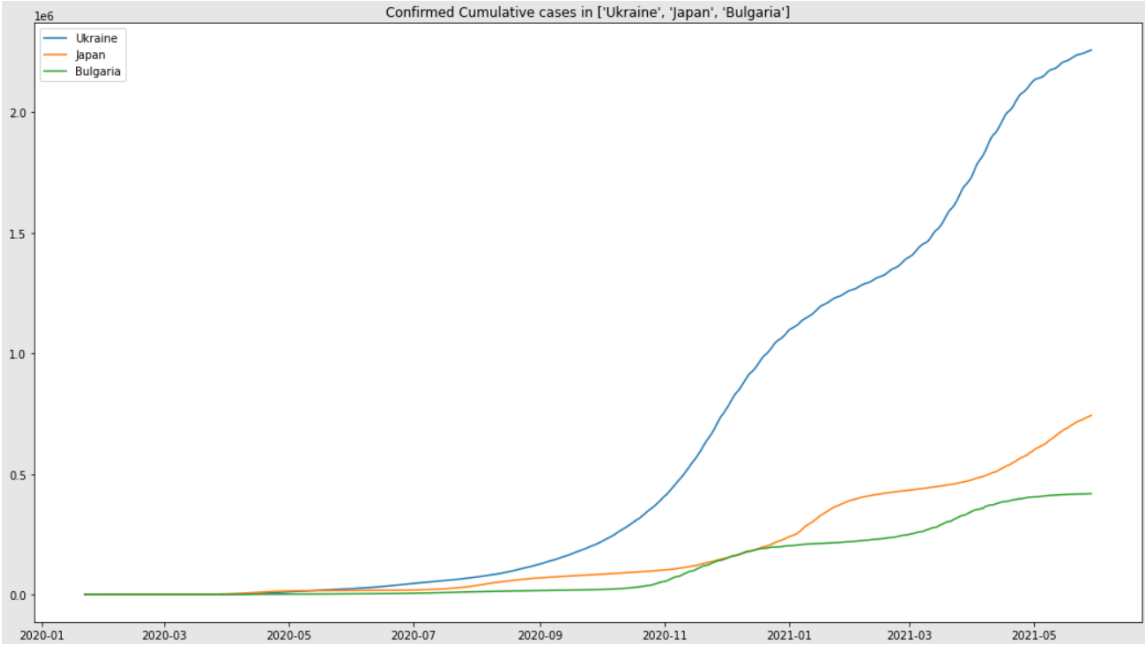


Figure 4.9: Confirmed cases in the countries of Ukraine, Bulgaria and Japan.

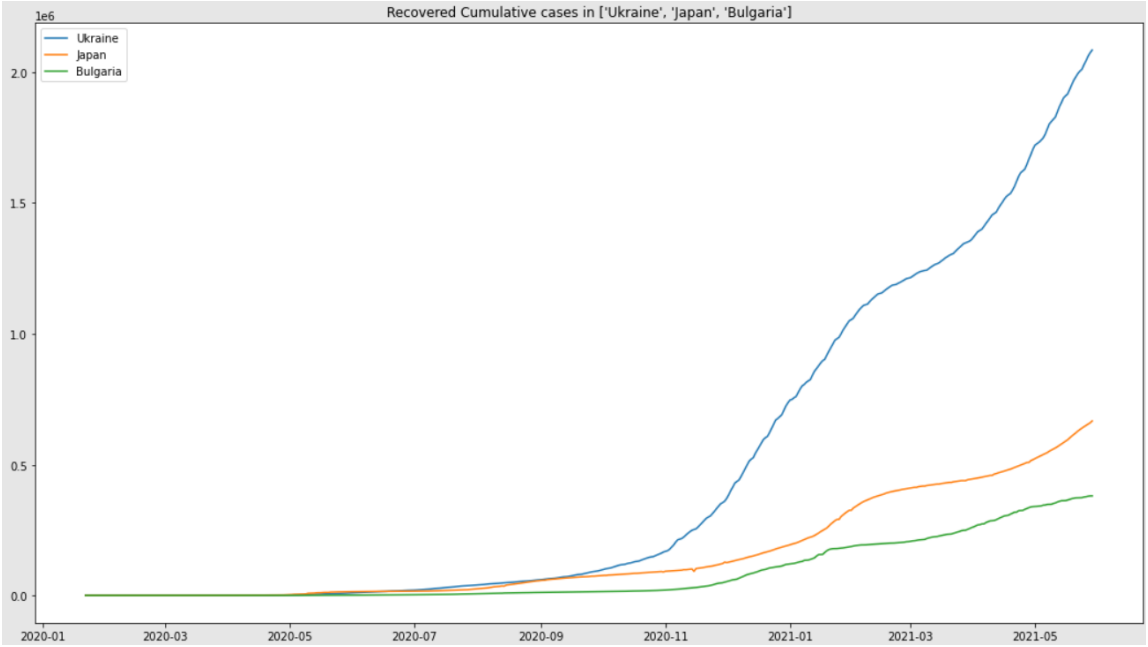


Figure 4.10: Recovered cases in the countries of Ukraine, Bulgaria and Japan.

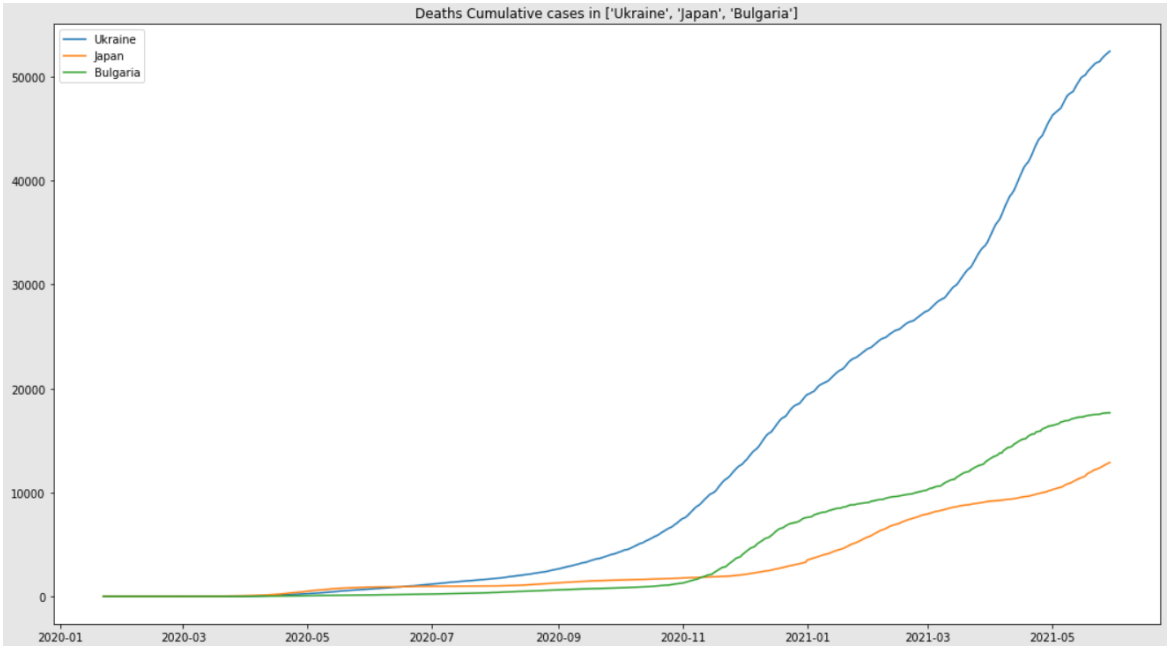


Figure 4.11: Death toll in the countries of Ukraine, Bulgaria and Japan.

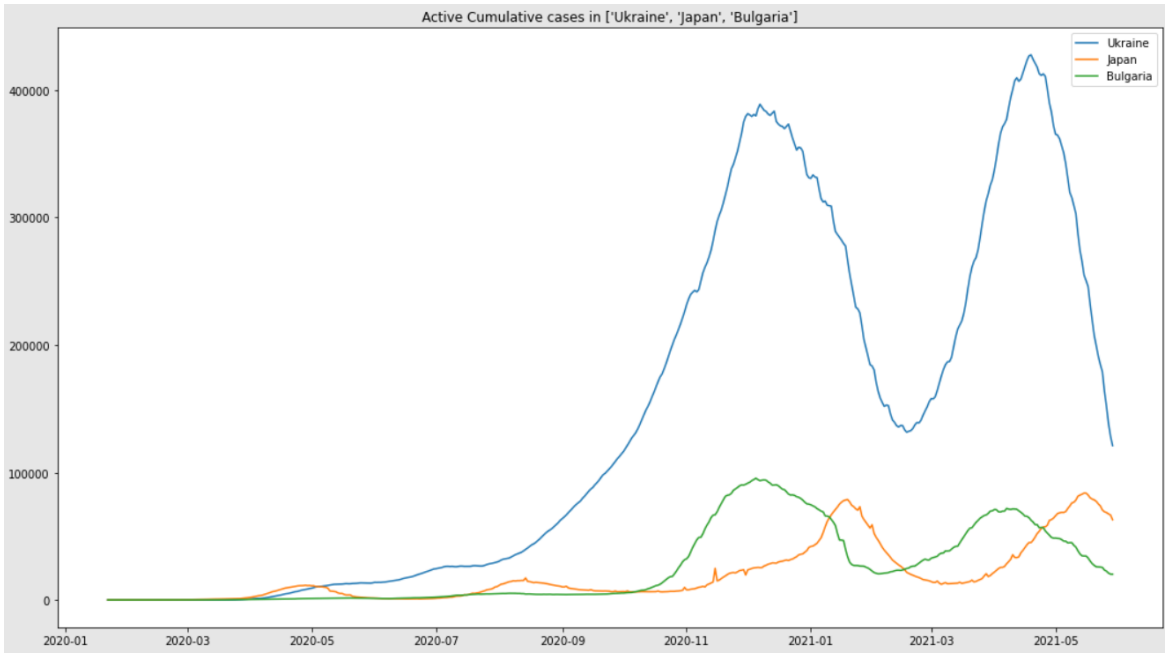


Figure 4.12: Active cases in the countries of Ukraine, Bulgaria and Japan.

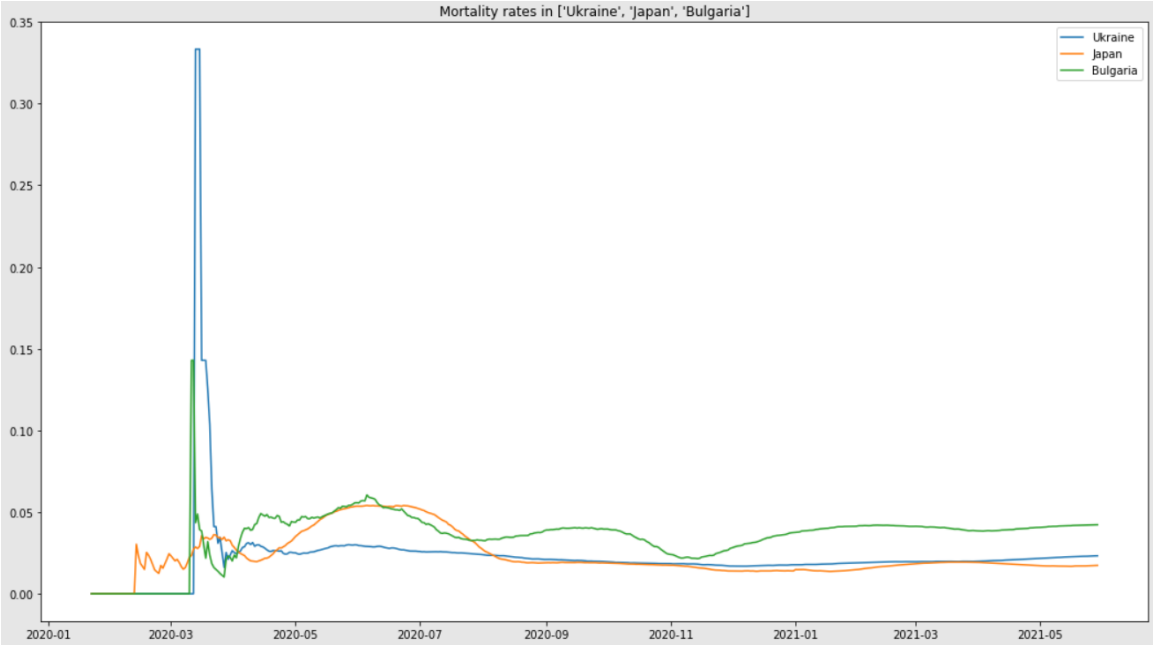


Figure 4.13: Mortality rate in the countries of Ukraine, Bulgaria and Japan.

The situation of Ukraine in detail can be seen in the Figure4.14:

(a) Ukraine's confirmed cases.

(b) Ukraine's death toll cases.

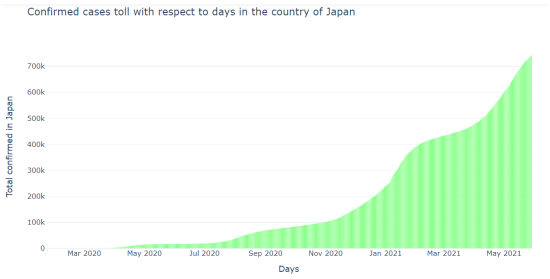
(c) Ukraine's recovered cases.

(d) Ukraine's active cases.

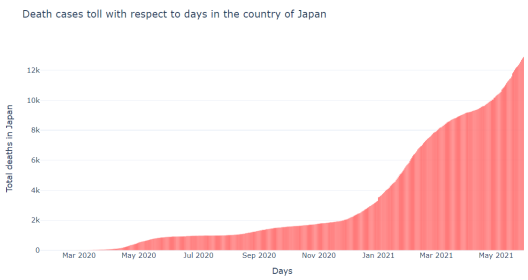
(e) Ukraine's mortality rate.

Figure 4.14: Ukraine's situation with the COVID-19.

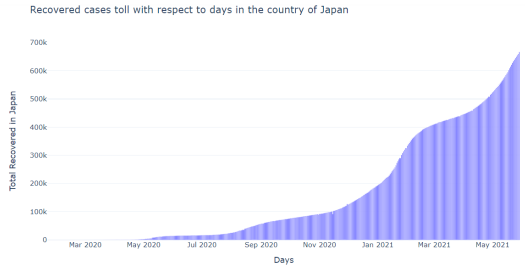
The situation of Japan in detail can be seen in the Figure 4.15:



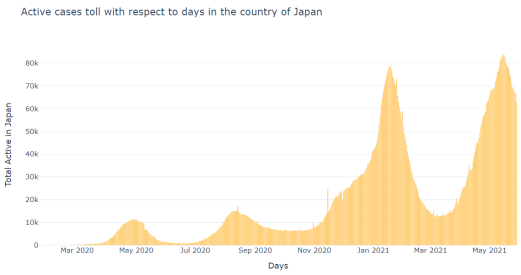
(a) Japan's confirmed cases.



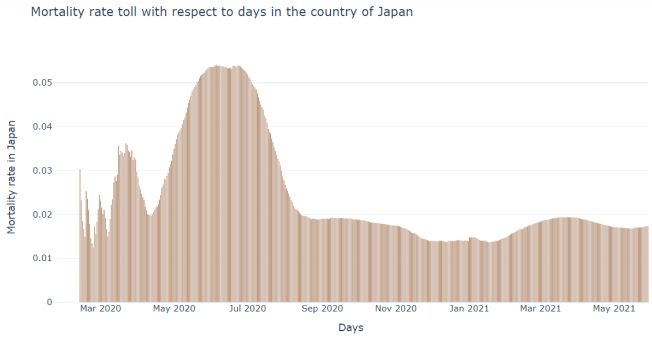
(b) Japan's death toll cases.



(c) Japan's recovered cases.



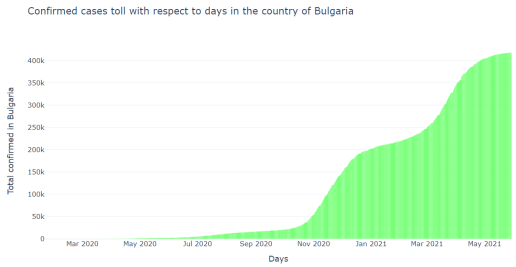
(d) Japan's active cases.



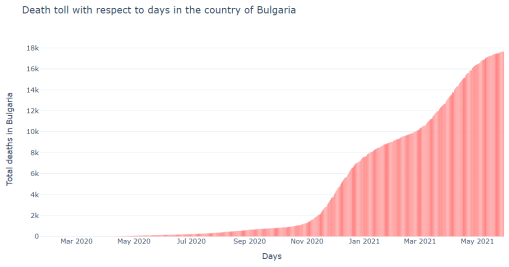
(e) Japan's mortality rate.

Figure 4.15: Japan's situation with the COVID-19.

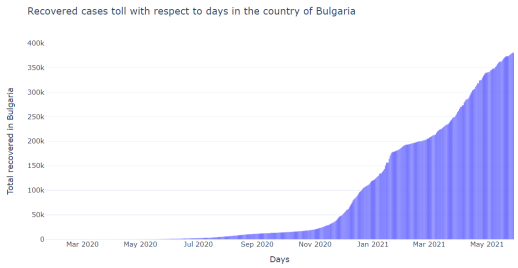
The situation of Bulgaria in detail can be seen in the Figure 4.16:



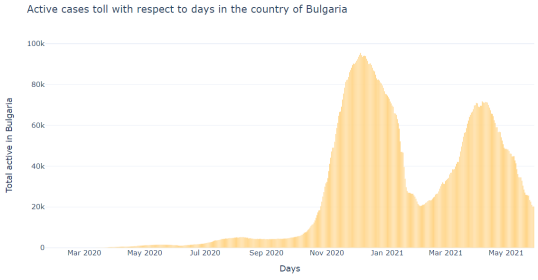
(a) Bulgaria's confirmed cases.



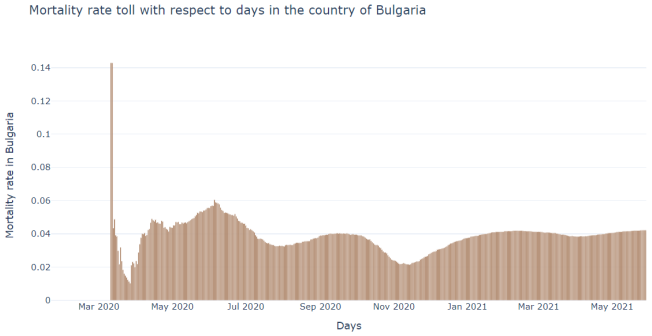
(b) Bulgaria's death toll cases.



(c) Bulgaria's recovered cases.



(d) Bulgaria's active cases.



(e) Bulgaria's mortality rate.

Figure 4.16: Bulgaria's situation with the COVID-19.

4.1 Metrics and Evaluation

The regression metrics that are going to be used are the coefficient of determination or simply R^2, mean squared error (MSE), mean absolute error (MAE) and their derivatives such as: root mean squared log error (RMSLE) and root mean squared error (RMSE). R^2 is by far the most intuitive metric to evaluate regression that computes the variance proportion on the target variables that can be explained by the predictor variables. Here down below are the formulas for every metric mentioned:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}|$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2$$

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2}$$

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(y_i + 1) - (\hat{y} + 1))^2}$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

where $\hat{y}$ is the predicted value of y, and $\bar{y}$ is the mean value of y.

For evaluation of the machine learning algorithms, the test split ratio is 85% training to 15% testing data. The predictions are made on the testing data. Here are going to be shown the tables with results with the metrics above, and also images that compare the actual unseen data compared to the prediction made by the algorithm.

4.2 Results of Support Vector Machines

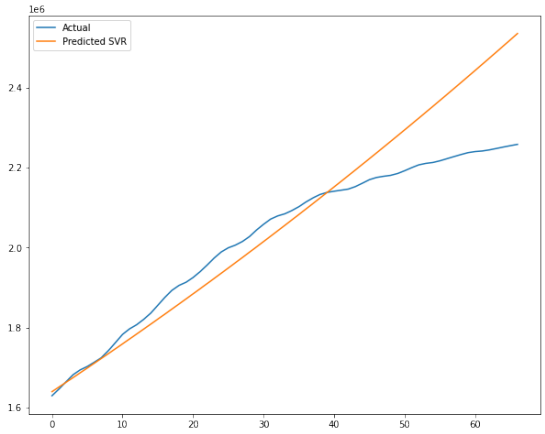
Ukraine's results on support vector machines are depicted in the table 4.1 and in Figure 4.17 are shown the lines of the actual compared to predicted values.

Category	MSE	MAE	RMSE	RMSLE	R-Squared
Confirmed	10458865190.74	70645.11	102268.59	23.07	0.7106
Deaths	5621752.87	2093.42	2371.02	15.54	0.85429
Recovered	1315099261.0	30320.96	36264.3	21.0	0.976081

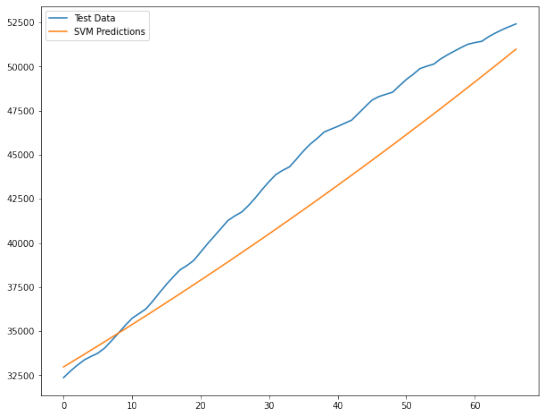
Table 4.1: Ukraine on SVM predictions scores.

These results were computed using a polynomial kernel, C and gamma parameters, C being the regularization parameter and gamma is the scale-length of the polynomial kernel, here exists a trade-off between these 2 parameters in which has to be found a balance. Instead of sticking with MSE, which is a bigger value to visualized, we'll used instead RMSE, tells us that the average deviation between the predicted values made by the model and the actual values is by average a resulted value. To make it clear, on the death category of Ukraine's data set, we can see on the RMSE, that the average deviation between these two variables is 2371.02. The goal is to obtain a lower RMSE and a higher R^2 tells us how well the independent variables can explain

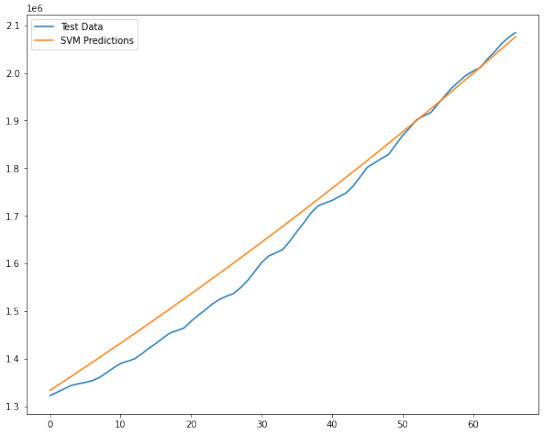
the variance of the dependent variables and this score observes how the algorithm performed well on the unseen data. The recovered and death categories obtained better results on R^2. On average Ukraine's R^2 is 85%. The reason that recovered got a bigger RMSE compared to deaths, is that in recovered are much bigger values compared to the deaths category.



(a) Confirmed cases SVM model fit.



(b) Death toll cases SVM model fit.



(c) Recovered cases SVM model fit.

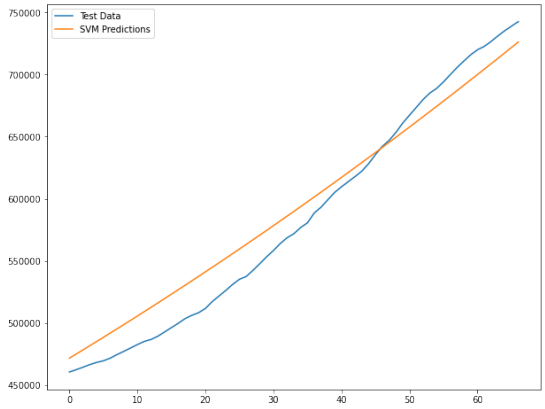
Figure 4.17: Ukraine's SVMs models fit.

Japan's results on support vector machines are depicted in the table 4.2 and in Figure 4.18 are shown the lines to the actual compared to predicted values.

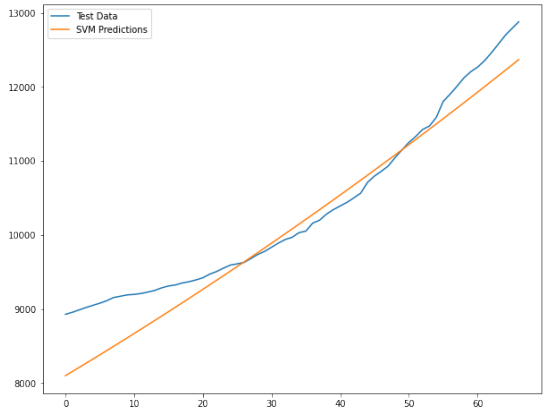
Category	MSE	MAE	RMSE	RMSLE	R-Squared
Confirmed	360180648.38	17490.87	18978.43	19.7	0.95525
Deaths	128328.61	272.5	358.23	11.76	0.90564
Recovered	170433042.52	10652.28	13055.0	18.95	0.964001

Table 4.2: Japan on SVM predictions scores.

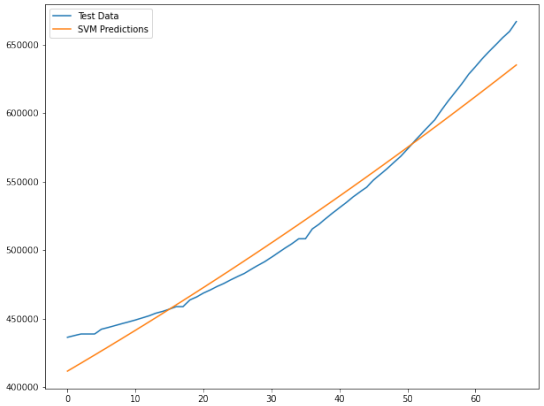
Here on the Japan data set, the support vector machine was configured with a polynomial kernel and as we can see, it obtained better results compared to Ukraine's score. The average result of the R^2 obtained a value of 94% which indicates the fact that the model fitted well the data set. Also the RMSE values of the three categories tells us there is not a big deviation difference between the 67 predicted and actual values.



(a) Confirmed cases SVM model fit.



(b) Death toll cases SVM model fit.



(c) Recovered cases SVM model fit.

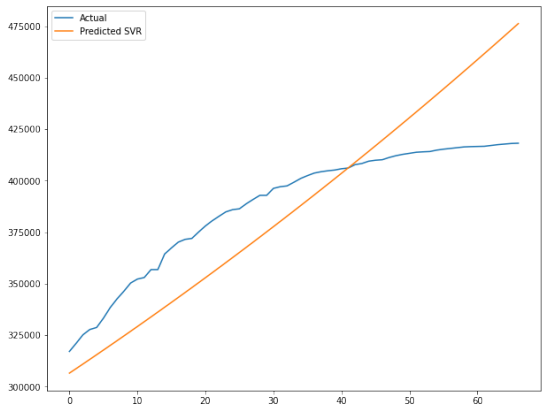
Figure 4.18: Japan's SVMs models fit.

Bulgaria's results on support vector machines are depicted in the table 4.3 and in Figure 4.19 are shown the lines of the actual compared to predicted values.

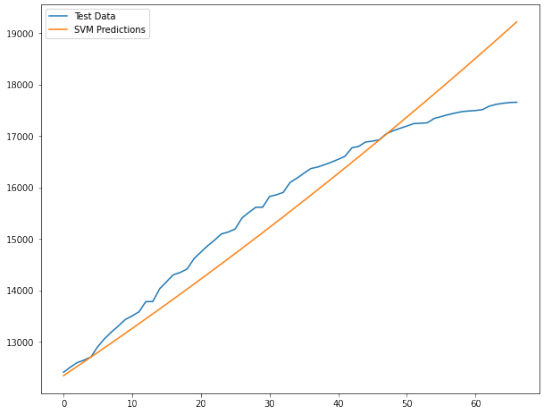
Category	MSE	MAE	RMSE	RMSLE	R-Squared
Confirmed	631533537.56	21621.16	25130.33	20.26	0.26802
Deaths	332541.19	460.96	576.66	12.71	0.87718
Recovered	230557117.48	13313.58	15184.11	19.26	0.871305

Table 4.3: Bulgaria on SVM predictions scores.

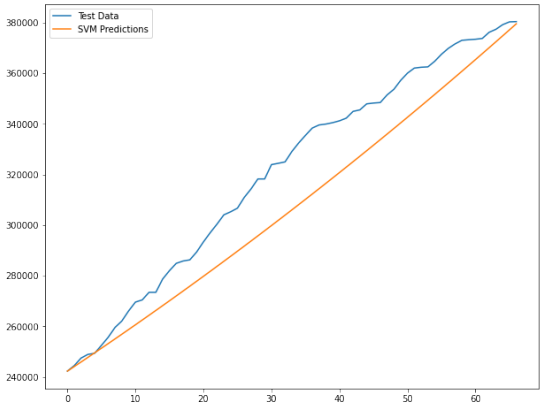
Same as the other 2 countries, on the support vector machine was used a polynomial kernel and it obtained lower results with an average of 67% of the R^2 and we can see down below how well the support vector machine fitted the unseen data. The confirmed category of the Bulgaria data set obtained a 26% meaning that it over-fited. To be noted is that the RMSE of the confirmed data category is way high: 25130, meaning that if the R^2 got a lower score, than the RMSE will have a higher value.



(a) Confirmed cases SVM model fit.



(b) Death toll cases SVM model fit.



(c) Recovered cases SVM model fit.

Figure 4.19: Bulgaria's SVMs models fit.

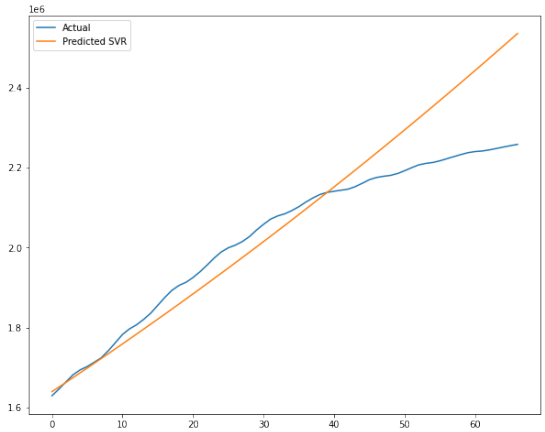
4.3 Results of the polynomial regression

Ukraine's results on polynomial regression are depicted in the table 4.4 and in Figure 4.20 are shown the lines comparing the actual and predicted values.

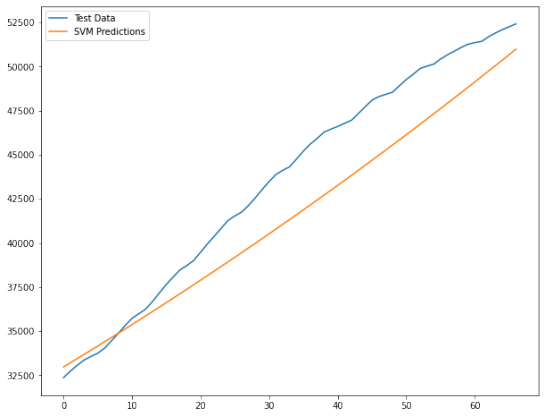
Category	MSE	MAE	RMSE	RMSLE	R-Squared
Confirmed	2390341807.14	42114.4	48891.12	21.59	0.93386
Deaths	3957951.1	1787.89	1989.46	15.19	0.89741
Recovered	27209624444.97	164008.12	164953.4	24.03	0.505104

Table 4.4: Ukraine on polynomial regression predictions scores.

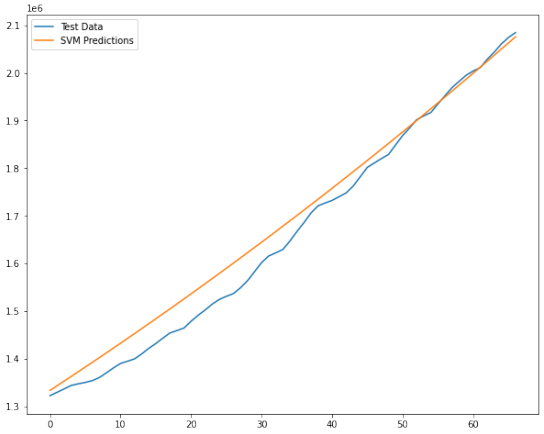
On the polynomial regression, the most important is to choose the degree of the polynomial, going past five, the model becomes complex and over-fits. Here on the Ukraine data set we can observe that on the recovered category the R^2 is quite small, indicating that the model over-fitted compared to the other two categories. Because it over-fitted the average R^2 is going to be smaller 77%. R^2 having a high values for the two categories (confirmed and deaths), the RMSE has a small value, meaning that it generalized.



(a) Confirmed cases regression model fit.



(b) Death toll cases regression model fit.



(c) Recovered cases regression model fit.

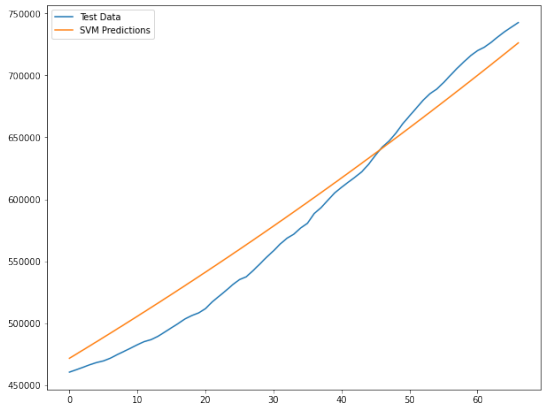
Figure 4.20: Ukraine's regression models fit.

Japan's results on polynomial regression are depicted in the table 4.5 and in Figure 4.21 are shown the lines comparing the actual and predicted values.

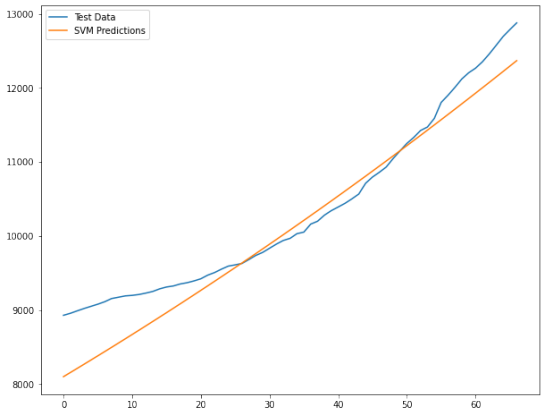
Category	MSE	MAE	RMSE	RMSLE	R-Squared
Confirmed	1271095340.61	30555.57	35652.42	20.96	0.84207
Deaths	86985.46	257.39	294.93	11.37	0.93604
Recovered	1634080584.39	38187.19	40423.76	21.21	0.654851

Table 4.5: Japan on polynomial regression predictions scores.

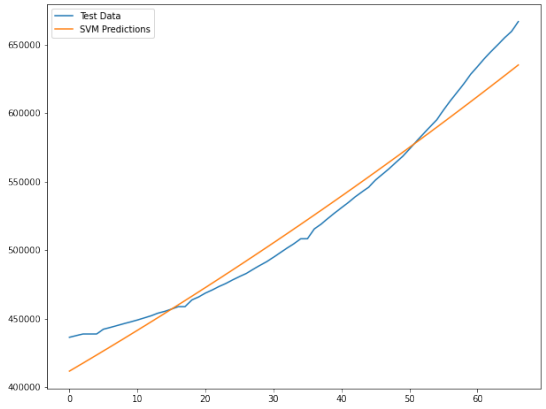
Compared to the support vector machine which performed better on the Japan data set, polynomial regression obtained lower results. But the average R^2 is not bad, having a value of 81%. On the recovered data of Japan, the model over-fitted, thus having a RMSE value bigger.



(a) Confirmed cases regression model fit.



(b) Death toll cases regression model fit.



(c) Recovered cases regression model fit.

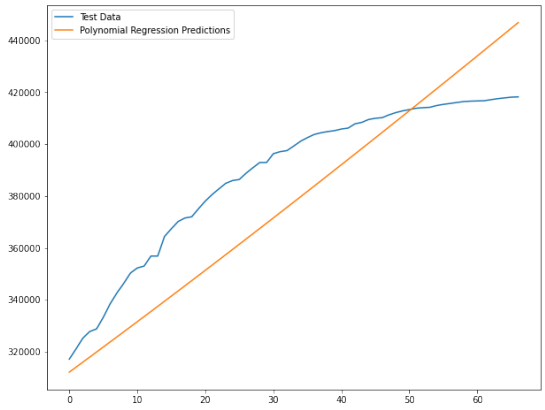
Figure 4.21: Japan's regression models fit.

Bulgaria's results on polynomial regression are depicted in the table 4.6 and in Figure 4.22 are shown the lines comparing the actual and predicted values.

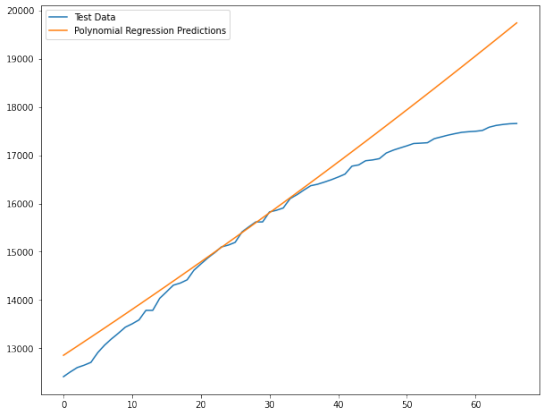
Category	MSE	MAE	RMSE	RMSLE	R-Squared
Confirmed	354400107.26	16974.5	18825.52	19.69	0.58923
Deaths	597163.78	529.08	772.76	13.3	0.77944
Recovered	49730853.86	5865.78	7052.01	17.72	0.972241

Table 4.6: Bulgaria on polynomial regression predictions scores.

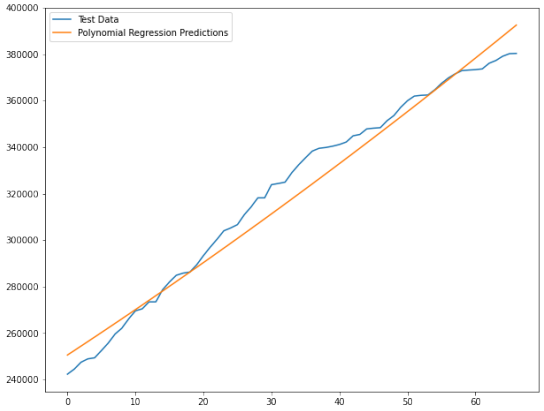
Bulgaria's data set performed better better obtaining an addition of +11% to the average R^2 score which is 78% compared to 67% on the support vector machine algorithm. The model over-fitted in the confirmed data and on deaths score it obtained a decent result of 78%.



(a) Confirmed cases regression model fit.



(b) Death toll cases regression model fit.



(c) Recovered cases regression model fit.

Figure 4.22: Bulgaria's regression models fit.

4.4 Results of Long short-term-memory

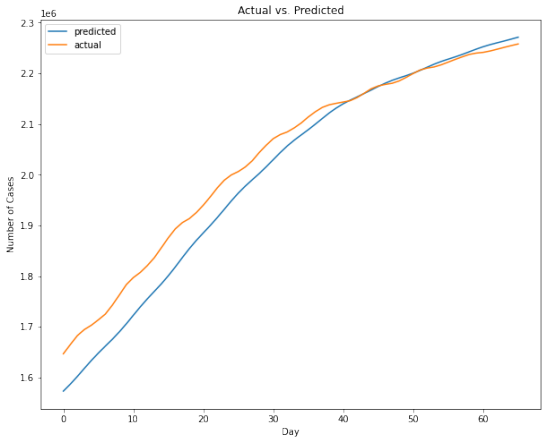
In addition to the R^2 and RMSE to measure the performance of the algorithm on the unseen data, we have graphs depicting loss training and loss validation, which determine if the model is over-fitting or under-fitting. To recognize under-fitting by looking to a graph like this, validation and train loss lines should be far from each other. Over-fitting is recognized by looking at the validation loss line, at some point it decreases, followed by an increase in the graph.

Ukraine's results on long short-term-memory are depicted in the table 4.7 and in Figure 4.23 are shown the lines comparing the actual and predicted values.

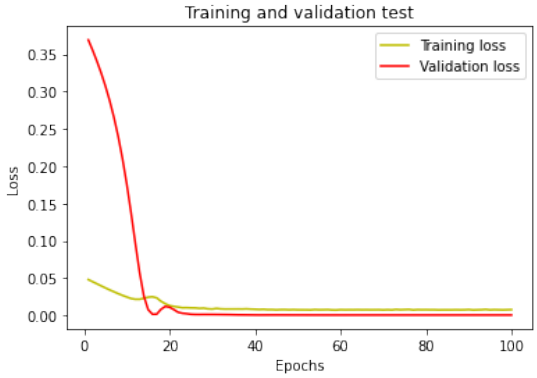
Category	MSE	MAE	RMSE	RMSLE	R-Squared
Confirmed	1988658094.79	34536.44	44594.37	21.41	0.94195
Deaths	698799.34	722.01	835.94	13.46	0.98124
Recovered	6209236206.85	78276.91	78798.71	22.55	0.88517

Table 4.7: Ukraine on LSTM predictions scores.

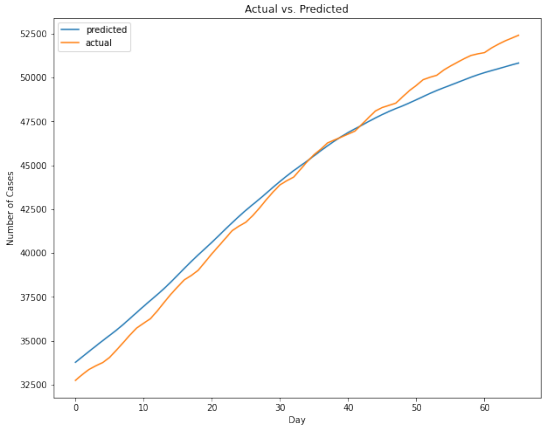
On the Ukraine data set, the long short-term-memory neural network obtained an average of 94% on the R^2 which is a great result compared to the other algorithms. To be noted is that R^2 having a high score on the deaths data category using LSTM of 98% and a RMSE value of 836, it can be observed from the other two algorithms on the deaths category data, the correlation between the R^2 and RMSE, for example on support vector machines R^2 is 85% with a RMSE of 2371 and on polynomial regression is 90% and with a RMSE of 1989. Also the graphs of the training and validation loss which can be seen below depict that the model generalized well.



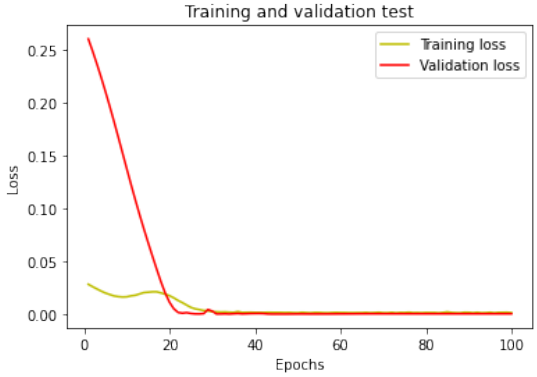
(a) Confirmed cases LSTM model fit.



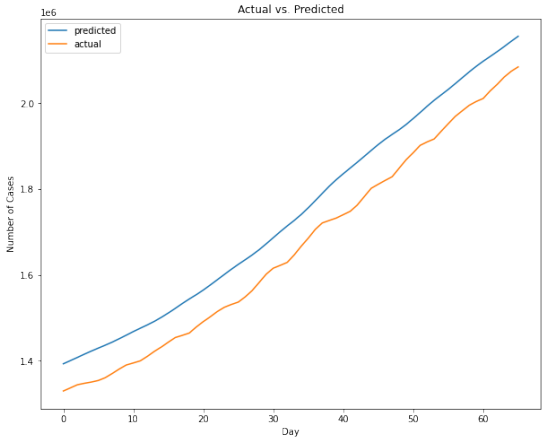
(b) Train and validation loss on the confirmed cases.



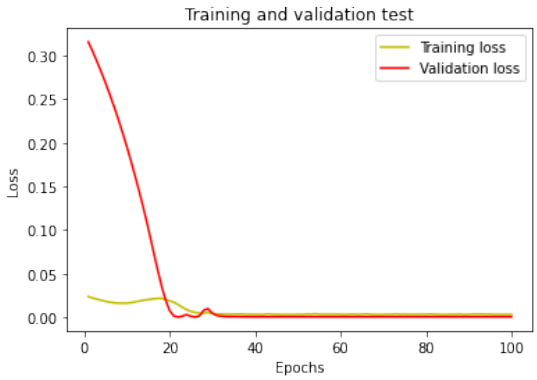
(c) Death toll cases LSTM model fit.



(d) Train and validation loss on the death toll cases.



(e) Recovered cases LSTM model fit.



(f) Train and validation loss on the recovered cases.

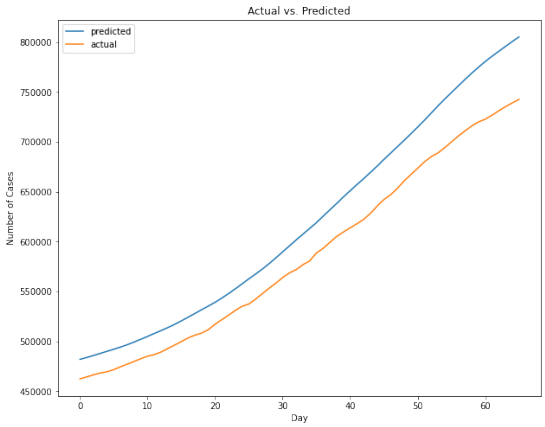
Figure 4.23: Ukraine LSTMs models fit.

Japan's results on long short-term-memory are depicted in the table 4.8 and in Figure 4.24 are shown the lines comparing the actual compared to and values.

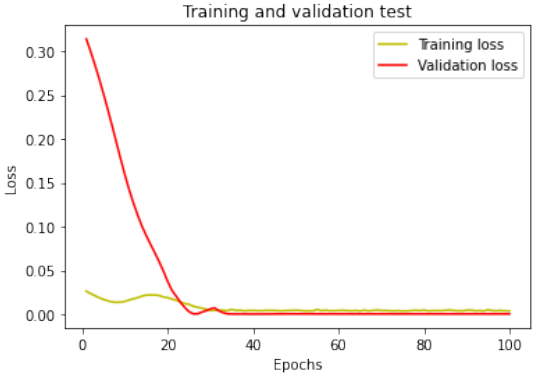
Category	MSE	MAE	RMSE	RMSLE	R-Squared
Confirmed	1288391410.21	33344.58	35894.17	20.98	0.83763
Deaths	180352.92	388.69	424.68	12.1	0.86641
Recovered	39628197.32	5554.88	6295.09	17.5	0.99156

Table 4.8: Japan on LSTM predictions scores.

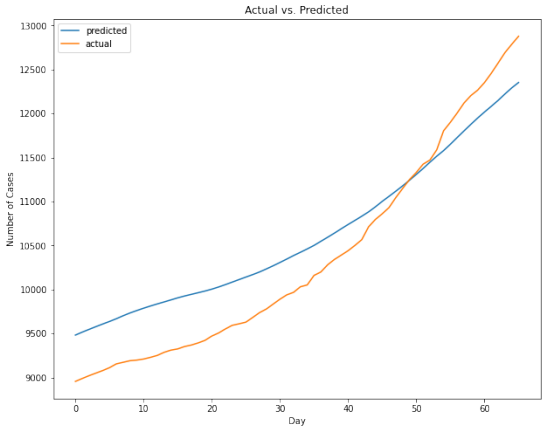
The average score of R^2 is 89% on average, meaning that the long short-term-memory was beaten by an addition of 5% of the support vector machine. What is to be remarked here is how well the long-short-term memory fitted the unseen data obtaining a score of 99% on R^2 and that the RMSE value is low, which is 6295 compared to the RMSE recovered data on the other algorithms: on support vector machine the RMSE is 13055 and the R^2 score is 96% and on polynomial regression the RMSE is 164008 and the score of R^2 is 65%



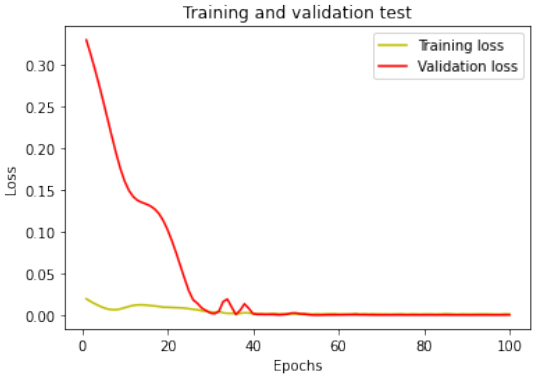
(a) Confirmed cases LSTM model fit.



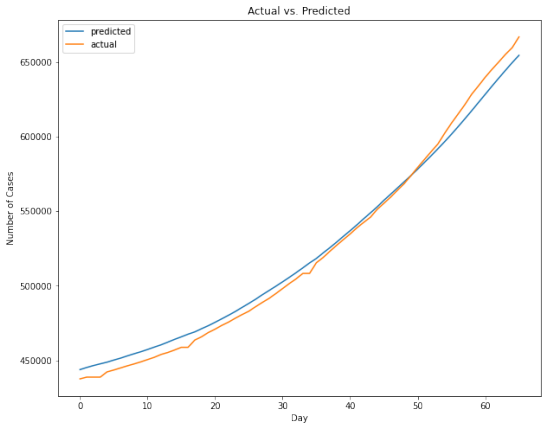
(b) Train and validation loss on the confirmed cases.



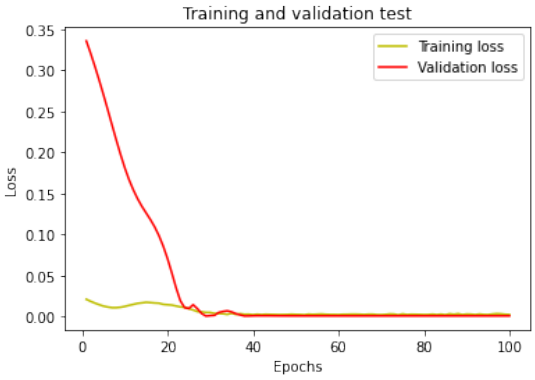
(c) Death toll cases LSTM model fit.



(d) Train and validation loss on the death toll cases.



(e) Recovered cases LSTM model fit.



(f) Train and validation loss on the recovered cases.

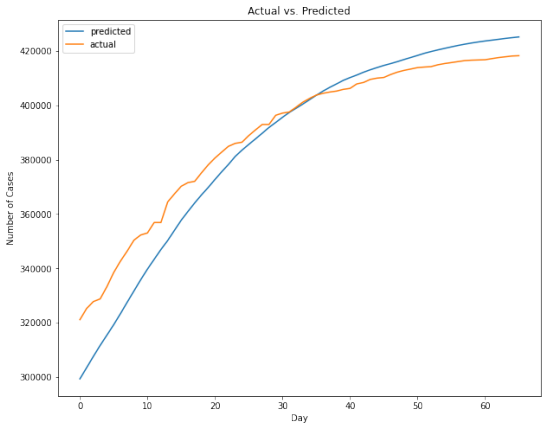
Figure 4.24: Japan LSTMs models fit.

Bulgaria's results on long short-term-memory are depicted in the table 4.9 and in Figure 4.25 are shown the lines comparing the actual compared to predicted values.

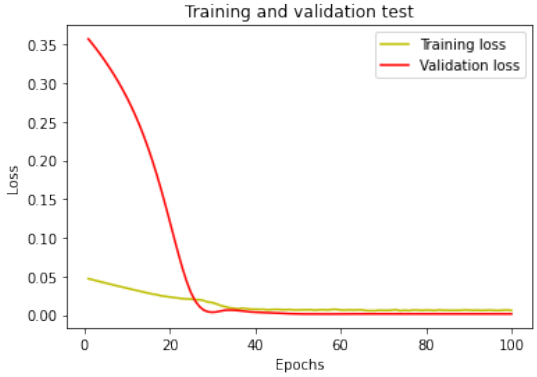
Category	MSE	MAE	RMSE	RMSLE	R-Squared
Confirmed	90959293.77	7494.22	9537.26	18.33	0.88602
Deaths	247924.19	487.18	497.92	12.42	0.90416
Recovered	332435846.44	15794.29	18232.82	19.62	0.80711

Table 4.9: Bulgaria on LSTM predictions scores.

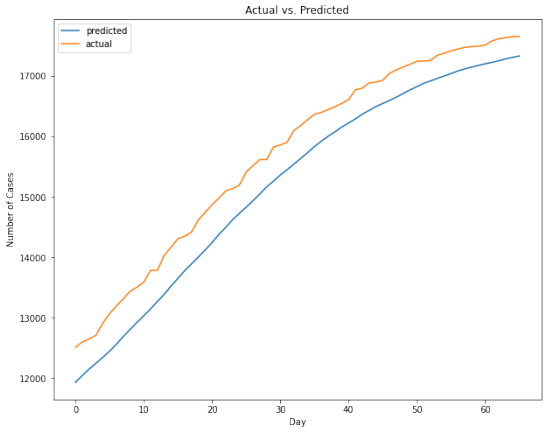
Bulgaria obtained an average of 86% on the R^2 score performed by the long short-term-memory, which is better than the support vector machine where it obtained 67% and polynomial regression with a 78% score.



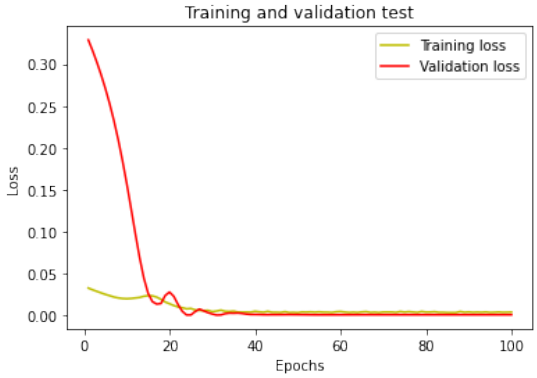
(a) Confirmed cases LSTM model fit.



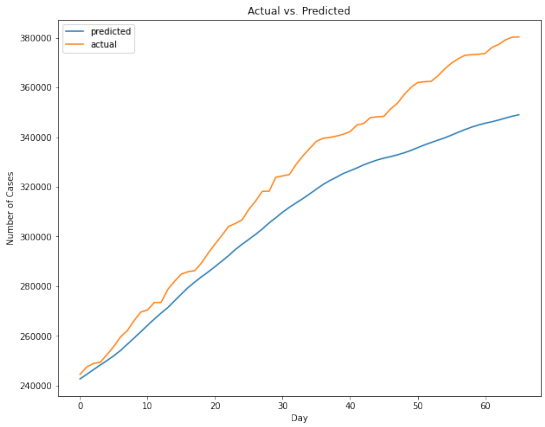
(b) Train and validation loss on the confirmed cases.



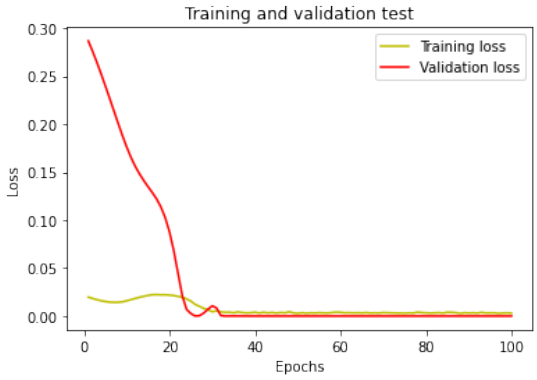
(c) Death toll cases LSTM model fit.



(d) Train and validation loss on the death toll cases.



(e) Recovered cases LSTM model fit.



(f) Train and validation loss on the recovered cases.

Figure 4.25: Bulgaria LSTMs models fit.

Conclusions

The global pandemic caused by the COVID-19 is a global security issue of many nations worldwide, because if the measures are not taken as soon as possible, hospitals could become overcrowded in ratio with the medical staff, and also many variants of the virus could evolve on becoming more deadly, or easily spreadable and scientists could work more on finding a cure on a variant.

Machine learning algorithms have become crucial in terms of predictions, they can be helpful in providing insights of the outbreak in terms of how spreadable is and how mortal it can be.

From the study I've conducted on machine learning on the COVID-19 data set of many countries of the world, and also from the scientific literature that I have been reading, firstly you should identify the nature of the problem, think of how you can model it and determine if the problem is a non-linear or simply linear. Applying simply machine learning on the given data set, will not result a decent performance. The collecting of the data should be filtered in the preprocessing method in order to produce slightly better results. Before that, the data should be scaled and after that algorithms should be applied, because it makes the model chosen to understand the problem easily.

When models fail, hyper parameter tuning and regularization should be applied, in order to combat problems such as over-fitting and under-fitting so that you get a generalized model.

From the algorithms that I have used on the three experimental countries, I can conclude that the model that performed the best was Long short-term-memory neural networks, because they obtained decent scores, the lowest being 81% on the Bulgaria's data set on the recovered category seen in the table 4.9. Also, the graphs, the images were I have represented the comparison between the actual and prediction values, shows that Long short-term-memory was able to fit the data better in most cases, com-

pared to Support vector machines and polynomial regression, where they had their ups, but also observable downs.

Bibliography

[1] Book by Andreas C. Muller and Sarah Guido, Introduction to Machine Learning - A Guide for Data Scientists, published by O'Reilly Media, 2016

[2] Book by Fadi Al-Turjman (editor), Artificial intelligence and machine learning for COVID-19, published by Springer, 2021

[3] Article by Muhammad, L. J.; Algehyne, Ebrahem A.; Usman, Sani Sharif; Ahmad, Abdulkadir; Chakraborty, Chinmay; Mohammed, I. A., Supervised Machine Learning Models for Prediction of COVID-19 Infection using Epidemiology Dataset, 2020, journal

[4] Article by Yadav, Milind; Perumal, Murukessan; Srinivas, M, Analysis on Novel Coronavirus (COVID-19) Using Machine Learning Methods, 2020, journal

[5] Support Vector Machine (SVM) and Kernels Trick

[6] Article by Khakharia, Aman; Shah, Vruddhi; Jain, Sankalp; Shah, Jash; Tiwari, Amanshu; Daphal, Prathamesh; Warang, Mahesh; Mehendale, Ninad, Outbreak Prediction of COVID-19 for Dense and Populated Countries Using Machine Learning, 2020, journal

[7] Recurrent Neural Networks-IBM

[8] Deep Learning-IBM

[9] Article by Kong, Yun-Long; Huang, Qingqing; Wang, Chengyi; Chen, Jingbo; Chen, Jiansheng; He, Dongxu, Long Short-Term Memory Neural Networks for Online Disturbance Detection in Satellite Image Time Series, 2018, journal

[10] An Introduction to Polynomial Regression