Time Series Forecasting

Spyridon Barmpakos

I. Description of the Project

The idea of this project is to use data from google trends to predict a possible abnormality in the health status of the general population.

I assumed that people usually google their symptoms when they are experiencing a health issue. For example, a person that has a heavy cough may think that he has pneumonia and thus google "pneumonia symptoms".

As shown in google trends and as expected, people search for this term more in the winter than in the summer. That is of course because people experience more respiratory problems in the winter. Also, if we only consider the data before the covid outbreak, we observe that roughly the same amount of people search for this term each year. In this project, I apply several Deep learning algorithms to predict the number of google searches for the phrase "pneumonia symptoms".

The idea is that, if the actual searches outnumber the predictions, there might be something going wrong with the health of the general population, as more people have pneumonia-like symptoms than expected.

II. Introduction to Time Series

Before diving into the main part of this project, let us first define and describe the area of machine learning we are going to explore. In general, Time Series Forecasting is used to predict the future by analyzing past events. In other machine learning fields, we assume that each observation is independent, while in Time series, we assume a temporal dependency between them and we make use of it when we train our models. Some examples of time series are: Daily temperature data, Daily sales of a specific product and daily births.

III. Fundamentals

Time series in general can be broken down into 4 parts.

- **Level** The average value of the series
- **Trend** The Tendency of the series to increase or to decrease.
- **Seasonality** The repeating patterns in the data
- **Noise** The random variability of the data, that cannot be modeled.

time series are a combination of the above characteristics.

IV. Data preparation

Another crucial part of the time series forecasting is the pre-processing of the data.

A. Supervised learning

One super important step in the data preparation is the transformation of the time series data to a supervised learning dataset, to be able to use machine learning algorithms. To do that, we use previous time steps as our independent variable to predict the future values. So, we set a window of observations as the input and a set of the following observations as the output. We then perform the same actions by sliding the window through all the observations and the dataset is ready. The size of the window is called the lag.

B. Data visualization

One way to visualize the data is a lag scatter plot, that models the relationship between an observation and the previous one. A lag scatter plot that is stretched along a line with a positive slope shows a positive correlation relationship and the opposite is true for a scatter plot that follows a negative-sloped line.

Another data visualization tool is the autocorrelation plot, which is used to prove that our data values are not random. In this kind of plot, we observe how an observation is correlated with all the lagged observations. An Autocorrelation plot with a sinusoidal shape indicates seasonality in the data.

C. Various data pre-processing techniques

An important step in data preprocessing is the resampling of the data. If we need less information and want data along bigger intervals, we use downsampling. If we need more detailed information and at a higher frequency, we upsample the data. But of course, we cannot produce information from zero, so upsampling involves using techniques like interpolation to fill in the gaps.

For the machine learning algorithms to perform well, the dataset has to have a fixed mean and variance, which is often not the case with time series data. So, to achieve that, we use power transforms. Time series that show a quadratic growth, need a square root transform to become linear (and as a result, to have a fixed mean a variance). If the series show an exponential growth, we can use a log transform.
If we want a more automated process, we can apply a Box-cot transform, which finds the optimal transformation of the data

to make them Gaussian.

Another process that is not only used for data preparation, but also for making predictions, is Moving Average Smoothing. As its name suggests, it smooths out the dataset, by replacing each observation with the mean of the window that consists of the current observation and its neighboring ones, or the current observation and a certain number of past observations. This procedure has the result of reducing the random fluctuations of the data and thus, revealing the exploitable information for the machine learning algorithms.

D. White Noise

In time series forecasting, given a set of observations, we try to determine the underlying signal that produces these observations and we make predictions by following this signal to the future. But, in most cases, the given observations are the sum of the underlying signal and some normally distributed noise, with mean equal to zero.
If the dataset consists of normally distributed observations with a mean of zero and a constant standard deviation, that means that there is no underlying signal present. In this case, we say that our Time Series is White Noise. If this is true, there is no point in trying to fit ML models to the data, as there is no useful information to be exploited.

E. Remove Trends and Seasonality

Generally, it is easier for models to fit and predict the data if the time signal has fewer constantly changing elements. Such elements can be the trend and the seasonality. If we try to model them and remove them from the observations, the signal becomes simpler, and the models can perform better on it.

We can remove a trend by fitting linear models to the data and then replacing the dataset with the residuals, or by differencing the time series. That requires replacing each observation with the difference of the original value and the previous one.

When removing seasonality, if we know the interval in which the Series is repeating, we can apply differencing by removing from each observation the corresponding observation from the previous repetition of the interval . We can also fit sinusoidal models to the data and replacing the values with the residuals.

V. Forecasting

When it comes to Time Series Forecasting, one can apply deep learning techniques, as well as standard statistical techniques. Before diving deeper into prediction with deep learning, let us first introduce Autoregressive Integrated Moving Average Model, or ARIMA for short.

A. ARIMA

- **AR:** Autoregression. The use of the previous observations as inputs to a regression equation to predict the future values.
- **I** Integrated. The process of making the data stationary, as described above.
- **MA** Moving Average. The application of regression models to the prediction errors.

The ARIMA method requires three parameters:

- p: How many lag values the model uses
- d: How many times we repeat the differencing process to make the Series stationary.
- q: The width of the window for the Moving Average algorithm.

B. Deep Learning Techniques

In this section, we are going to introduce the types of Neural Network architectures used for Time series forecasting.

- **MLP** MultiLayer Perceptron: The simplest form of neural networks, that requires a fixed number of inputs and outputs.
- **RNN** Recurrent Neural networks: Maybe the most suitable form of Neural Networks for time Series forecasting. This is because, RNNs use the state of the network in previous time steps as an input for the current time step. This way, information about the temporal dependency of the data can be encoded and modeled.
- **CNN** Convolutional Neural Networks. CNNs may outperform MLPs for Time Series Forecasting, because they use convolutional techniques to reveal additional information encoded into the Data

VI. AutoML

A. Introduction

Artificial intelligence has gain a lot of popularity in the recent years and more and more fields make use of machine learning algorithms to solve problems. However, the people that want to apply those algorithms may not be data scientists and thus want a simpler way of implementing them, without knowing exactly what is going on internally. This is where AutoML comes in handy. It consists of a pipeline of actions that prepare the data, search for the right model to apply, tune its hyperparameters and evaluate it. This idea has been implemented in various frameworks. Now, let's take a closer look into its pipeline..

B. pipeline

- **Data pre-processing**. In this part, the algorithm takes the provided data or generates new ones. Then, it applies feature engineering to the data, so that they are ready to be fed into the machine learning algorithms. This can include transformations, dimension reduction techniques, interpolation, etc.
- **Model generation**. Here, the algorithm first chooses the right model, and then optimizes it. The way this is done is described below.

- Model evaluation. In this part, the algorithm chooses the right evaluation method and trains the selected model. The research regarding AutoML has made breakthroughs in accelerating this process.

C. The Nas algorithm

Selecting the right neural architecture that best fits your data can be a time consuming task that needs a lot of resources.So, many companies that focus on productivity, make use of this method. At first, the algorithm defines a search space, which consists of a set of operations (e.g. fully connected, convolution, pooling) and the way those operations interact to make a neural architecture.

In order to do that, it has to take into consideration the performance and the application of each operation and combine them in an optimal way to create candidate architectures for the proposed problem.

Finally, the algorithm evaluates the candidate architectures and further tunes their hyperparameters. After the algorithm is finished, the right model architecture is chosen.

D. Frameworks

1) autoWEKA: autoWeka is based on a platform called WEKA (Waikato Environment for Knowledge Analysis). The main advantage of the WEKA platform is that it is easy to use and provides the user with access to a large variety of machine learning algorithms. AutoWEKA automates this task and chooses the right algorithm for the proposed problem. It is easy to use and focuses on non-expert users.

2) auto-sklearn: auto-sklearn is an open-source library that searches through the scikit-learn library for algorithms using the bayesian optimization method and picks the one that gives the best performance. What makes auto-sklearn different is that when searching for the right algorithm, it takes into account the performance of the candidate models applied to similar datasets. This is also known as meta-learning

REFERENCES

[1] Brownlee, Jason. introduction to Time Series Forecasting with Python

[2] Brownlee, Jason. Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python

[3] Brownlee, Jason. Automated Machine Learning (AutoML) Libraries for Python

[4] AutoML: A survey of the state-of-the-art Xin He, Kaiyong Zhao, Xiaowen Chu