Model Summary

The various algorithm used in building this model are as follows:

* Linear Regression
* Random Forest Regressor
* Random Forest Classifier
* Prophet
* Recurrent Neural Network
* SARIMAX

The dataset was collected from FRSC database. The data was quite dirty and it had to cleaned manually using Excel and python. After cleaning the data and performing descriptive statistics on the data, it was used in building models.

Each model was evaluated on three metrics to check its performance. The lower the error metric, the more accurate the model. The three metrics used are: **Mean Absolute Error, Mean Squared Error, and Mean Percentage Error**.

After a couple of cross validation, RNN and SARIMAX were later chosen. SARIMAX outperformed the RNN in capturing seasonal variation in accidents, hence, it was chosen for the final forecasting.

RNN

Neural networks are capable of transforming your input into the desired output. The process is the same in deep learning, though the transformation is more complex. Deep learning, as opposed to a simpler neural network with a few layers, relies on more layers to perform complex transformations. The output of a data source is connected to the neural network's input layer, and the data is processed by the input layer. The hidden layers map the patterns and associate them with a specific output, which can be a value or a probability. This process works flawlessly with any type of input, but it works especially well with images. After each layer processes its data, it sends the transformed data to the next layer. The data is processed by the next layer independently of the previous layers. The use of this strategy implies that if you feed a video to your neural network, the network will process each image individually, one after the other, and the result will not change even if you shuffle the order of the provided images. You will not benefit from the order of information processing if you run a network in this manner.

## Recurrent neural networks: Modeling sequences using memory

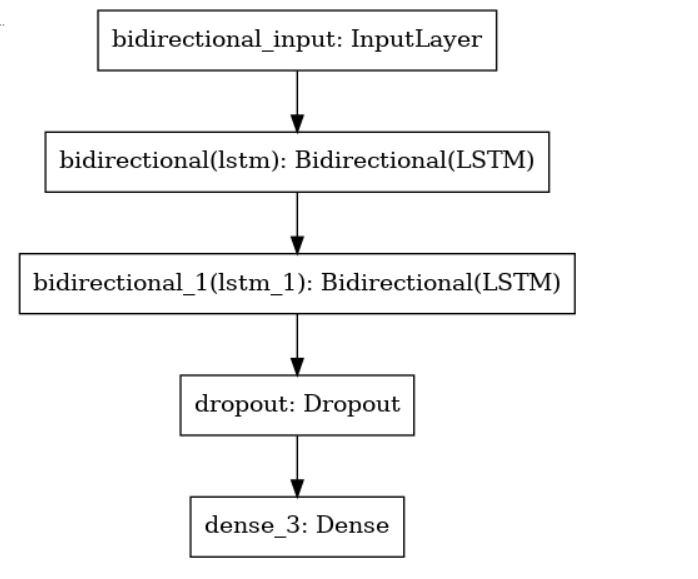
## The idea behind RNNs is simple, they examine each element of the sequence once and retain memory of it so they can reuse it when examining the next element in the sequence. It’s akin to how the human mind works when reading text: a person reads letter by letter the text but understands words by remembering each letter in the word. In a similar fashion, an RNN can associate a word to a result by remembering the sequence of letters it receives. Deep Learning Interview Guide for 2021 - Blogs | Fireblaze AI School

Dropout for RNN

Dropout is a famous methodology for improving the generalization error.

The basic concept is to, at each minibatch, set to zero 50% of the outputs in the neural net (20% for the input layer).

Applying dropout in RNN is difficult,



*What is SARIMAX?*

Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors, or [SARIMAX](https://www.statsmodels.org/dev/examples/notebooks/generated/statespace_sarimax_stata.html), is an extension of the [ARIMA](https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average) class of models. Intuitively, ARIMA models compose 2 parts: the autoregressive term (AR) and the moving-average term (MA). The former views the value at one time just as a weighted sum of past values. The latter model that same value also as a weighted sum but of past residuals (confer. *time series decomposition*). There is also an integrated term (I) to difference the time series (we will discuss this further below).

Overall, ARIMA is a very decent type of models. However, the problem with this vanilla version is that it cannot handle seasonality — a big weakness. Comes SARIMA — the predecessor of SARIMAX. One shorthand notation for SARIMA models is:





where *p* = non-seasonal autoregressive (AR) order, *d* = non-seasonal differencing, *q*= non-seasonal moving average (MA) order, *P* = seasonal AR order, *D* = seasonal differencing, *Q* = seasonal MA order, and *S* = length of repeating seasonal pattern. We will use this notation from now on. By adding those seasonal AR and seasonal MA components, SARIMA solves the seasonality problem.

SARIMAX extends on this framework just by adding the capability to handle exogenous variables. Holidays is the go-to option, but you can also get your own domain-specific features if you need to.

The SARIMAX implementation we used in the project was from the statsmodels package.

*Lag*

Lags are simply delays in time steps within a series. Consider a time index t, the lag 1 time index with respect to t is simply t-1, lag 2 is t-2, and so on.

*Stationarity*

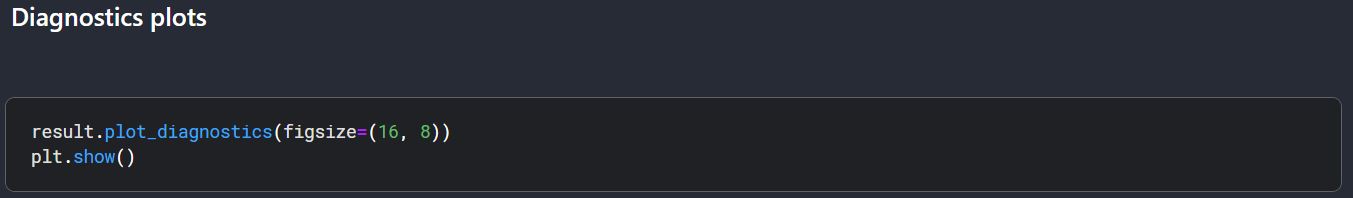
A stationary time series is one that has its mean, variance and autocorrelation structure unchanging overtime. In other words, it does not have any cycle/trend or seasonality. The ARMA models family is actually built on this concept.

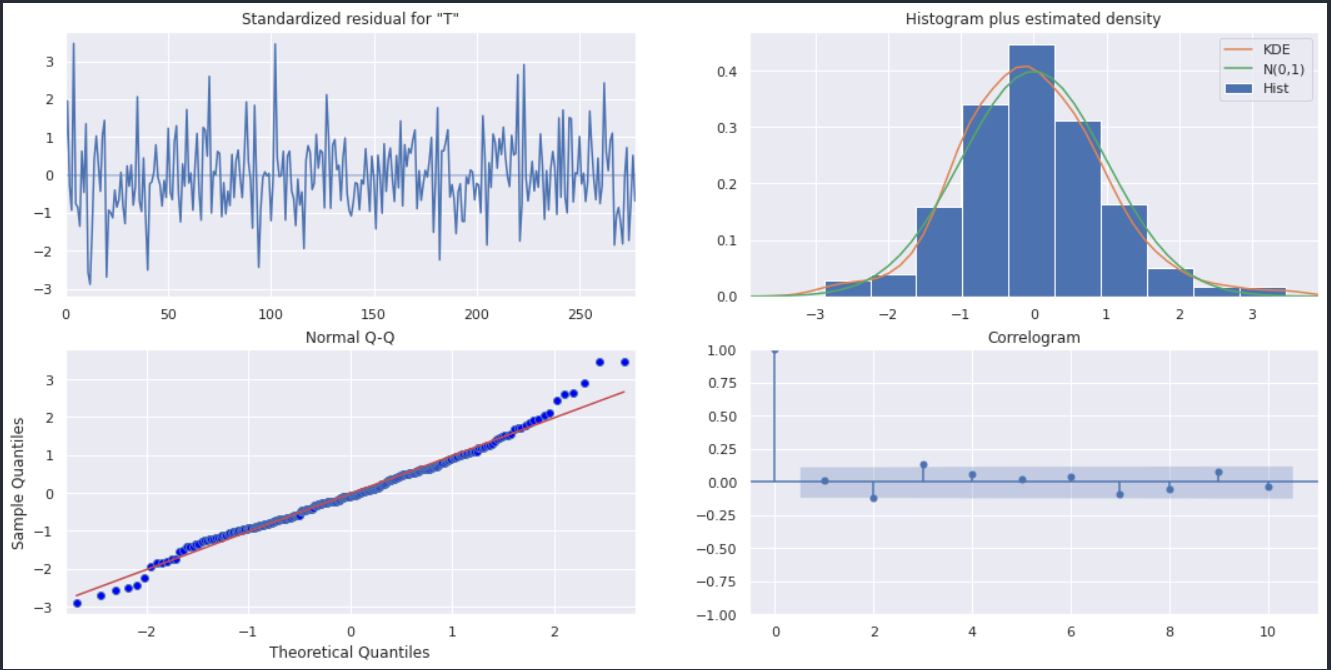
Model Validation

*Residual Diagnostics*

To determine the goodness of fit of the model, we can examine its residuals using the standard assumption: they should be normally distributed around 0, or in other words, white noise.

We can check this by looking at the various plots showing the distribution of the residuals. This can be generated conveniently using the plot\_diagnostics method. In addition, the Ljung-Box test can also be used to do this more precisely.





**Time Series Forecasting**

Time series forecasting is the process of making scientific predictions based on time stamped data from the past. It entails developing models based on historical data and using them to make observations and drive future strategic decisions. An important distinction in forecasting is that the future outcome is completely unknown at the time of work and can only be estimated through careful analysis and evidence-based priors.

**What is time series forecasting?**

The process of analyzing time series data using statistics and modeling to make predictions and inform strategic decision-making is known as time series forecasting. It's not always an exact prediction, and the likelihood of forecasts can vary wildly—especially when dealing with time series data's commonly fluctuating variables as well as factors beyond our control. Forecasting insight, on the other hand, about which outcomes are more likely—or less likely—to occur than other potential outcomes. Often, the more detailed the data, the more accurate the forecasts can be. While forecasting and “prediction” generally mean the same thing, there is a notable distinction. In some industries, forecasting might refer to data at a specific future point in time, while prediction refers to future data in general. Series forecasting is often used in conjunction with time series analysis. Time series analysis involves developing models to gain an understanding of the data to understand the underlying causes.

