MTH 517 Course Project

Forecasting and Comparing time series of Frequency of Road Accidents in various regions

Devyanshi Singh
180237

Dhruvil Sangani
180243

Rishabh Gupta
180607

Utkarsh Gupta
180837

Yash Gidwani
180885

Course Instructor: Prof. Amit Mitra

1. Introduction

Road traffic accidents (RTA) are quite a significant contributor to the mortality rates, and usually, this issue receives inadequate attention. With the use of statistical methods & Machine Learning models, it is possible to predict the future occurrence of road traffic accidents using the available data. The frequency of accidents mapped to the corresponding time points acts as a time series, and the analysis, if done accurately, can help us draw immensely useful observations and predictions from it.

Forecasting road traffic accidents is useful to monitor the effectiveness of various road safety policies. Predictive models are very useful for identifying various related factors of road traffic accidental deaths. One of the most effective methods of forecasting future events is time series analysis. Amongst the several methodologies present for time series analysis, in this project, we have incorporated **Autoregressive Integrated Moving Average (ARIMA) model** and the **Random Forest Supervised Learning Algorithm** for making predictions on the series, comparing their respective accuracies, and drawing qualitative and statistical conclusions.

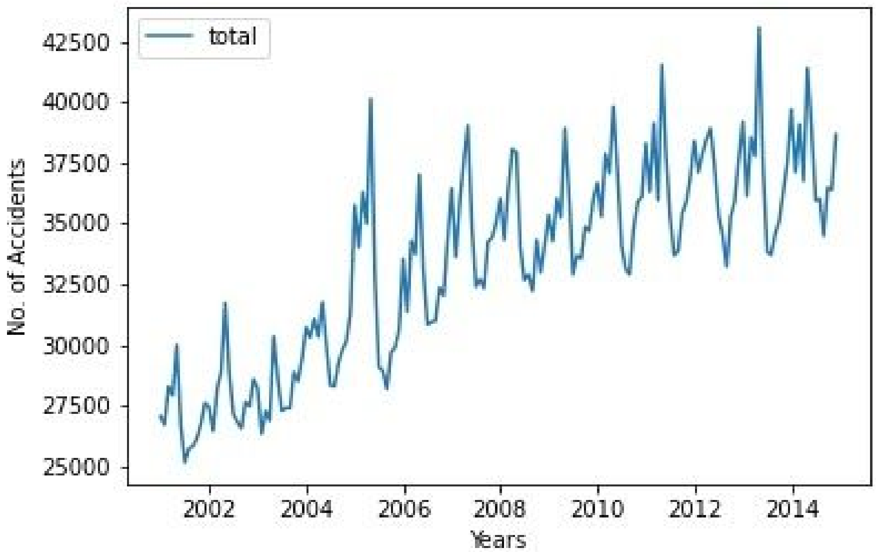
1.1 Dataset:

In our time series analysis, we incorporated the state-wise monthly data of road accidents in India from January 2001 to December 2014.
Link: [Road_Accidents_India_2001-2014]
Below is a snippet of how the original dataset looks like. We made modifications to it as per necessity while analyzing region-wise statistics, or the country's statistics as a whole.

STATE/UT	YEAR	JANUARY	FEBRUARY	MARCH	APRIL	MAY	JUNE	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER	TOTAL
A & N Islands	2001	8	23	15	15	14	19	14	19	7	12	13	22	181
A & N Islands	2002	12	10	14	16	10	7	16	11	23	21	11	17	168
A & N Islands	2003	19	13	15	13	13	12	8	16	17	25	14	15	180
A & N Islands	2004	21	14	22	17	13	18	16	19	16	20	15	24	215
A & N Islands	2005	19	21	22	17	13	19	21	14	15	19	10	16	206
A & N Islands	2006	21	13	4	22	9	14	12	14	8	14	6	18	155
A & N Islands	2007	17	16	12	22	12	14	8	10	11	7	11	12	152
A & N Islands	2008	17	22	15	16	15	17	13	11	13	17	11	24	191
A & N Islands	2009	16	23	23	21	21	19	24	25	31	22	20	26	271
A & N Islands	2010	16	30	28	15	29	24	22	18	25	30	27	21	285
A & N Islands	2011	24	10	19	24	13	28	17	18	25	17	18	22	235
A & N Islands	2012	25	15	24	24	18	16	17	18	18	25	17	19	236
A & N Islands	2013	24	23	16	15	13	16	14	25	14	15	14	11	200
A & N Islands	2014	25	13	19	19	18	15	15	16	15	23	18	22	218
Andhra Pradesh	2001	2204	2437	2405	2351	2550	2284	2025	2077	2070	2276	2122	2387	27188
Andhra Pradesh	2002	2492	2453	2835	2786	3195	2880	2645	2607	2555	2624	2646	2859	32577
Andhra Pradesh	2003	2783	2569	2870	2635	3265	2924	2857	2934	2787	2881	3037	3215	34537
Andhra Pradesh	2004	3019	3131	3211	3100	3257	2942	2827	3079	2972	3041	3129	3370	37078
Andhra Pradesh	2005	3189	3193	3182	3056	3612	3247	2907	3028	2742	2928	2975	3230	37289
Andhra Pradesh	2006	3568	3224	3496	3634	3962	3400	3334	3311	3232	3306	3268	3588	41323
Andhra Pradesh	2007	3978	3530	3728	3842	4099	3594	3519	3348	3246	3447	3617	3646	43594
Andhra Pradesh	2008	3594	3468	3848	3967	3811	3391	3260	3324	3169	3352	3319	3603	42106
Andhra Pradesh	2009	3682	3494	3775	3450	4048	3763	3412	3488	3017	3253	3298	3331	42011
Andhra Pradesh	2010	3515	3434	3749	3857	3960	3765	3206	3416	3115	8439	3397	3575	42428
Andhra Pradesh	2011	3540	3195	3584	3396	3916	3793	3237	3106	3067	3398	3442	3392	41066
Andhra Pradesh	2012	3347	3390	3693	3589	3250	3187	3160	3177	2893	3205	2985	3468	39344
Andhra Pradesh	2013	3732	3482	3715	3648	4531	3736	3018	3465	3450	3137	3432	3702	43048
Andhra Pradesh	2014	3809	3657	3641	3582	3986	3664	3167	3587	3225	3410	3346	4158	43232
Arunachal Pradesh	2001	19	17	21	16	19	14	21	14	23	17	30	25	236
Arunachal Pradesh	2002	25	16	21	23	16	19	11	17	18	21	25	23	235
Arunachal Pradesh	2003	16	24	22	13	18	17	19	21	23	18	16	22	229
Arunachal Pradesh	2004	23	24	23	11	15	11	14	8	21	29	17	21	217
Arunachal Pradesh	2005	26	14	29	12	17	20	20	14	19	21	24	21	237
Arunachal Pradesh	2006	14	20	17	19	20	31	11	25	18	25	21	22	243
Arunachal Pradesh	2007	22	20	26	19	25	19	14	15	22	18	11	19	230

2. Approach I: Forecasting Time Series by fitting ARIMA model

Starting with the dataset represented in the snippet before, we transformed it by summing up the number of road accidents for each state, resulting in the monthly accident statistics for the whole country. Below is the plot representing the country's total accidents corresponding to each month in the 14 years(2001-14).



2.1 Initial Observations:

- **Test for Stationarity using Augmented Dicky Fuller Test:**
  The Augmented Dickey-Fuller is a type of statistical test called a **unit root test**. The intuition behind a unit root test is that it determines how strongly a time series is defined by a trend. It uses an autoregressive model and optimizes an information criterion across multiple different lag values.
  - **Null Hypothesis (H0):** Will have a unit root. Hence, it is non-stationary, i.e. it has some time dependent structure.
  - **Alternate Hypothesis (H1):** It is stationary. It has a root>1

  We interpret this result using the p-value from this test. A p-value below a threshold(such as 5% or strongly 1%) suggests we reject the null hypothesis (stationary), otherwise a p-value above the threshold suggests we fail to reject the null hypothesis (non-stationarity).

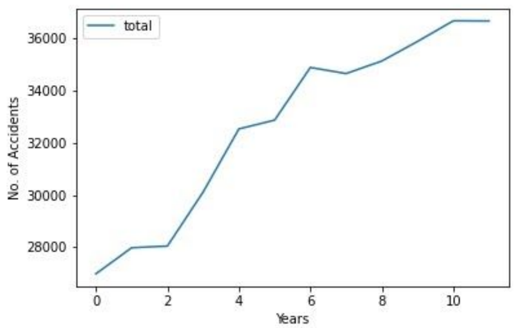
  We observed an initial p-value of 0.531 suggesting that the null hypothesis holds, and the time series is **not stationary**.

```
ADF Statistic: -1.503842
p-value: 0.531610
Critical Values:
        1%: -3.482
        5%: -2.884
        10%: -2.579
```

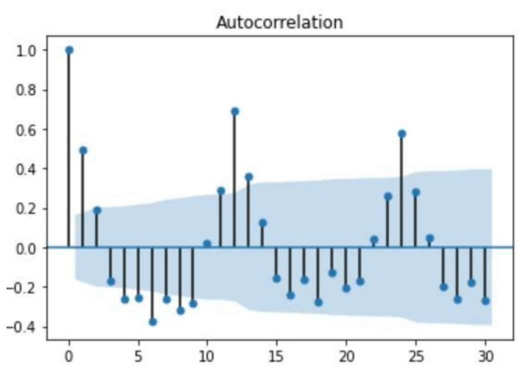
- We also observed a strong seasonality component in the series, with the frequency of accidents peaking and dipping in some particular months every year. Additionaly, the time trend seems somewhat linear, and varying quite slowly through the years. For this reason, while calculating the seasonal component, we used the slow time trend method.

2.2 Decomposing the Time Series:

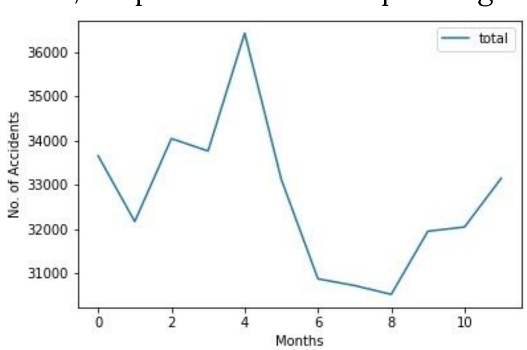
- **Trend Component:** Assuming the trend to be slow-moving, we estimate the trend component of the series. Later, subtracting this trend from the series would enable us to extract the seasonality component.



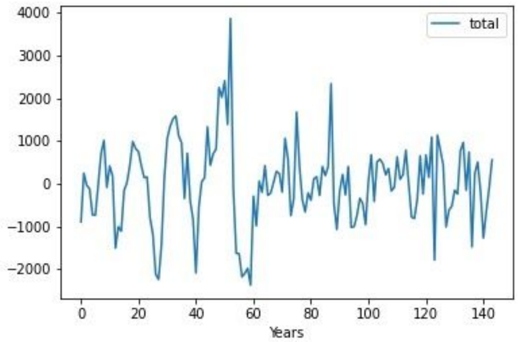
- **Seasonal Component:** To estimate the seasonal component in the time series, we plotted the ACF plot for the detrended series. Autocorrelation refers to how correlated a time series is with its past values. We observe oscillations in the ACF plot, with the pattern repeating every **12 months**.



Then, we plotted the corresponding seasonal component wrt to time shown below:



- **Residual Component:** We obtain the residual component by subtracting the trend and seasonality components from the original series.
  Residual Component = Original Series - Trend Component - Seasonal Component



On applying the Augmented Dickey-Fuller test on the residual component, we observed that the null hypothesis can be rejected, and the residual component is stationary.

```
ADF Statistic: -5.974721
p-value: 0.000000
Critical Values:
        1%: -3.479
        5%: -2.883
        10%: -2.578
```

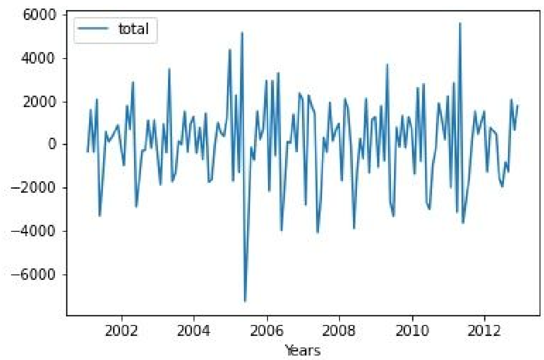
2.3 Tuning ARIMA Parameters:

An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends.
An ARIMA model can be understood by outlining each of its components as follows:

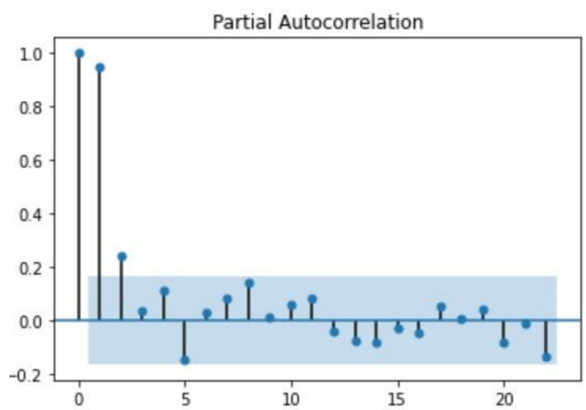
- **Autoregression (AR)** refers to a model that shows a changing variable that regresses on its own lagged, or prior, values.
- **Integrated (I)** represents the differencing of raw observations to allow for the time series to become stationary, i.e., data values are replaced by the difference between the data values and the previous values.
- **Moving average (MA)** incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

For an ARIMA model to be fitted to this component we would need to tune 3 parameters **(p,d,q)** to minimize the error after comparing forecasted data to our original data. While fitting our time series to the ARIMA model, we had two options: fitting the residual series to ARIMA taking d=0, or fitting just the deseasonalized series into ARIMA and determining the value of d with differencing method. We experimented with both the methods, and the latter worked better in our case.

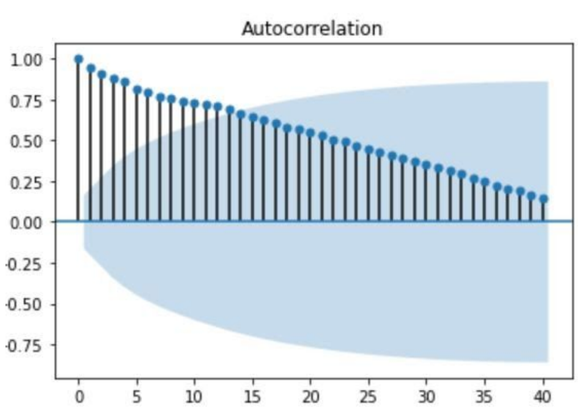
Determining d parameter: We performed first order differencing on the original time series and observed that the trend became linear(depicted below). Thus we conclude **d = 1**.



Determining p parameter: To estimate p, we plotted the PACF. A partial autocorrelation is the amount of correlation between a variable and a lag of itself that is not explained by correlations at all lower-order-lags. we can determine how many AR terms, we need to use to explain the autocorrelation pattern in a time series: if the PACF "cuts off" at lag k , then this suggests that we should try fitting an autoregressive model of order k.
We observed that the spikes cut-off after 2 or 3 spikes. So we experiment with **p = 2 or 3**.

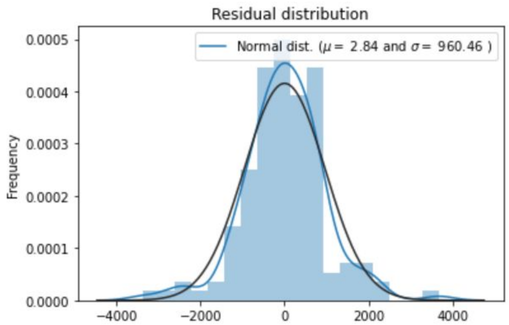


Determining q parameter: To estimate q, we plotted the ACF and ovserved that it continuously decayed. This observation leads us to the conclusion that the series has no MA component. Thus we take **q = 0**.



2.4 Model Verification

A model is performing well if the residuals are uncorrelated and follows normal distribution. We checked both those things and the results were the following:



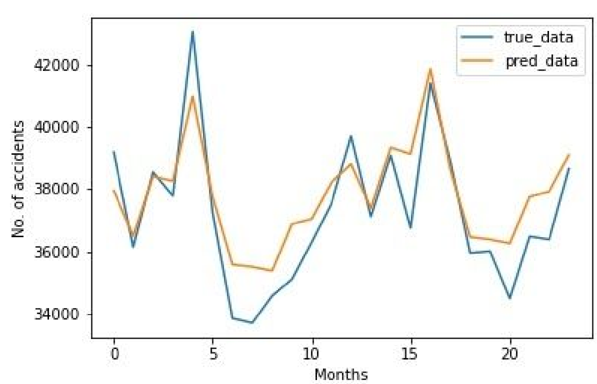
The above depicted graphs verify our ARIMA model fitting.

2.5 Forecasting with ARIMA(p,d,q):

We have the optimal values for p as 2 or 3, d as 1, and q as 0. With p=2, the Mean Absolute Error comes out to be approximately 2.87% and AIC = 2381.26. With p=3, the **Mean Absolute Error** comes out to be approximately **2.55%** and AIC = 2379.92. Therefore, **p = 3** is the more appropriate parameter.

Error above was calculated after forecasting the time series for 24 months and then comparing the forecasted time series and the original time series. The comparison between true data and the predicted data can be seen through this plot:

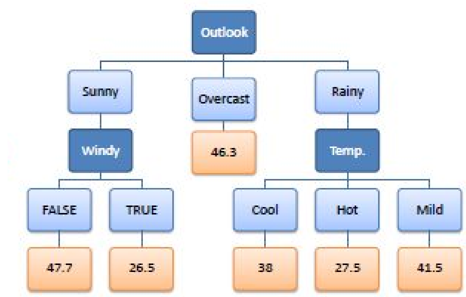
	true_data	pred_data
0	39185.0	38061.704906
1	36137.0	36674.505568
2	38543.0	38629.522431
3	37782.0	38419.804953
4	43064.0	41158.672920
5	37249.0	37927.029765
6	33847.0	35756.398904
7	33698.0	35679.653572
8	34565.0	35554.730682
9	35103.0	37056.974433
10	36297.0	37226.635528
11	37531.0	38402.213094
12	39699.0	38986.290675
13	37115.0	37576.618264
14	39085.0	39525.945849
15	36747.0	39319.273435
16	41404.0	42057.601021
17	38909.0	38825.928607
18	35946.0	36655.339527
19	36003.0	36578.583779
20	34485.0	36453.661365
21	36478.0	37955.905618
22	36378.0	38125.566537
23	38649.0	39301.144123



3. Approach II: Forecasting Time Series using Supervised Learning (Random Forest Algorithm)

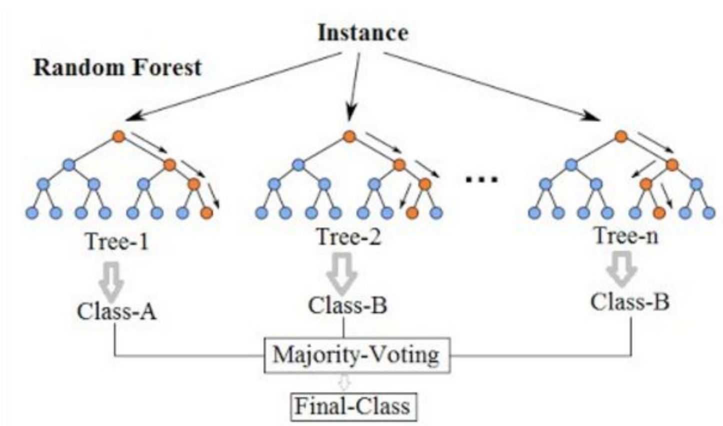
3.1 What is Random Forest?

Random forests or random decision forests are an **Ensemble Learning** method for Classification, **Regression** and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees.



Decision Trees for the most fundamental units of Random Forests. A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from the root to leaf represent classification rules.

A random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.



3.2 Why Random Forest?

The dataset that we had at hand consisted of the monthly statistics of road accidents spanned across 14 years, i.e. just **168 data points**. Supervised Learning Algorithms that work on neural networks usually require large numbers of training data points in order to make accurate predictions. However, algorithms with decision trees as their fundamental units prove to be more efficient in terms of **training time** and **accuracy** for **smaller datasets** too. Thus we went for Random Forest which is one of the most efficient Ensemble ML algorithms working with decision trees.

3.3 Converting the Time Series to a Supervised Learning Problem

Given a sequence of numbers for a time series dataset, we can restructure the data to look like a supervised learning problem. We can do this by using **previous time steps** as **input** variables and use the **next time step** as the **output variable**.

Using the shift( ) function in Pandas, we transformed the original time series dataset to the required Supervised Learning format as follows:

Year	t
2001.000000	27039
2001.083333	26694
2001.166667	28284
2001.250000	27921
2001.333333	29978

Year	t	t-1
2001.000000	27039	NaN
2001.083333	26694	27039.0
2001.166667	28284	26694.0
2001.250000	27921	28284.0
2001.333333	29978	27921.0

Fitting and Evaluating the Model

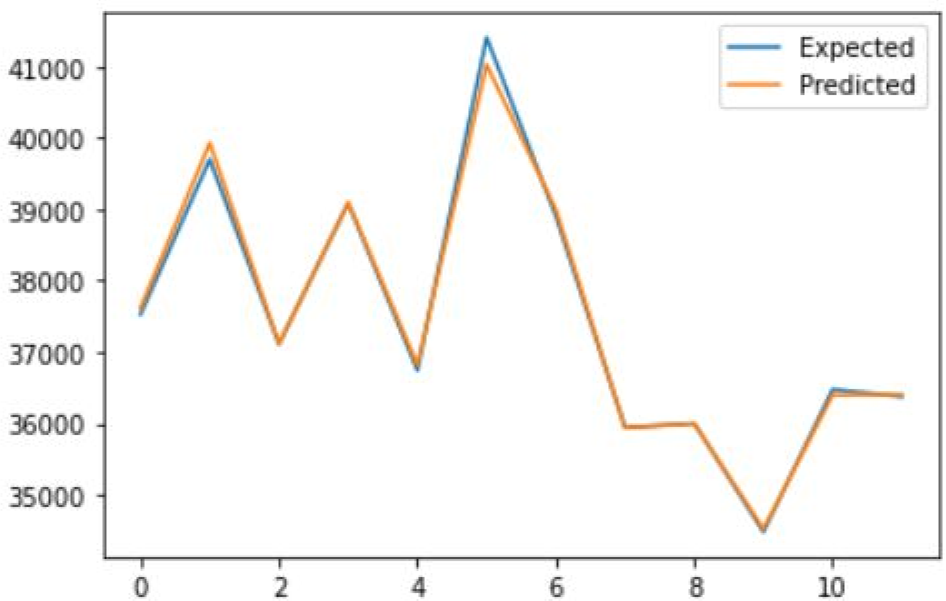
Unlike ARIMA model fitting, supervised learning algorithms for Time Series do not require us to decompose the time series before fitting. The Random Forest takes into account both the deterministic and the random components while training.

Once the dataset was prepared, we had to take care of certain aspects while fitting and evaluating the model. For instance, it would not be valid to fit the model on data from the future and have it predict the past. The model must be trained on the past and predict the future. Thus, the methods that randomize the data while evaluating, like k-fold cross-validation, were not used. Instead, we went for a technique called **Walk-forward Validation.**

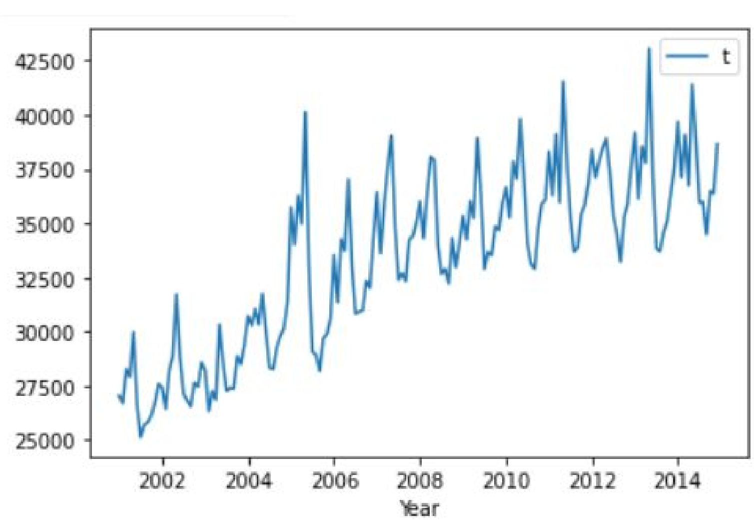
In walk-forward validation, the dataset is first **split into train and test** sets by selecting a cut point. In our case, all data except the **last 12 months** was used for training and the last 12 months is used for testing. We used **1000 decision trees** in the ensemble to prevent underlearning.

The snippet below is the expected(original) time series value v/s the values predicted by the Random Forest trained model. The **Mean Absolute Error** comes out to be 83.623, which is approximately **0.251 percent**. These predictions are indeed substantially better than the previous ARIMA fitting.

```
>expected=37531.0, predicted=37621.0
>expected=39699.0, predicted=39908.8
>expected=37115.0, predicted=37111.6
>expected=39085.0, predicted=39081.9
>expected=36747.0, predicted=36833.5
>expected=41404.0, predicted=41043.6
>expected=38909.0, predicted=38998.6
>expected=35946.0, predicted=35941.0
>expected=36003.0, predicted=36005.4
>expected=34485.0, predicted=34525.6
>expected=36478.0, predicted=36403.9
>expected=36378.0, predicted=36416.8
Mean Absolute Error: 83.623
Percent error: 0.251
```



4. Subjective Observations on the Time Series:

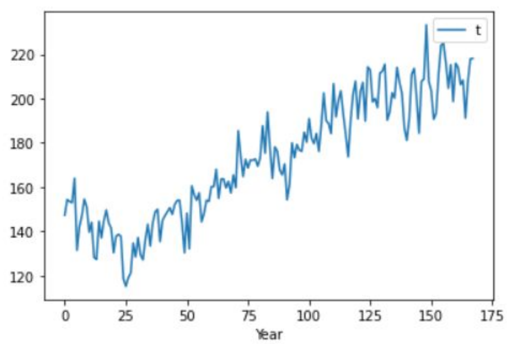


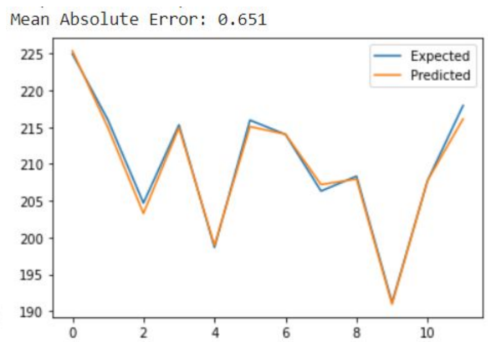
It is observed that there is a general linear increasing trend in the frequency of road accidents from 2001-2014. The reason can be a similar increasing trend in sales of vehicles in India. Also by seeing the above seasonality plot, we observed that there is a spike in the month of May-April and a trough in August-September almost every year.

5. Forecasting Road Accidents of various regions in India using Random Forest:

We subdivided the Road Accidents time series dataset for India into **six consequent regions** and performed forecasting on each using the Random Forest Algorithm. Here we present the time series graphs plotted for each region for some qualitative analysis of the relative accident in these regions, and the corresponding fitted models, showing expected v/s predicted values, for the six regions. The values have been scaled down by dividing the figures by the population percentage of the region so as to get uniform observations.

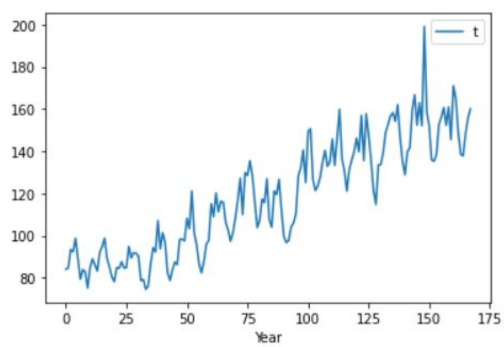
5.1 North Region(Jammu & Kashmir, Punjab, HP, Haryana, UP, Uttarakhand, Delhi):

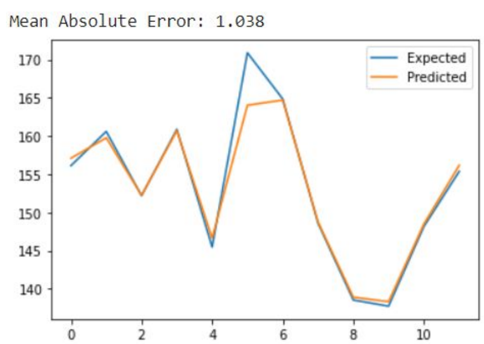




There was a dip observed in accidents around 2002-03 in the north zone, probably due to the government restrictions caused by the riots.

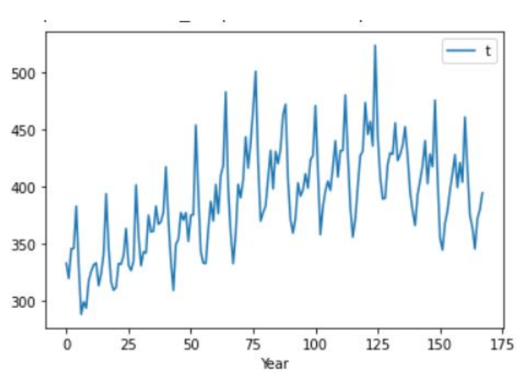
5.2 East Region(Bihar, Odisha, Jharkhand, West Bengal)

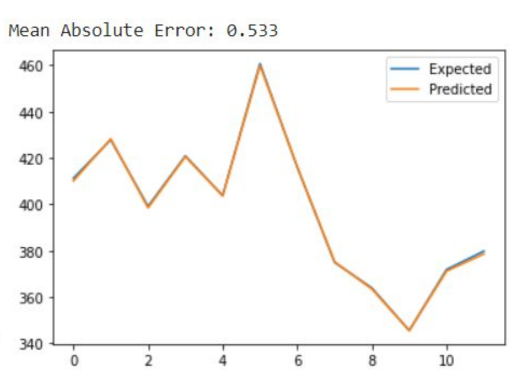




This is the general increasing trend observed ranging from 80-100 to 180-200 (scaled-down).

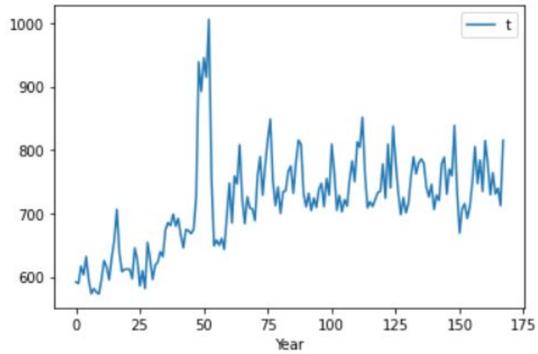
5.3 West Region(Rajasthan, Gujarat, Goa, Maharashtra)

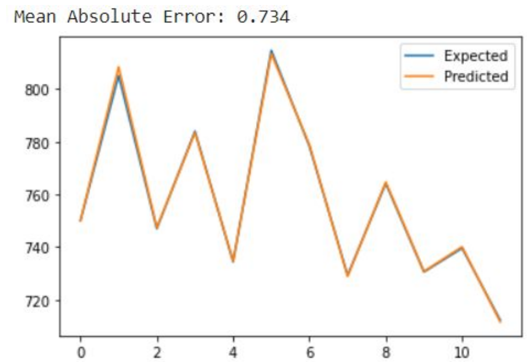




The frequency of accidents was quite high from 2001-08 but due to road traffic regulations, the number dropped towards the end.

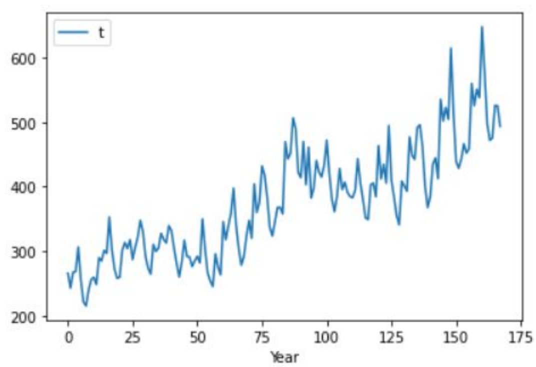
5.4 South Region(Andhra Pradesh, Karnataka, Kerala, Tamil Nadu, Telangana)

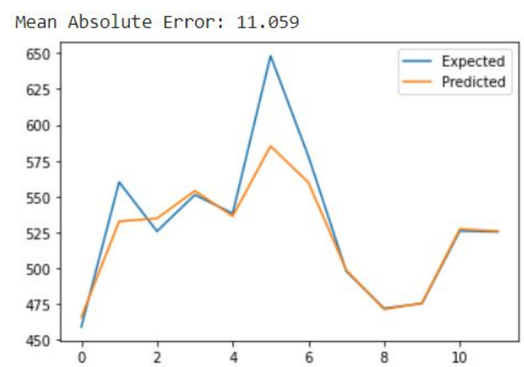




The general trend is increasing, however there was a sharp peak in the number of accidents in 2004-05.

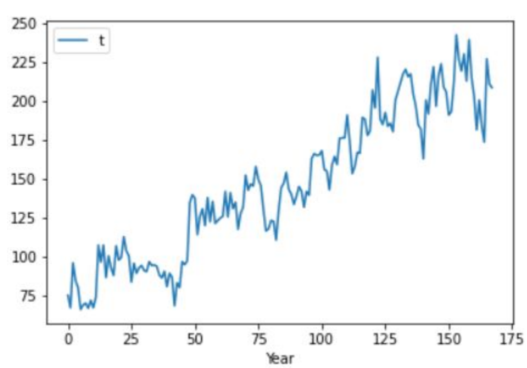
5.5 Central Region(MP, Chhattisgarh)

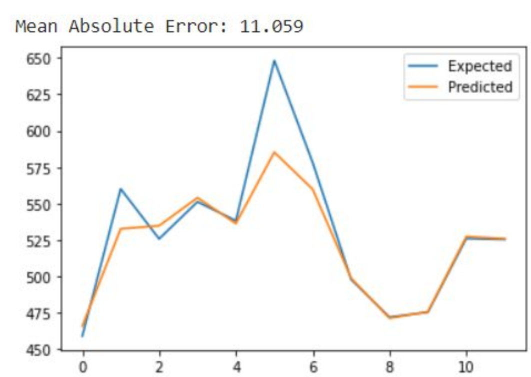




The general trend is increasing, it is one of the higher accidents prone region in India.

5.6 North-East Region(Assam, Sikkim, Nagaland, Meghalaya, Mizoram, Tripura, AP, Manipur)





There are relatively lesser accidents in these areas, due to less population density.

6. Conclusion

In this project, we attempted to forecast the time series corresponding to the number of road accidents in India through ARIMA and Random Forest model fitting. While we obtained a mean absolute errror of 2.55 percent in case of ARIMA, the MAE for Random Forest predictions came out to be 0.251 percent. We also qualitatively analyzed the pattern for 14 years of data for the country as a whole, and its various regions.

Even though it is not possible to make completely accurate forecasts on the accident rates in the future, time series models like these certainly assist us in taking more informed decisions while determining needed policies and regulations. We can further experiment with a larger dataset and more algorithms and models to develop more accurate models.

7. Acknowledgement

We would like to extend our gratitude towards our course instructor, Prof. Amit Mitra for giving us an opportunity to work on this project. Working on a real-life problem in the form of time series analysis further stregthened the theoretical concepts we developed during the course, and broadened our perspective towards analyzing and possibly solving similar problems encountered in our day-to-day lives.