A REPORT

ON

Forecasting unemployment among graduates from different Fields of study

BY

Utkarsh Agarwal **2016B3A70581P**

IN FULFILMENT OF THE COURSE

Study Project (ECON F266)

SUBMITTED TO

Dr. Satyendra K Sharma

Department of Management

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

Table of Contents

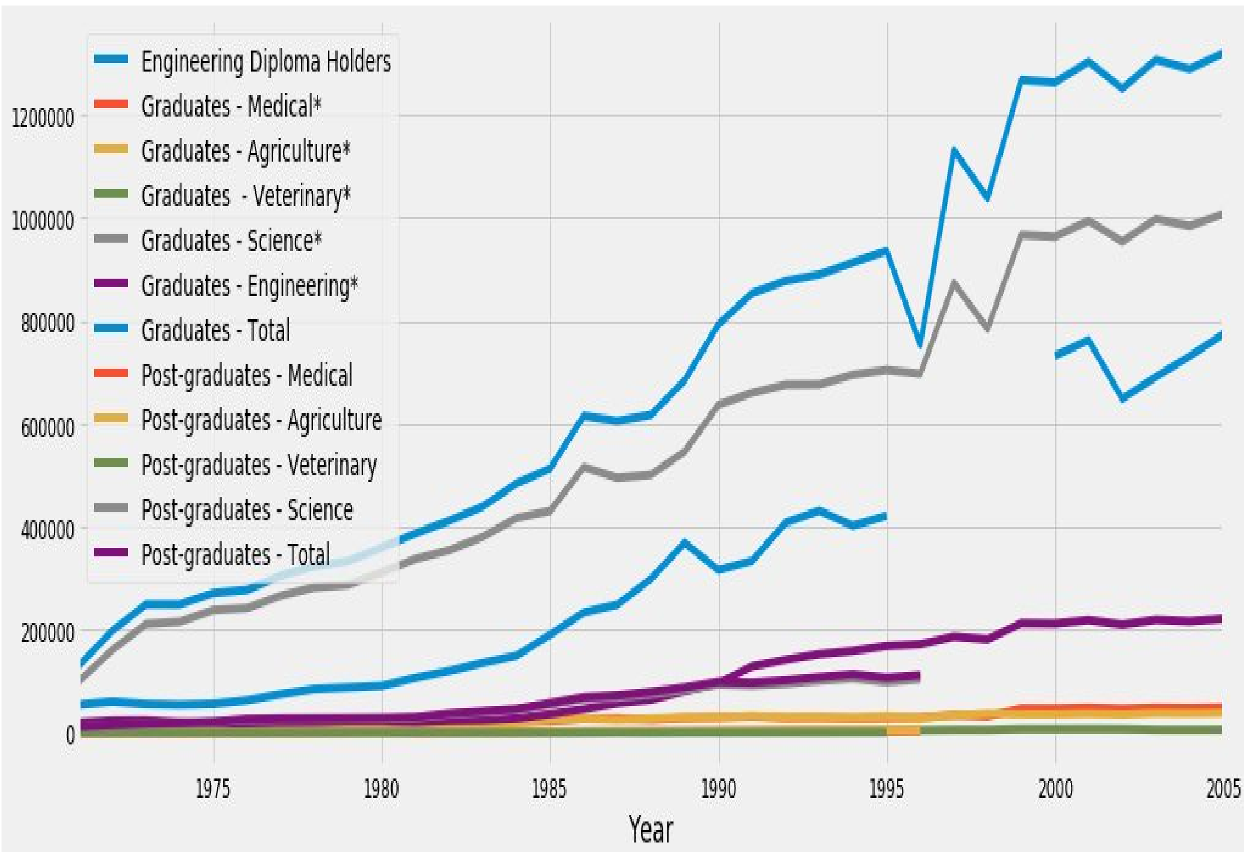
The Data

We are using data on the number of job seekers on the live registers on the employment exchanges by various Science & Technology fields. The data are taken from the publication - **Analysis of Budgeted Expenditure on Education, Department of Education, MHRD**
We have a good 35 years unemployed unemployed graduates data.

Year	Engineering Diploma Holders	Graduates - Medical*	Graduates - Agriculture*	Graduates - Veterinary*	Graduates - Science*	Graduates - Engineering*	Graduates - Total
1971	54056	3848	7325	361	99189	19050	129773
1972	59847	5127	9092	200	160644	22808	197871
1973	55215	5664	8913	371	210716	23093	248757
1974	53901	6682	7370	376	215089	19344	248861
1975	55564	7301	7958	511	237607	17316	270693
1976	62447	8249	8285	489	241319	18385	276727
1977	74319	8948	9763	299	265656	19798	304464
1978	84317	10637	9765	399	281693	20113	322607
1979	87275	12923	10841	433	286639	21781	332617
1980	90306	14809	11375	356	310692	21862	359094
1981	106183	15536	13046	353	337190	20393	386518
1982	119345	17700	16075	578	353918	22982	411253
1983	135141	17607	17027	528	379931	23825	438918
1984	148985	20636	18938	661	416623	27044	483902
1985	189836	21800	24437	713	430095	35523	512568
1986	232965	26772	26677	864	515342	45226	614881
1987	248179	27233	24539	1613	494812	56992	605189
1988	298535	24781	26805	1138	500694	63538	616956
1989	368561	28051	28860	1619	545552	80198	684280
1990	315502	28988	29352	2067	636505	95563	792475
1991	333174	30827	31705	2503	659813	128422	853270
1992	407837	28080	29701	2826	675481	141225	877313
1993	430577	28560	29529	3328	676099	152015	889531
1994	401832	27446	28495	3566	695084	158509	913100
1995	421029	28207	31322	3887	704059	168066	935541
1996	NA	28600	27800	4400	696900	171400	753300
1997	NA	33700	33000	5100	873100	186100	1131000
1998	553000	31310	36600	5000	784000	180900	1037810
1999	NA	46600	35300	6800	966700	212200	1267600
2000	731700	46400	35200	6700	963300	211400	1263000
2001	762500	47900	36300	6900	993800	218200	1303100
2002	648400	46000	34900	6700	954000	209400	1251000
2003	690891	48000	37300	5300	997600	218800	1307000
2004	730944	47300	36800	5200	984000	215800	1289100
2005	773908	48500	37700	5400	1007400	220900	1319900

Time Series Analysis Introduction

Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values.
Time series are widely used for non-stationary data, like economic, weather, stock price, and retail in this post. We will work on different approaches for forecasting unemployed graduates in different fields time series.



Data Preprocessing

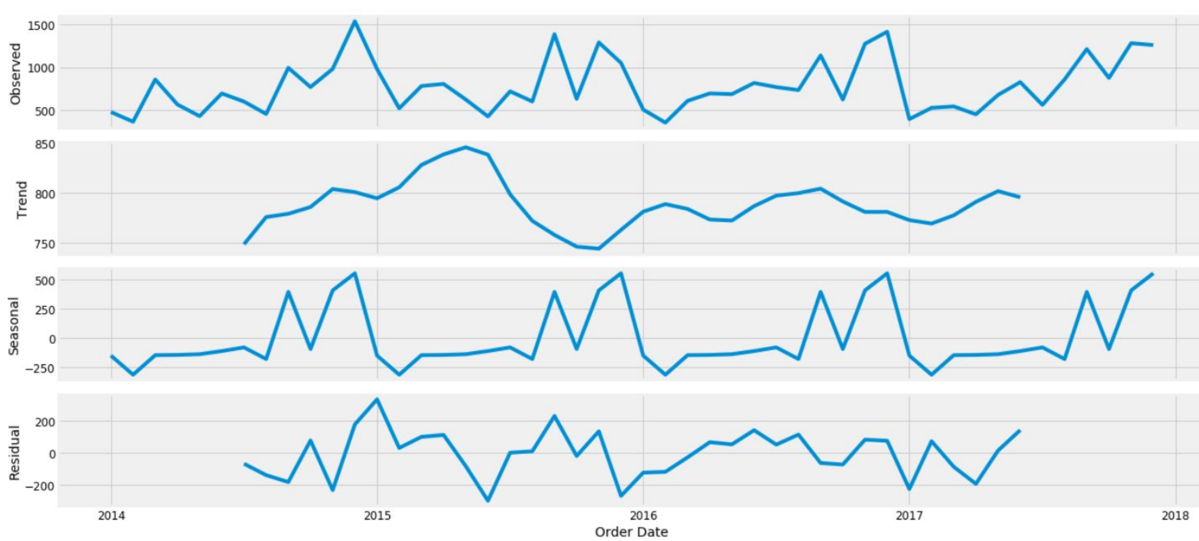
This step includes removing columns we do not need, check missing values, aggregate by date and so on.

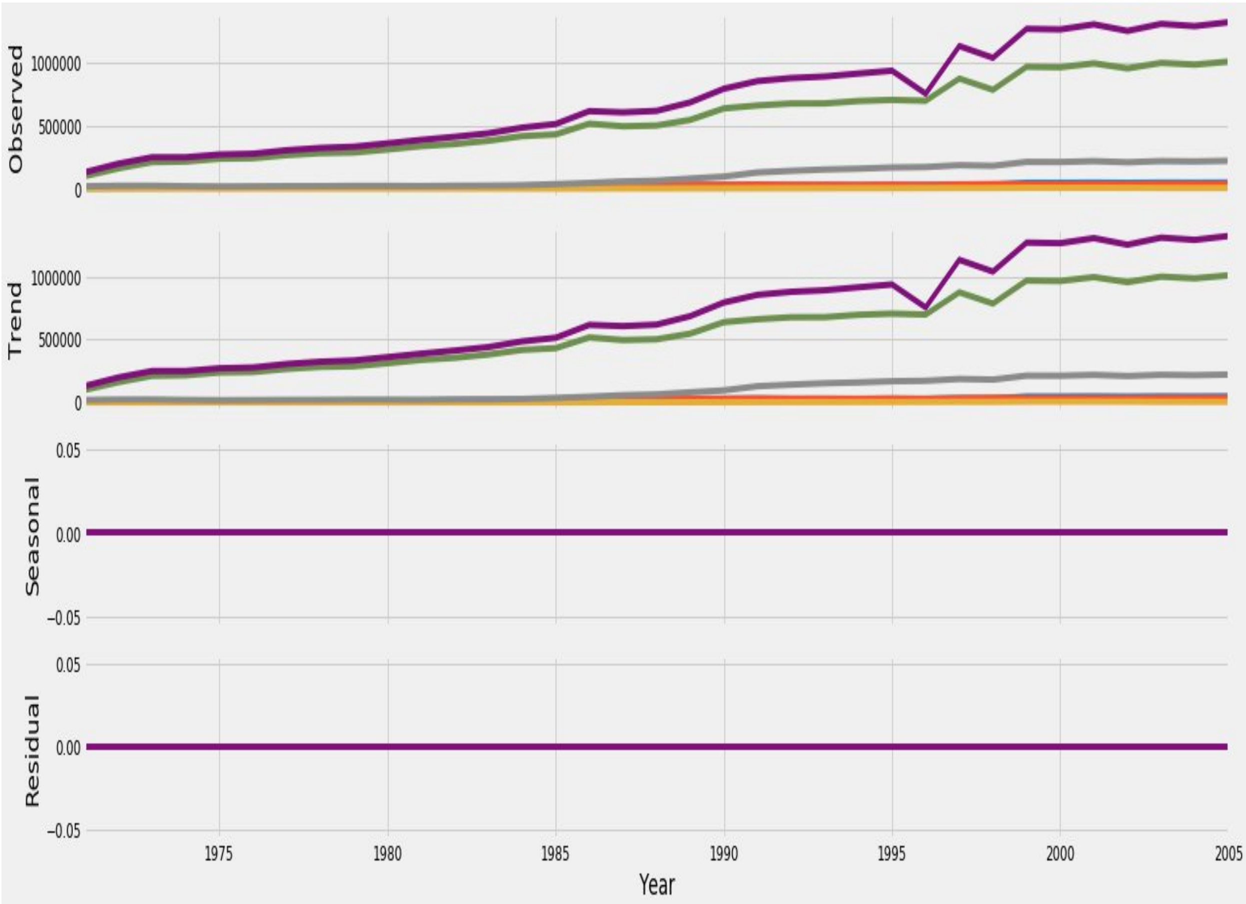
Indexing with Time Series Data

```
DatetimeIndex(['1971-01-01', '1972-01-01', '1973-01-01', '1974-01-01',
               '1975-01-01', '1976-01-01', '1977-01-01', '1978-01-01',
               '1979-01-01', '1980-01-01', '1981-01-01', '1982-01-01',
               '1983-01-01', '1984-01-01', '1985-01-01', '1986-01-01',
               '1987-01-01', '1988-01-01', '1989-01-01', '1990-01-01',
               '1991-01-01', '1992-01-01', '1993-01-01', '1994-01-01',
               '1995-01-01', '1996-01-01', '1997-01-01', '1998-01-01',
               '1999-01-01', '2000-01-01', '2001-01-01', '2002-01-01',
               '2003-01-01', '2004-01-01', '2005-01-01'],
              dtype='datetime64[ns]', name='Year', freq=None)
```

For indexing purposes, we assumed the date of each year unemployed graduates as the first day of the respective year.

Visualizing unemployed graduates -Time Series Data



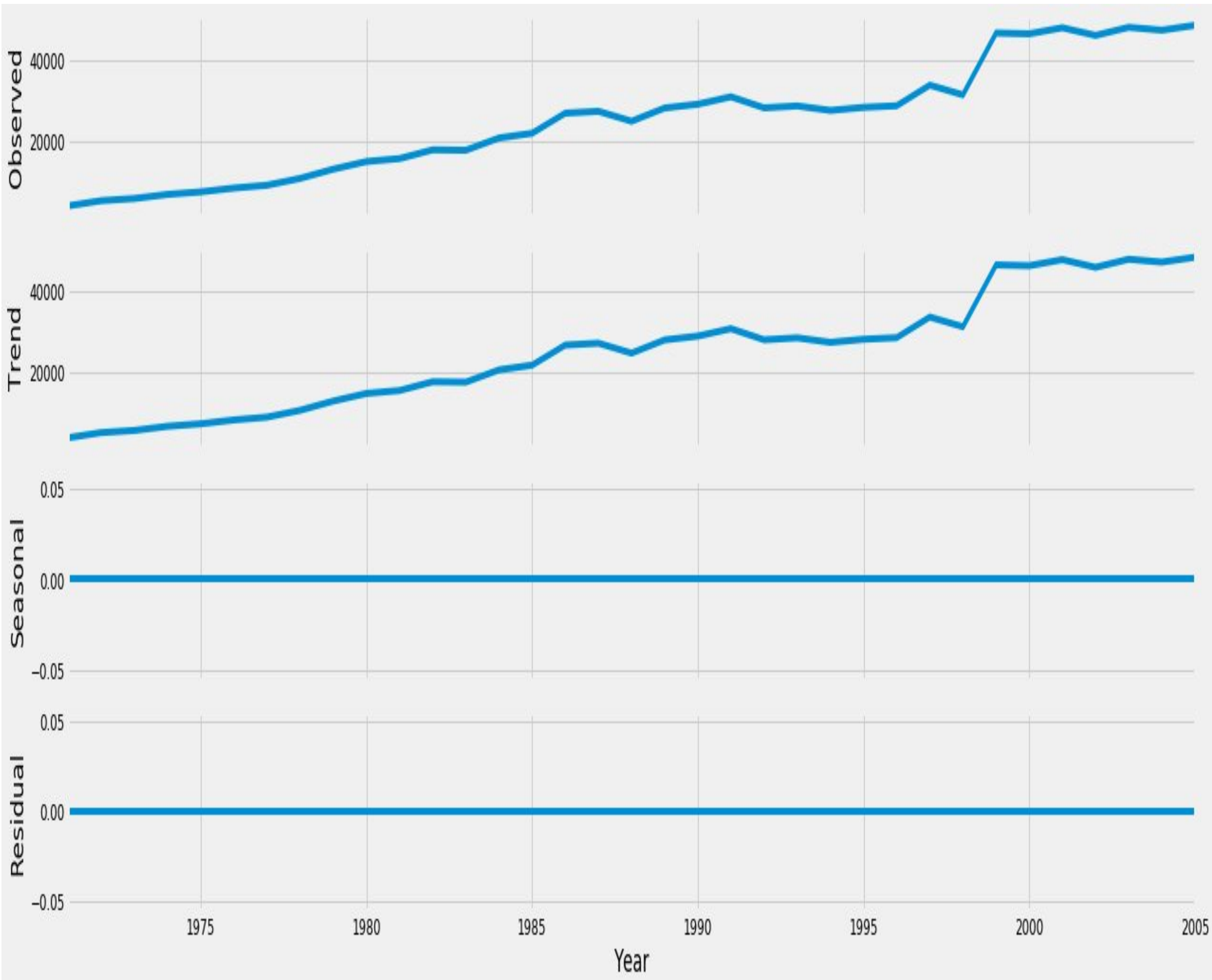


The above figure depicts different behavior. We can also visualize our data using a method called time-series decomposition that allows us to decompose our time series into three distinct components: trend, seasonality, and noise.

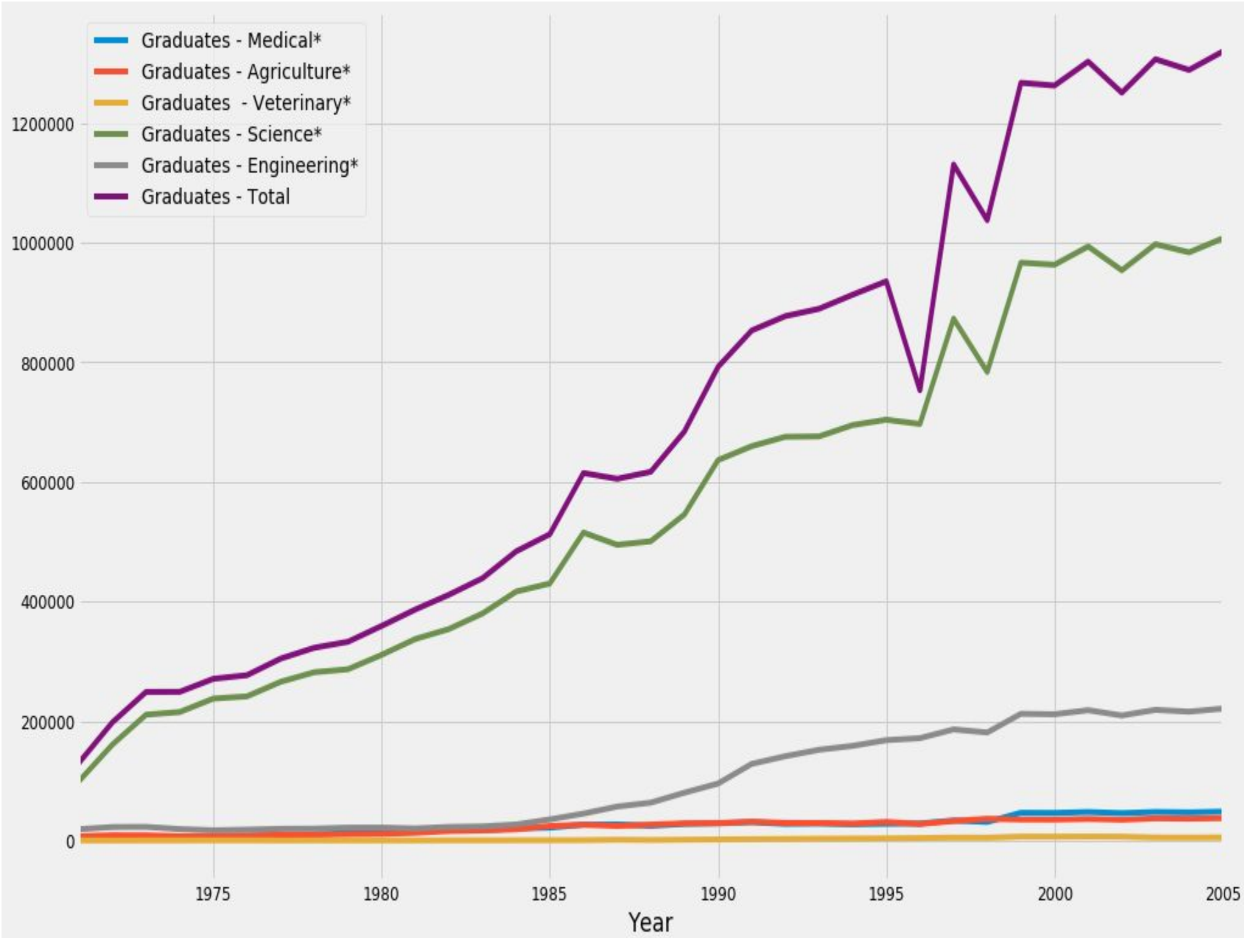
Some distinguishable patterns appear when we plot the data. The time-series has an upward trend between any year and the growth rate for the unemployed unemployed graduates are increasing rapidly.

The Analysis for our data for unemployed graduates:

Analysis of the breakdown component:



Analysis of the Combined effect:



Time series forecasting with ARIMA

We are going to apply for one of the most commonly used methods for time-series forecasting, known as ARIMA, which stands for **Autoregressive Integrated Moving Average**.
ARIMA models are denoted with the notation ARIMA(p, d, q). These three parameters account for seasonality, trend, and noise in data:

```
Examples of parameter combinations for Seasonal ARIMA...
SARIMAX: (0, 0, 1) x (0, 0, 1, 12)
SARIMAX: (0, 0, 1) x (0, 1, 0, 12)
SARIMAX: (0, 1, 0) x (0, 1, 1, 12)
SARIMAX: (0, 1, 0) x (1, 0, 0, 12)
```

```
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in
list(itertools.product(p, d, q))]


print('Examples of parameter combinations for Seasonal ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))
```

This step is parameter **Selection for our unemployed graduates's ARIMA Time Series Model. Our goal here is to use a "grid search"** to find the optimal set of parameters that yields the best performance for our model.

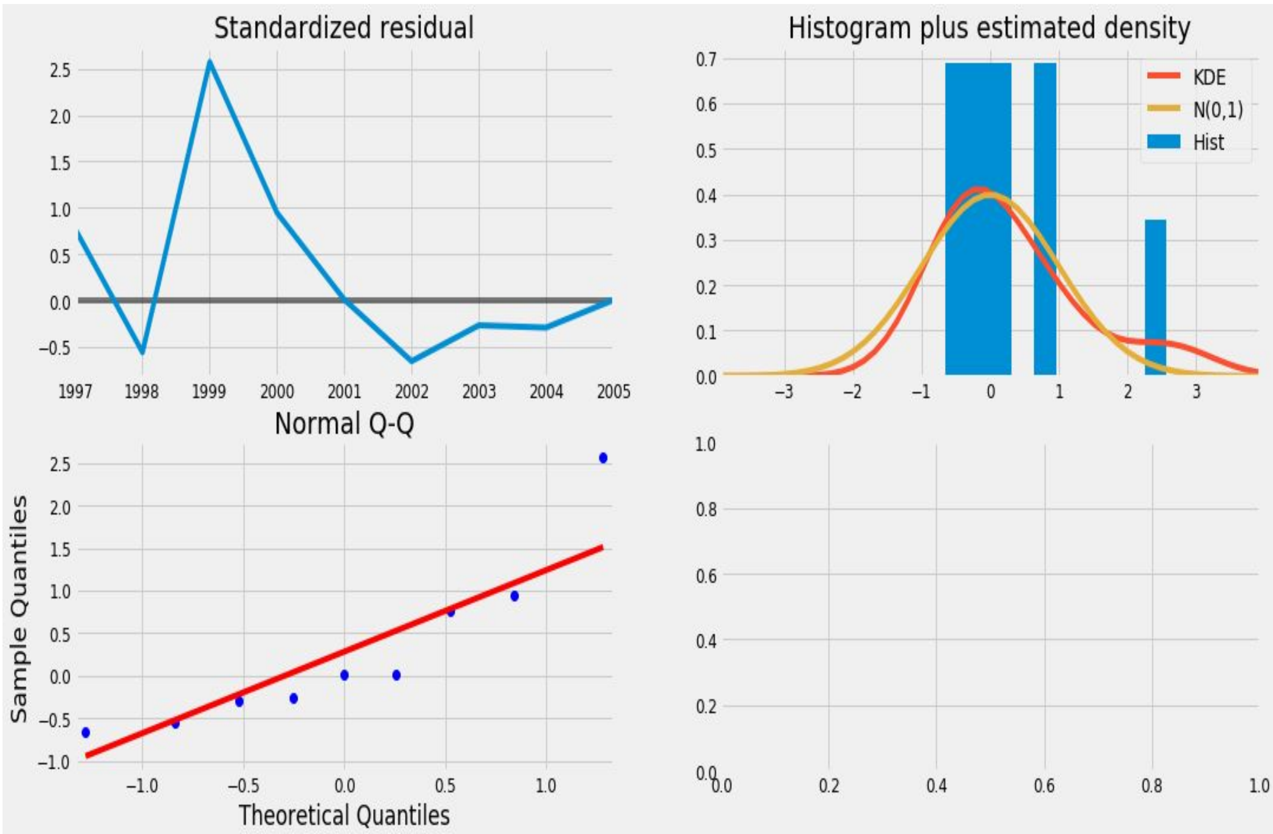
```
ARIMA(0, 0, 0)x(0, 0, 0, 12)12 - AIC:797.6083922553523
ARIMA(0, 0, 0)x(0, 1, 0, 12)12 - AIC:491.8509270035107
ARIMA(0, 0, 0)x(1, 0, 0, 12)12 - AIC:485.63120623339006
ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:226.87345341869212
ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:755.4436705970667
ARIMA(0, 0, 1)x(0, 1, 0, 12)12 - AIC:458.50807066980843
ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:523.9724363701845
ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:240.7670566777688
ARIMA(0, 1, 0)x(0, 0, 0, 12)12 - AIC:630.7347939331952
ARIMA(0, 1, 0)x(0, 1, 0, 12)12 - AIC:412.83451485404544
ARIMA(0, 1, 0)x(1, 0, 0, 12)12 - AIC:430.52289480525724
ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:203.4793800541981
ARIMA(0, 1, 1)x(0, 0, 0, 12)12 - AIC:614.4259345219376
ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:394.20645131166697
ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:432.0631171476646
ARIMA(0, 1, 1)x(1, 1, 0, 12)12 - AIC:205.05091416811905
ARIMA(1, 0, 0)x(0, 0, 0, 12)12 - AIC:647.3370346988787
ARIMA(1, 0, 0)x(0, 1, 0, 12)12 - AIC:433.4663130510298
ARIMA(1, 0, 0)x(1, 0, 0, 12)12 - AIC:432.77317108769165
ARIMA(1, 0, 0)x(1, 1, 0, 12)12 - AIC:207.94477937686477
ARIMA(1, 0, 1)x(0, 0, 0, 12)12 - AIC:629.304866847507
ARIMA(1, 0, 1)x(0, 1, 0, 12)12 - AIC:414.7075898171331
ARIMA(1, 0, 1)x(1, 0, 0, 12)12 - AIC:432.10971798342723
ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:205.63157760729592
ARIMA(1, 1, 0)x(0, 0, 0, 12)12 - AIC:632.2419880606586
ARIMA(1, 1, 0)x(0, 1, 0, 12)12 - AIC:412.68368817037924
ARIMA(1, 1, 0)x(1, 0, 0, 12)12 - AIC:413.54719959787536
ARIMA(1, 1, 0)x(1, 1, 0, 12)12 - AIC:185.2797115778119
ARIMA(1, 1, 1)x(0, 0, 0, 12)12 - AIC:613.611388600176
ARIMA(1, 1, 1)x(0, 1, 0, 12)12 - AIC:395.8057835897643
ARIMA(1, 1, 1)x(1, 0, 0, 12)12 - AIC:413.28784704340524
ARIMA(1, 1, 1)x(1, 1, 0, 12)12 - AIC:187.12874154766428
```

The above output suggests that SARIMAX ARIMA(1, 1, 0)x(1, 1, 0, 12)12 yields the lowest AIC value of 185.27. Therefore we should consider this to be optimal option.

Fitting the ARIMA model

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.0146	0.342	0.043	0.966	-0.655	0.684
ma.L1	-1.0000	0.360	-2.781	0.005	-1.705	-0.295
ar.S.L12	-0.0253	0.042	-0.609	0.543	-0.107	0.056
sigma2	2.958e+04	1.22e-05	2.43e+09	0.000	2.96e+04	2.96e+04

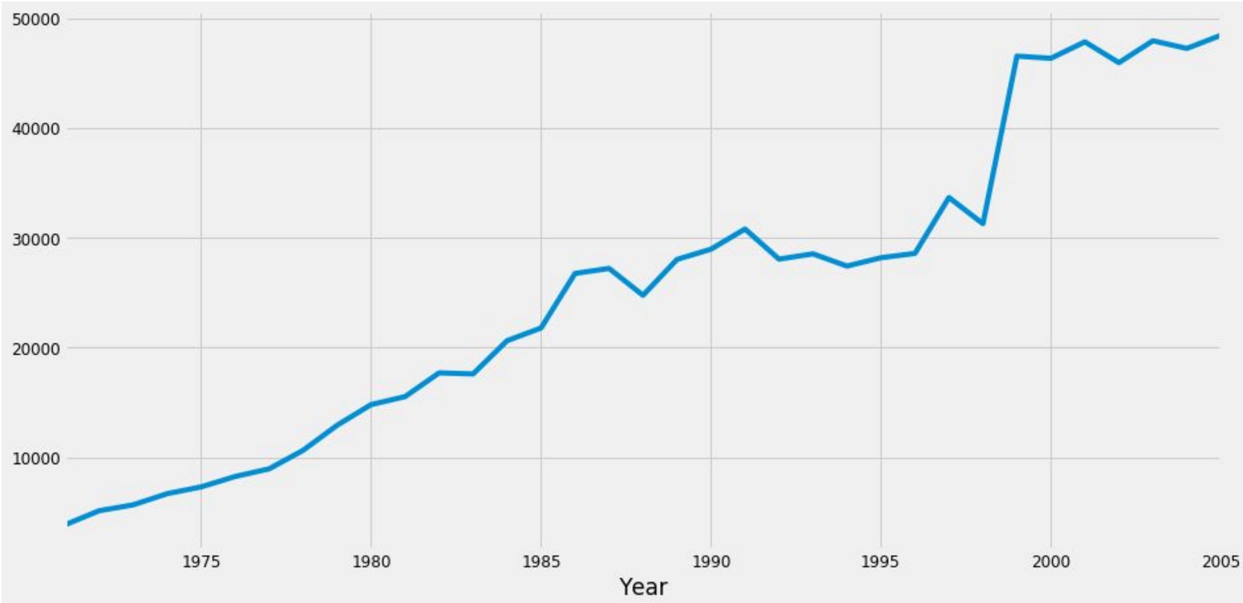
We should always run model diagnostics to investigate any unusual behavior.



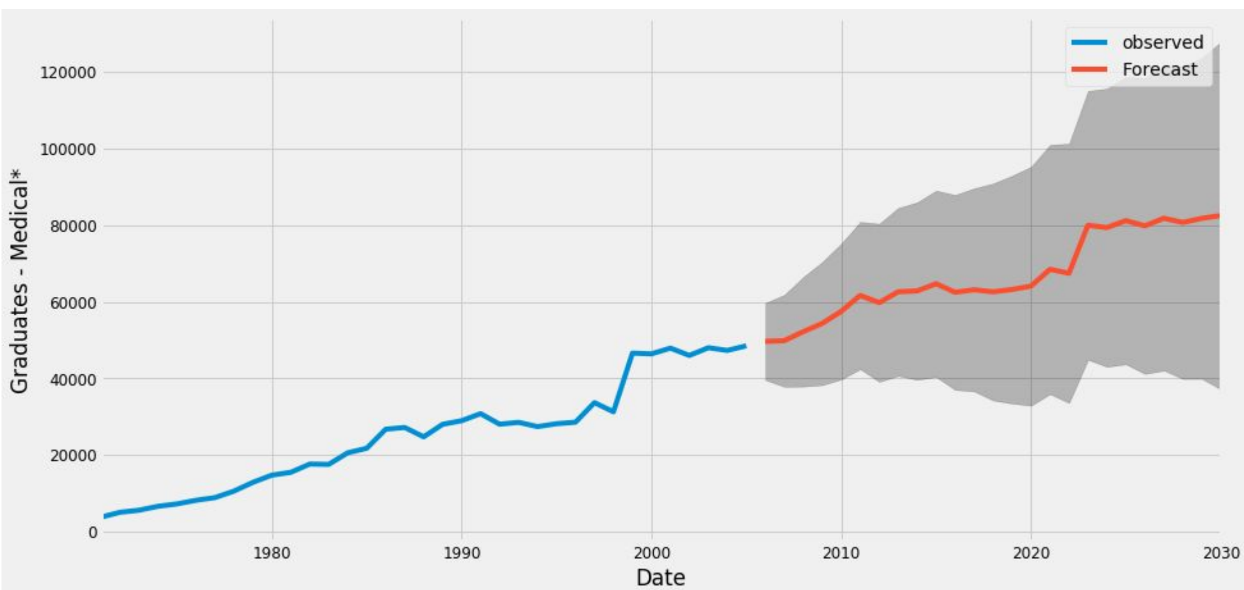
It is not perfect, however, our model diagnostics suggests that the model residuals are near normally distributed.

Validating forecasts

To help us understand the accuracy of our forecasts, we compare predicted to real of the time series, and we set forecasts to start at 2005 to the end of the data.



The line plot is showing the observed values compared to the rolling forecast predictions. Overall, our forecasts align with the true values very well, showing an upward trend starts from the beginning of the year and captured the seasonality toward the end of the year.



The Mean Squared Error of our forecasts is 1165.51

The Root Mean Squared Error of our forecasts is 34.14.

In statistics, **the mean squared error (MSE)** of an estimator measures the average of the squares of the errors — that is, the average squared difference between the estimated values and what is estimated. The MSE is a measure of the quality of an estimator — it is always non-negative, and the smaller the MSE, the closer we are to finding the line of best fit.

Root Mean Square Error (RMSE) tells us that our model was able to forecast yearly unemployed graduates in the test set within 34.14 of the real sales. Our unemployed graduates yearly ranges from around 40000 to over 80000. In my opinion, this is a pretty good model so far.

Code Snippet for the ARIMA model:

The below written code by me used for fitting the ARIMA model to the dataset. Complete 350 lines of code is available in a separate file, it only shows ARIMA model.

```
]: import warnings
   import itertools
   import numpy as np
   import matplotlib.pyplot as plt
   warnings.filterwarnings("ignore")
   plt.style.use('fivethirtyeight')
   import pandas as pd
   import statsmodels.api as sm
   import matplotlib
   import pandas as pd
   from datetime import datetime
```

```
]: import statsmodels.api as sm
```

```
]: matplotlib.rcParams['axes.labelsize'] = 14
   matplotlib.rcParams['xtick.labelsize'] = 12
   matplotlib.rcParams['ytick.labelsize'] = 12
   matplotlib.rcParams['text.color'] = 'k'
```

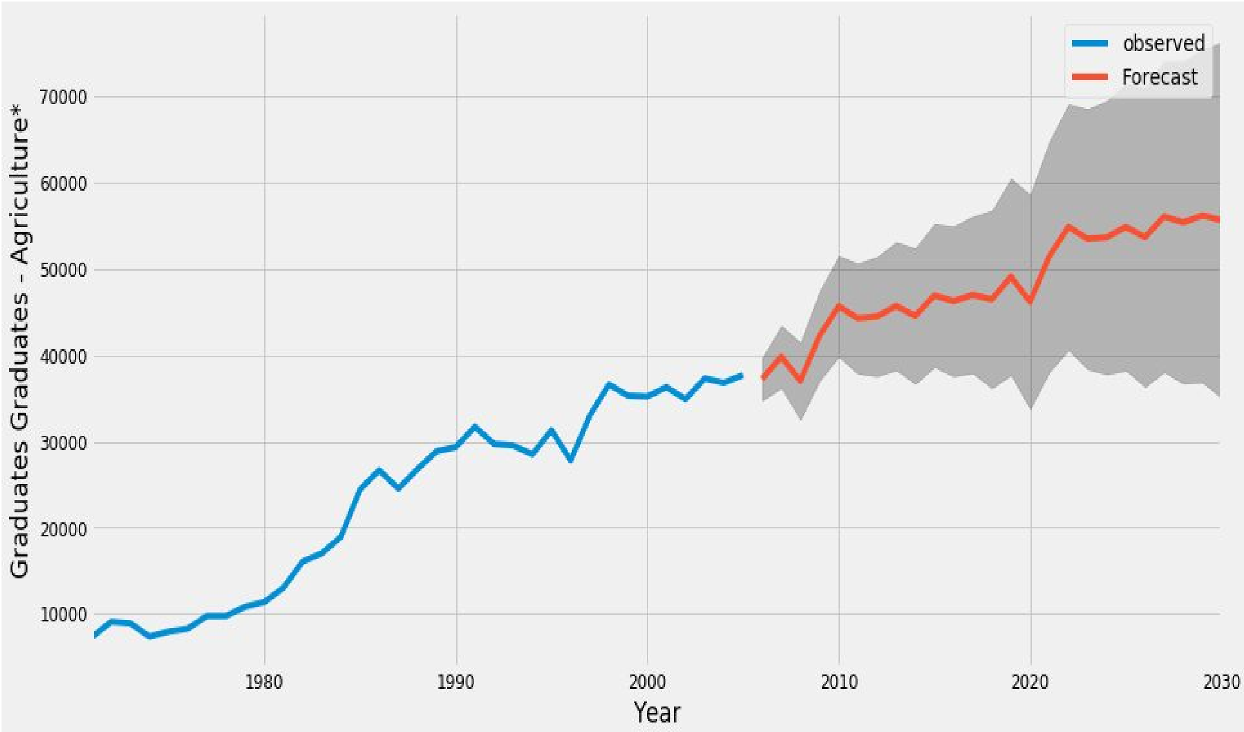
```
]: df = pd.read_csv("Downloads/data/Data_distribution.csv")
```

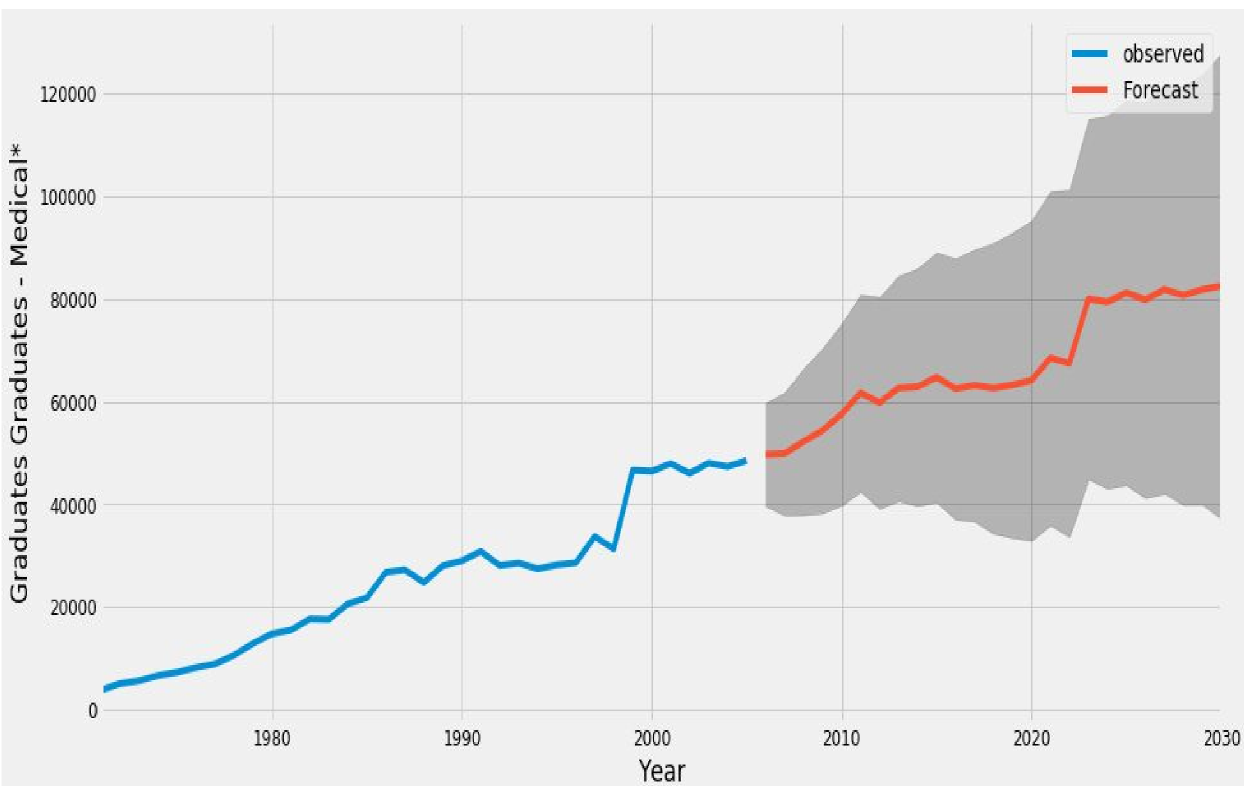
```
]: df.head(10)
```

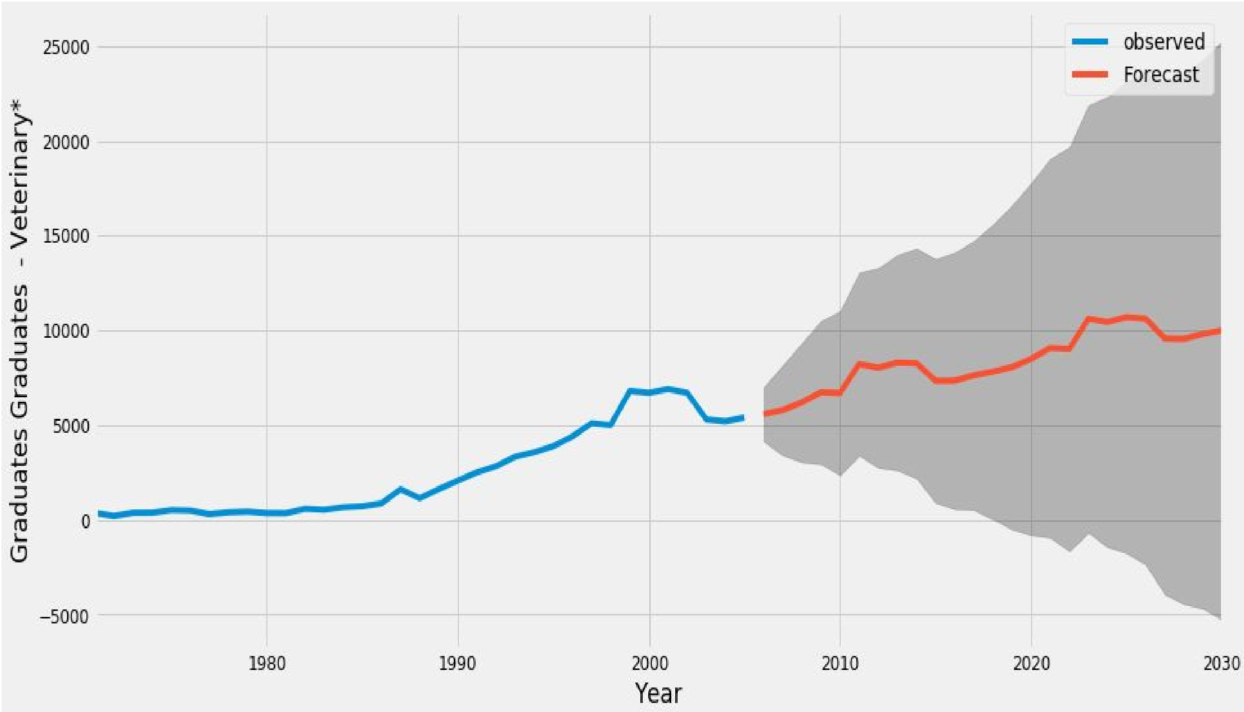
```python
def arima(field):
    for param in pdq:
        for param_seasonal in seasonal_pdq:
            try:
                mod = sm.tsa.statespace.SARIMAX(graduates[field],order=param,seasonal_order=param_seasonal,
                                                enforce_stationarity=False,
                                                enforce_invertibility=False)
                #print("ds")
                results = mod.fit()
                #print("ds")
                #print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic))
            except:
                continue

    y=graduates[field]

    mod = sm.tsa.statespace.SARIMAX(y,
                                    order=(1, 1, 0),
                                    seasonal_order=(1, 1, 0, 12),
                                    enforce_stationarity=False,
                                    enforce_invertibility=False)
    results = mod.fit()
    print(results.summary().tables[1])


    pred_uc = results.get_forecast(steps=25)
    pred_ci = pred_uc.conf_int()
    ax = y.plot(label='observed', figsize=(14, 7))
    pred_uc.predicted_mean.plot(ax=ax, label='Forecast')
    ax.fill_between(pred_ci.index,
                    pred_ci.iloc[:, 0],
                    pred_ci.iloc[:, 1], color='k', alpha=.25)
    ax.set_xlabel('Year')
    ax.set_ylabel('Graduates '+field)
    plt.legend()
    plt.show()
    y_forecasted = pred_uc.predicted_mean
    dfa=pd.DataFrame({'Year':y_forecasted.index, 'Predicited_'+field :y_forecasted.values})
    dfa['Year']= dfa['Year'].dt.year
    dfa.set_index('Year',inplace= True)
    return dfa
```

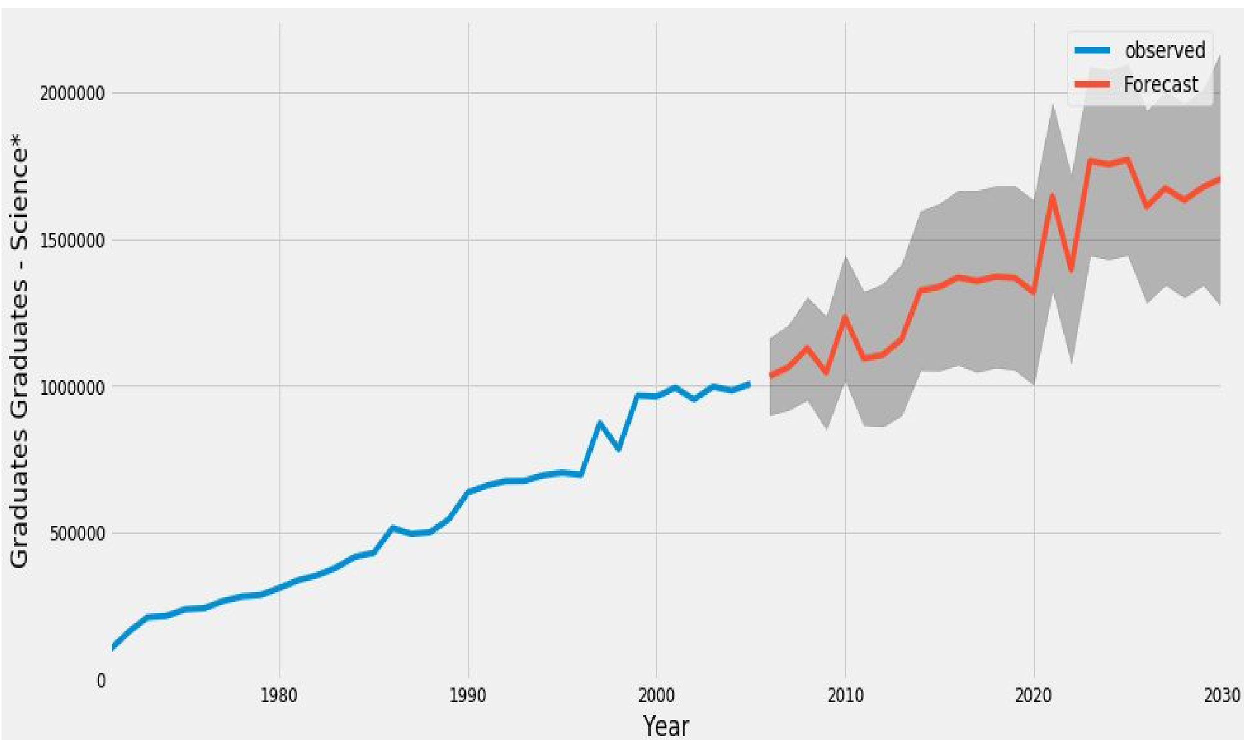
Producing and visualizing forecasts

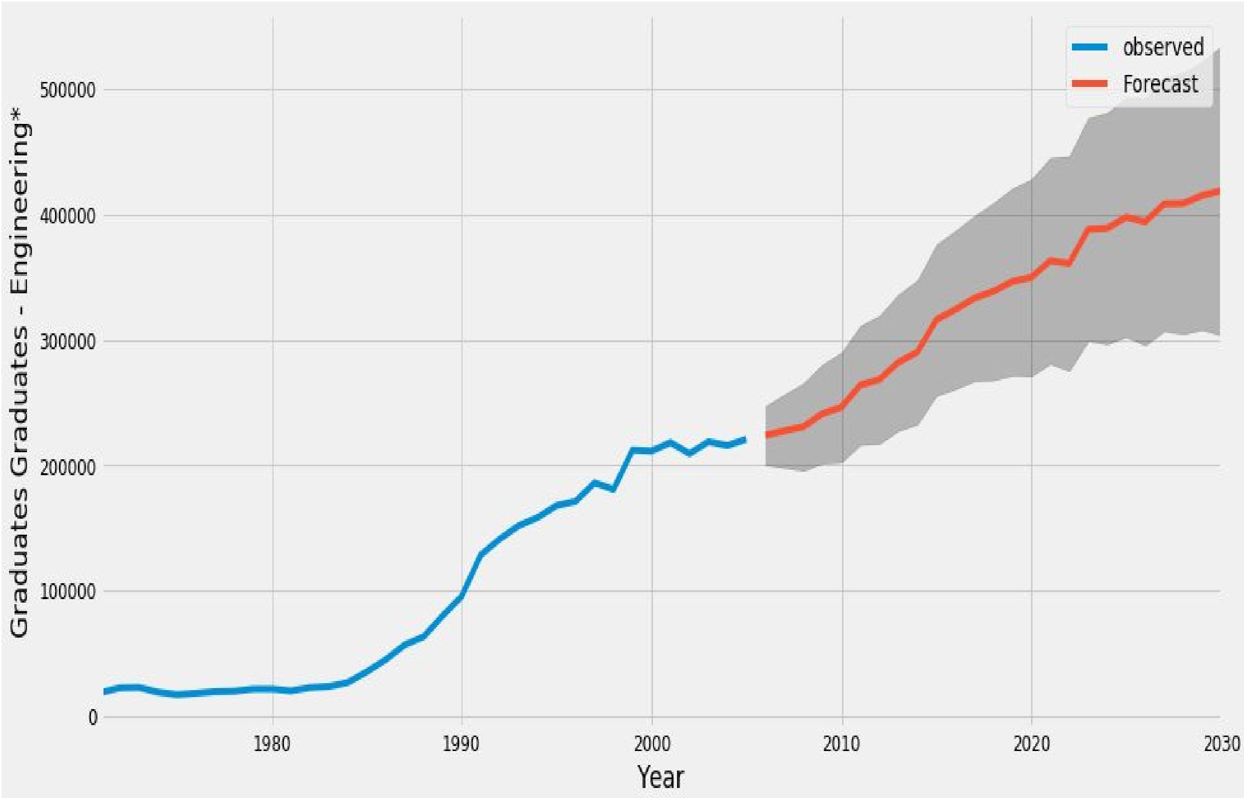
Our model clearly captured unemployed graduates trend. **As we forecast further out into the future, it is natural for us to become less confident in our values. This is reflected by the confidence intervals generated by our model, which grow larger as we move further out into the future.**

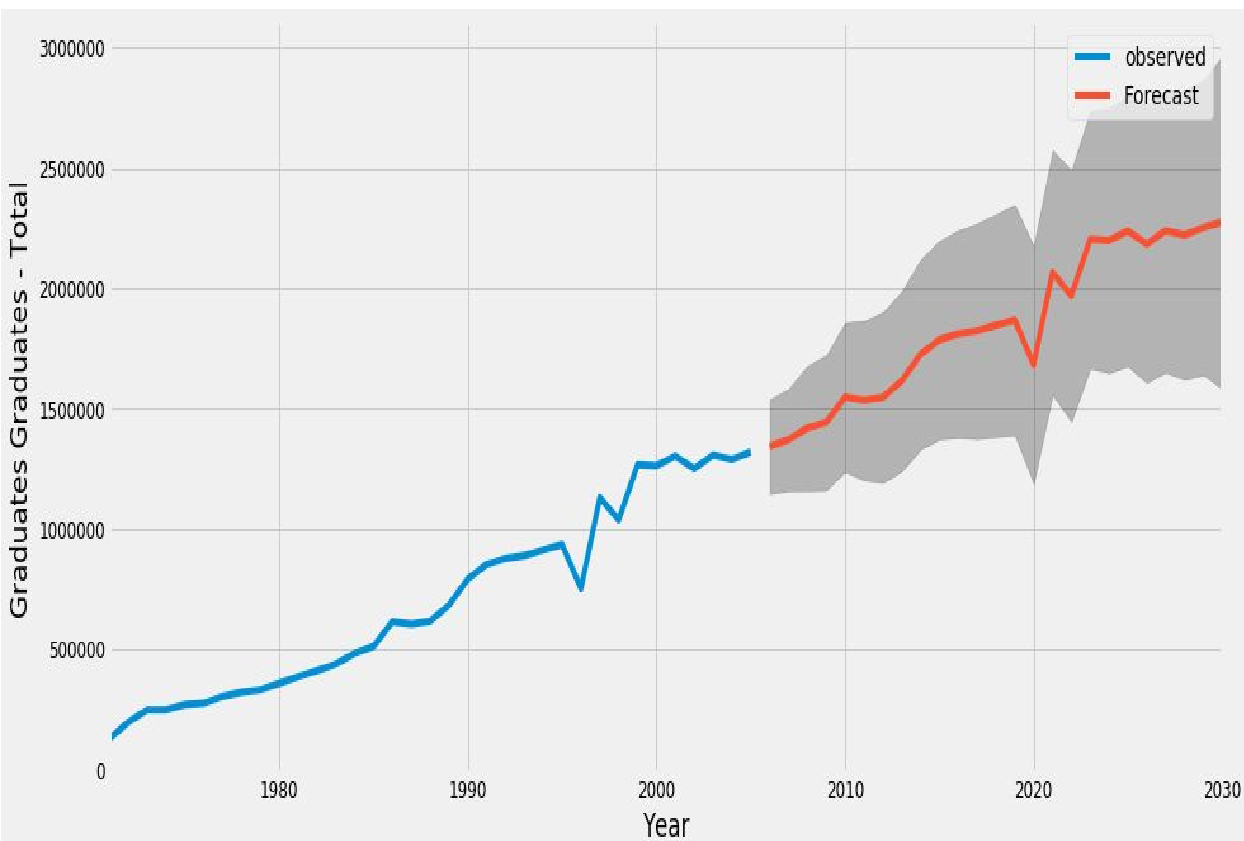

observed
Forecast
70000
60000
50000
40000
30000
20000
10000
Graduates Graduates - Agriculture*
1980
1990
2000
2010
2020
2030
Year


observed
Forecast
120000
100000
80000
60000
40000
20000
0
Graduates Graduates - Medical*
1980
1990
2000
2010
2020
2030
Year


observed
Forecast
25000
20000
15000
10000
5000
0
−5000
Graduates Graduates - Veterinary*
1980
1990
2000
2010
2020
2030
Year


observed
Forecast
2000000
1500000
1000000
500000
0
Graduates Graduates - Science*
1980
1990
2000
2010
2020
2030
Year


observed
Forecast
Graduates Graduates - Engineering*
0
100000
200000
300000
400000
500000
1980
1990
2000
2010
2020
2030
Year


observed
Forecast
Graduates Graduates - Total
0
500000
1000000
1500000
2000000
2500000
3000000
1980
1990
2000
2010
2020
2030
Year

	Predicited_Graduates - Medical*	Predicited_Graduates - Agriculture*	Predicited_Graduates - Veterinary*	Predicited_Graduates - Science*	Predicited_Graduates - Engineering*	Predicited_Graduates - Total
Year						
2006	49658.509790	37248.071192	5578.799810	1.032473e+06	223912.028593	1.342755e+06
2007	49844.241995	39832.445678	5786.277856	1.063288e+06	227569.776136	1.371654e+06
2008	52192.268786	37022.321957	6195.685148	1.128274e+06	230772.470534	1.419570e+06
2009	54346.844240	42261.500284	6727.152660	1.045418e+06	241134.665349	1.443083e+06
2010	57478.262847	45683.347479	6691.074670	1.233089e+06	246354.569415	1.548263e+06
2011	61641.994184	44273.573553	8220.623616	1.093655e+06	263993.134042	1.534901e+06
2012	59751.813353	44483.508163	8024.132352	1.105166e+06	268330.788043	1.546835e+06
2013	62580.091097	45708.608564	8296.366471	1.158280e+06	282027.060620	1.614651e+06
2014	62808.851781	44556.451454	8262.983864	1.325531e+06	290128.358932	1.725372e+06
2015	64688.056834	46950.294681	7335.083699	1.336975e+06	315935.891645	1.786265e+06
2016	62452.041886	46253.277922	7343.851588	1.369726e+06	323988.713555	1.810965e+06
2017	63111.776467	47012.851084	7621.506486	1.357051e+06	333068.373568	1.822894e+06
2018	62565.059816	46484.673753	7815.528732	1.372491e+06	338515.738011	1.846470e+06
2019	63182.456606	49100.831013	8052.197356	1.368730e+06	346299.499187	1.868810e+06
2020	64063.487599	46197.454927	8488.242076	1.319507e+06	349594.030419	1.682891e+06
2021	68428.224711	51431.501082	9063.045388	1.646752e+06	362990.142394	2.066250e+06
2022	67416.530029	54876.685436	9010.530735	1.396278e+06	360922.230529	1.969890e+06
2023	79929.092482	53481.291366	10609.622437	1.766801e+06	388115.780088	2.203565e+06
2024	79307.174599	53650.626034	10437.942569	1.754707e+06	388860.092304	2.198701e+06
2025	81138.751033	54859.338886	10691.602682	1.772022e+06	397733.022476	2.238358e+06
2026	79770.151095	53674.715550	10615.376712	1.611499e+06	394013.333734	2.183657e+06
2027	81739.997193	56069.365284	9566.082589	1.673848e+06	408345.225451	2.239579e+06
2028	80656.563562	55398.156773	9546.882317	1.633223e+06	408667.560490	2.220998e+06
2029	81721.697731	56176.125174	9804.569329	1.677656e+06	414963.794719	2.252100e+06
2030	82454.533822	55657.936033	9994.677336	1.708346e+06	418707.853710	2.274943e+06

Conclusion

So the above mentioned forecast are done for requirement of graduates for the period till 2030 in **INDIA.** The analysis is based **on MHRD** data available on government official webpage. The model used is **Autoregressive Integrated Moving Average,** which is one of the best model to fit the dataset. Root Mean Square Error (RMSE) tells us that our model was able to forecast yearly unemployed graduates in the test set within 34.14 of the real sales. Our unemployed graduates yearly ranges from around 40000 to over 80000. In my opinion, this is a pretty good model so far. The time-series has an upward trend between any year and the growth rate for the unemployed graduates are increasing rapidly. So we need to create a much more jobs to meet the requirement for the upcoming demands.