Comparative Analysis of Energy Consumption in Kazakhstan by different regions and Prediction with Machine Learning time-series technique (ARIMA)

Astana IT University

Sultan Khajidursunogly, Batyrkhan Shutenov

Instructor:

Zhakiyev Nurkhat

2022, Nur-Sultan

Agenda

1. Abstract
2. Introduction
    a. The relevance of research
    b. Air pollution in Pavlodar region
    c. Energy generation and energy balance levels in Kazakhstan
3. Formulation of the problem
    a. Perspectives of detailed analysis by regions
    b. The reason of choosing Pavlodar's energy consumption data
4. Main Body
    a. Machine Learning Prediction Methods
    b. Performance evaluation
    c. Comparative Analysis between regression and moving-average
    d. Implementation of ARIMA model
    e. Predictions results and test-summary
5. The discussion of the results and Conclusion
6. References

Abstract

The rapid development of cities requires more and more energy. Load forecasting is an essential part of energy systems. This article presents an intelligent analysis of energy consumption in the cities of Kazakhstan, as well as a predictive model of energy consumption based on methods of computational and intellectual data analysis for the city of Pavlodar. The key concepts of the research is predict energy generation values by given data and make conceptual analysis, also conclusions. Data on energy consumption was taken from the Internet and other resources. Firstly, had been checked regression techniques to see how our data is distributed by linear and polynomial functions, after that we decided to use ARIMA statistical analysis model was used for time-series forecasting.

Keywords: ARIMA, forecast, energy consumption, statistical models

Introduction

Every year the level of energy consumption is growing all over the world. Accordingly, the importance of optimizing energy consumption comes first. A key factor underlying the contrasting trends in energy demand in developed and emerging economies are the significant differences in the level of energy consumption per capita. These differences in energy consumption largely reflect differences in economic development and prosperity, as well as a range of other factors, including economic structure, local climatic conditions and differences in natural resource endowments. These differences in the current levels of energy consumption between developed and emerging economies are also reflected in average carbon emissions per capita, offset only partially by the lower average carbon intensity of the fuel mix in the developed world relative to emerging economies. After analyzing the energy consumption between cities, it is necessary to identify in which regions the most energy is consumed. After analyzing the data, it is required to conduct a comparative analysis between the regions, thereby proving the importance of the energy balance between the regions and raising the environmental problem of air pollution. The accuracy of load forecasting balances and also optimizes the efficiency of generating equipment. Therefore, the better the model, the more. balanced energy consumption.

In this study, the Pavlodar region will be taken as the region with the highest energy consumption, for which energy consumption will be predicted.

Weather data and hourly energy consumption helped to build a good model with decent accuracy. The figure below shows that the energy demand is growing every year in all regions, but in the Pavlodar region, it is growing very rapidly. The reasons are industrial facilities that require a lot of energy.

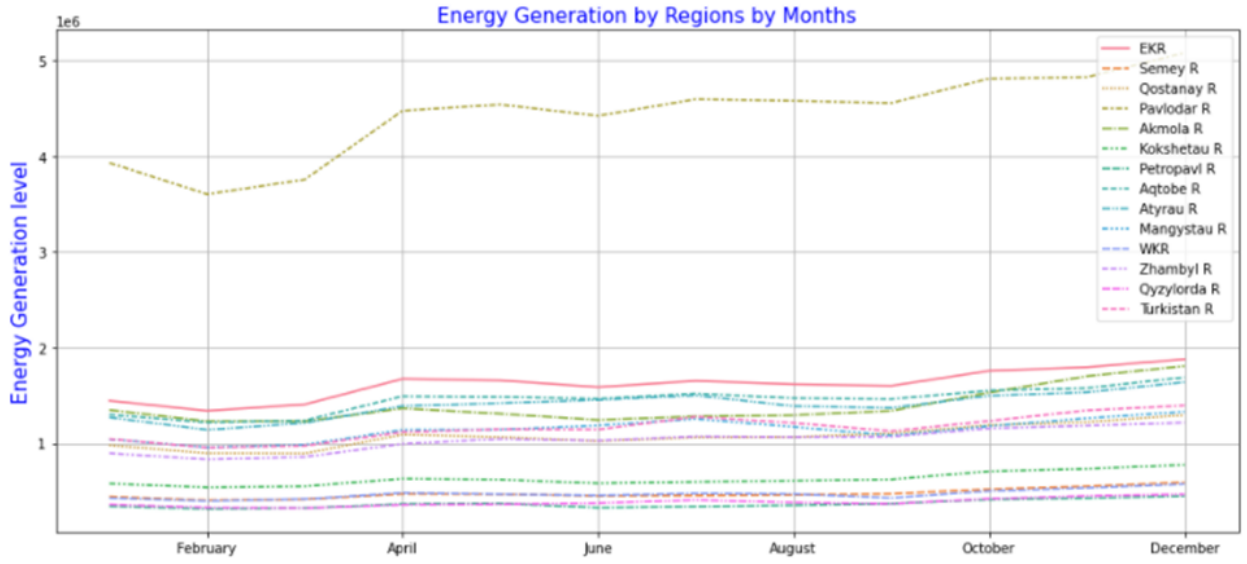

Fig. 0 - Comparative Analysis of Energy-Balance level between regions by year-period

Formulation of the problem

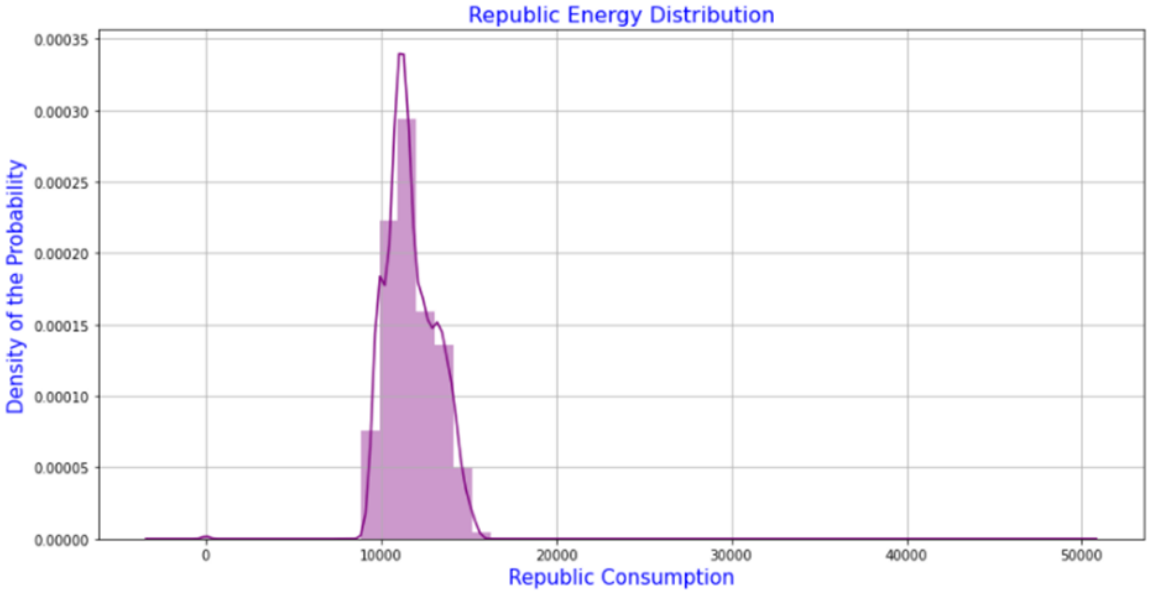

Fig. 1 - Probability Distribution (Consumption in MWt)

We can observe a near-normal distribution(bell curve) over consumption values, moreover, can be concluded that we have some errors and outliers in the data which are negative or large values which means that we also have some incorrectly working sensors with some defects.

The purpose of our project is to make a detailed analysis of the CHP data, to draw different conclusions, as well as to identify anomalous data. We analyzed the data of the CHP of the city of Pavlodar and made a comparative analysis with other regions, thereby proving the importance of the energy balance between the regions and raising the environmental problem of air pollution in the Pavlodar region. “The champion in terms of emissions is the Pavlodar region with the highest emissions into the environment - 717 thousand tons of emissions per year. Over the past three years, emissions have increased by 14%, which is 100,000 tons,” Mirzagaliyev said. Thereby optimizing the energy balance to reduce fuel consumption using different machine learning models and making predictions.

Main Body

Machine Learning Prediction Methods

There are too many methods for predicting energy consumption. The K nearest neighbour (KNN) method is one of the accepted ones since it relies on similar samples to make good predictions.

First of all, we used Linear Regression to identify is our data can be linearly predicted, results can be seen by sklearn.metrics library such as R-Squared, MAE, MSE, RMSE.

```
In [199]: from sklearn.metrics import r2_score

In [66]: from sklearn.model_selection import train_test_split
         from sklearn.linear_model import LinearRegression
         X_train, X_test, y_train, y_test = train_test_split(dff.loc[:,'Month':'p'],
                                                             dff['Consumption'],
                                                             random_state=42)

In [67]: reg = LinearRegression().fit(X_train, y_train)
         reg.score(X_test, y_test)

Out[67]: 0.5306641896309613
```

Fig. 2 - Python Script of Implementation Regression models by sklearn library

So, by the linear regression, R-Squared coef. gives us 53%, which means 3 years period data does not totally follow the linear trend and we cannot make the right predictions with it.

There are also regression models, such as Linear regression, but in most cases, it shows poor results, since the data may be nonlinear. And it is also necessary to mention the polynomial regression, which still takes into account some descents and ups, adding a polynomial to the algorithm. There are also a lot of projects using ARIMA, GARCH, LSTM statistical forecasting models, which in most cases shows a good result since it makes forecasts based on recent data.

Performance evaluation

We used some metrics to evaluate the accuracy of the forecasting model (all of them can be found in sklearn.metrics library in a python programming language):

R-Squared:

$$R^2 = \frac{\text{Variance explained by the model}}{\text{Total variance}}$$

Mean Absolute Error (MAE): MAE measures the average magnitude of the errors in a set of predictions, without considering their direction.

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Root mean squared error (RMSE): RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

These components (Dependent columns) are defined as follows:

- **Year**: Year of the generated dataset
- **Month**: The increasing or decreasing value in the series.
- **Week**: Week of the dataset
- **Day**
- **Sky:**
- **Temperature:** for that day
- **Wind Velocity**
- **Pressure:** Air pressure in millimetre of mercury
- **Consumption:** Consumption in MWatt in the hour

```
[67]: from sklearn.preprocessing import PolynomialFeatures

      poly = PolynomialFeatures(degree = 2)
      X_poly = poly.fit_transform(X_train)
      X_test_poly = poly.fit_transform(X_test)

      poly.fit(X_poly, y_train)
      lin2 = LinearRegression()
      lin2.fit(X_poly,y_train)

[67]: LinearRegression()

[68]: lin2.score(X_test_poly,y_test)

[68]: 0.5979190723656687
```

Fig. 3 - Python Script of Implementation Polynomial Regression with degree 4 by sklearn library

Method have demonstrated better coefficient of determination (R-Squared) 59.8% of polynomial regression with degree 4 by giving columns as features, which was lower compared to traditional forecasting methods such as BPNN, ARIMA.

A time series consists of a set of input parameters (one of which is time) and one output parameter that depends on the inputs. Our task is to find this dependence. A direct and naive approach in this situation would be a linear regression of the form a1x1 + a2x2 + ... + anxn.

The main problem with this approach is the autocorrelation of the time series - the dependence of the indicators of the time series on previous values. This ultimately leads to the impossibility of assessing the significance of the regression coefficients and makes the forecasting confidence interval unreliable.

The problem of autocorrelation can be resolved using the ARIMA model (also known as the Box-Jenkins method) - an integrated autoregressive - moving average model.

After the analysis, we concluded that using linear regression is not an appropriate way to make predictions for the energy generation in short term year periods, because our data is mostly cycled data in short-term, therefore data must be divided into several parts such as long-term and short-term predictions, so, in that situation, we can classify models by time-series predictions. Linear Regression can be used if the dataset consists of long-term year periods such as 1990-2020, however, we have to use other ML models to work with high volatility in short term periods.

In the picture below, can be represented that approximately energy consumption by one day period for every hour. And data is not linear and have some volatility, decreasing and increasing points, to optimize, we must use time series forecasting models, such as ARIMA (*Using ARIMA model, we can forecast a time series using the series past values. In this post, we build an optimal ARIMA model from scratch and extend it to Seasonal ARIMA (SARIMA) and SARIMAX models.* ARIMA, short for 'Auto-Regressive Integrated Moving Average' is actually a class of models that 'explains' a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values.)

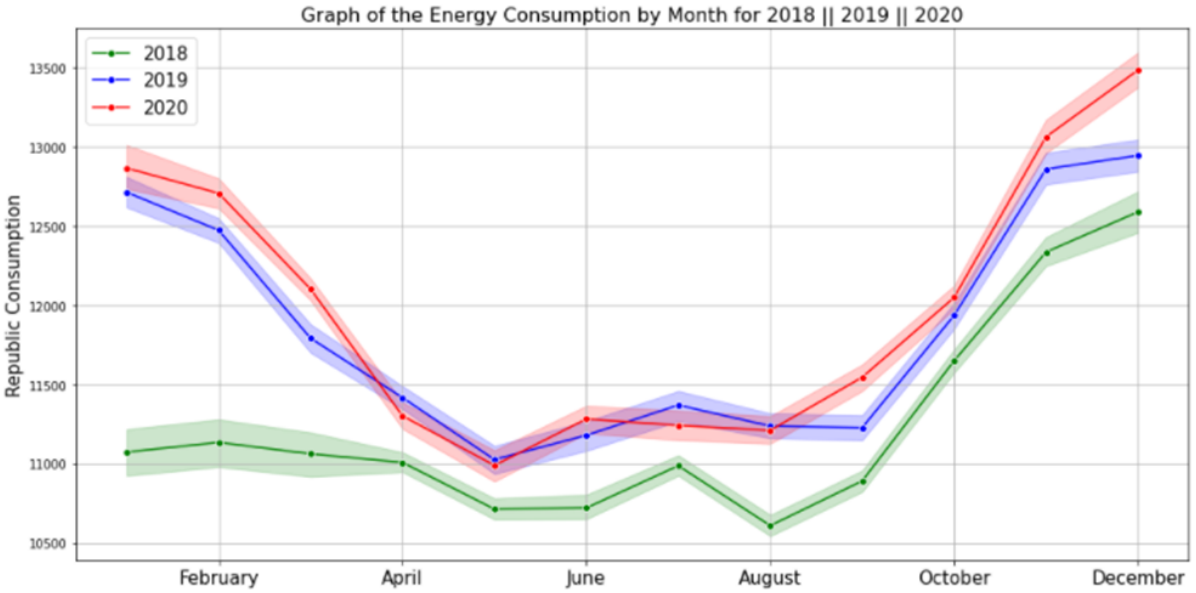

Fig. 4 - Line Graph of Energy consumption of 3 year-periods in MWt by months

The ARIMA model uses three integer parameters: p, d, and q.

- p is the order of autoregression (AR). It can be interpreted as the expression "an element of the row will be close to X if the previous p elements were close to X".
- d is the order of integration (differences of the initial time series). It can be understood as "an element will be close in value to the previous d elements if their difference is minimal."
- q is the order of the moving average (MA), which allows the model error to be set as a linear combination of previously observed error values.

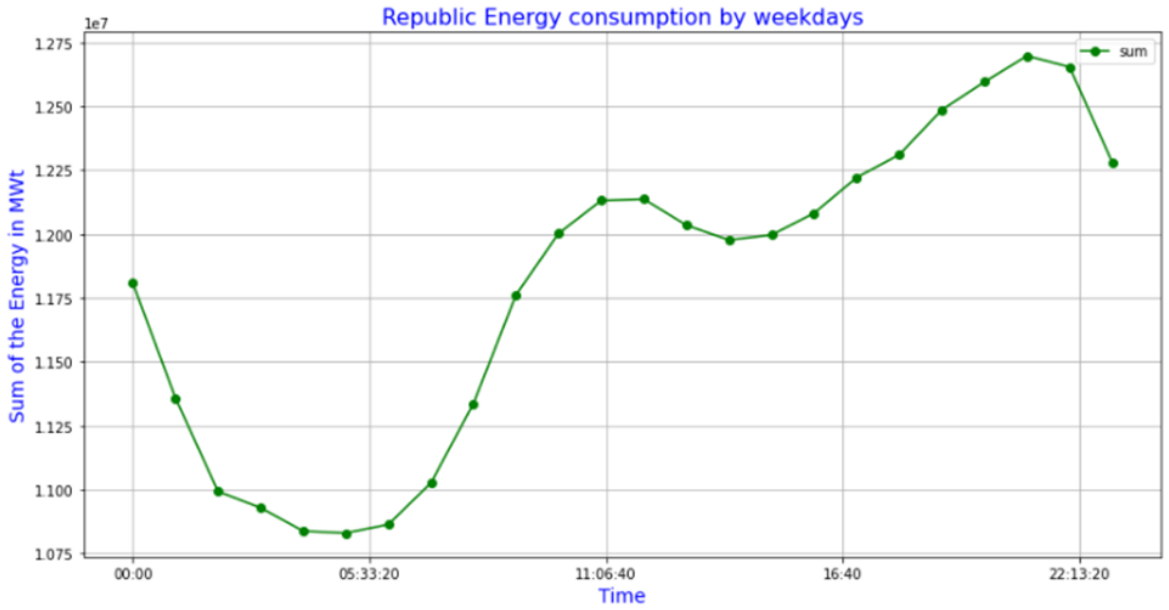

Fig. 5 - Republic Energy Consumption [MWt] by weekday

ARIMA, because LSTM finds out hidden periodicities in the big energy consumption dataset. Overall, the ARIMA method has shown excellent potential in short-term forecasting with high-volatility. .

Implementation of ARIMA model

A rolling average is calculated by taking input for the past 12 months and giving a mean consumption value at every point further ahead in the series. After finding the mean, we take the difference between the series and the mean at every point in the series.

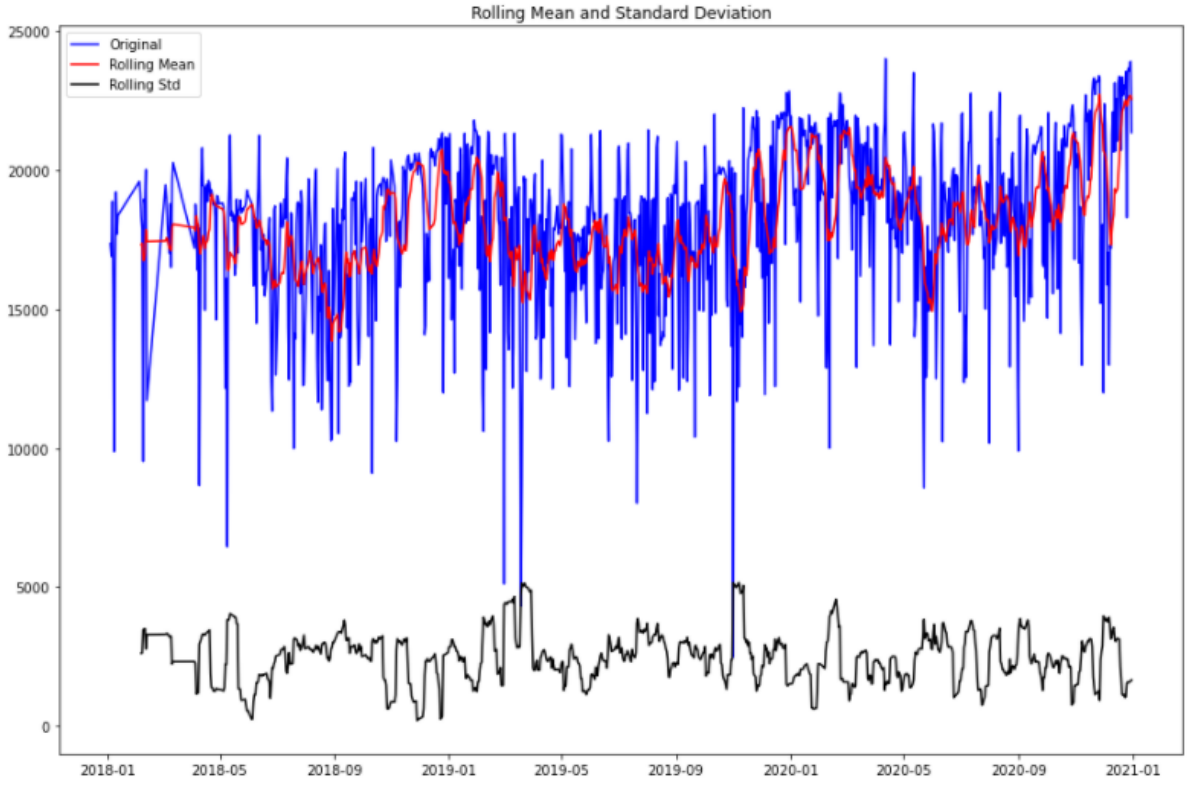

Fig. 6 - Pavlodar's data with Moving Average and Standard Deviation

Through the above graph, we can see the increasing mean and standard deviation and hence our series is stationary.

```
Results of dickey fuller test
Test Statistics                -6.923207e+00
p-value                         1.132505e-09
No. of lags used                7.000000e+00
Number of observations used     9.980000e+02
critical value (1%)            -3.436919e+00
critical value (5%)            -2.864440e+00
critical value (10%)           -2.568314e+00
dtype: float64
```

The test statistics is less than the critical values, so the data is stationary and cyclic, but we can make it more stationary by removing the trend and seasonality from the series.

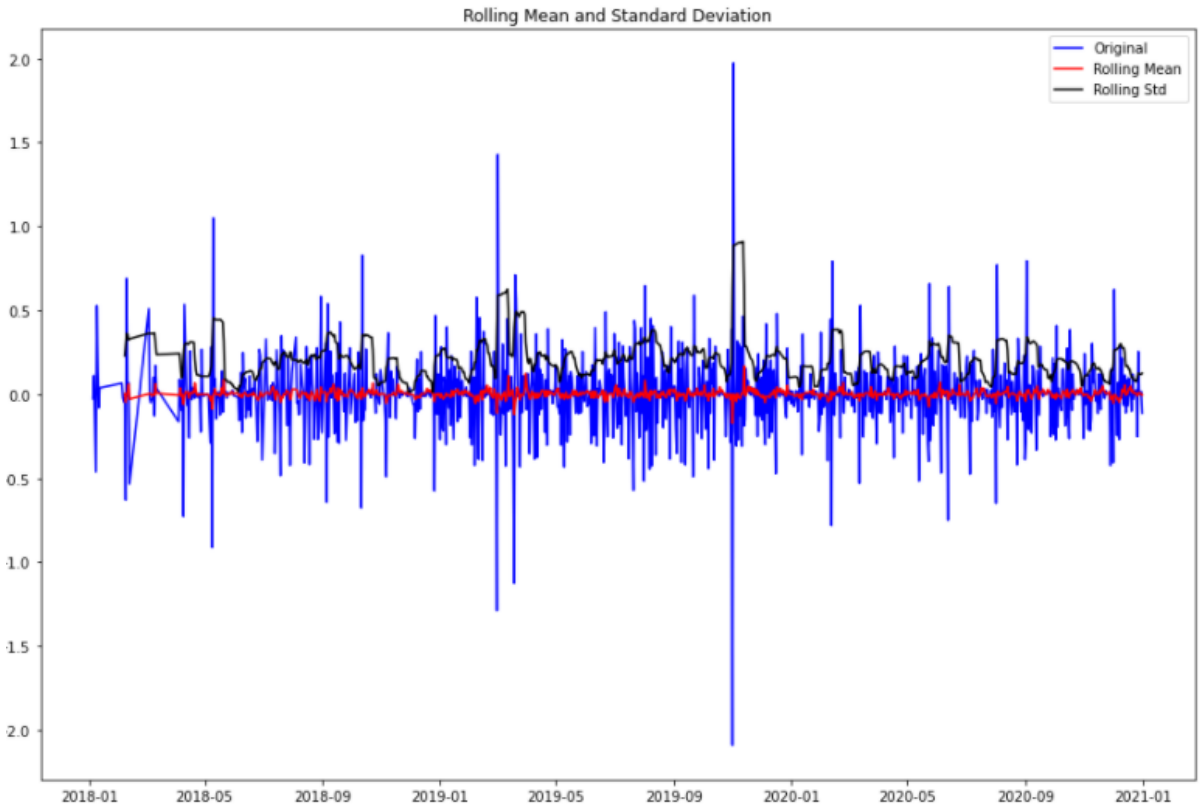

Fig. 7 - Scaled Energy Consumption data of Pavlodar which remained more stationary

From the above graph, we observed that the data attained stationarity.

```
Results of dickey fuller test
Test Statistics                -1.143737e+01
p-value                         6.299856e-21
No. of lags used                2.100000e+01
Number of observations used     9.830000e+02
critical value (1%)            -3.437020e+00
critical value (5%)            -2.864485e+00
critical value (10%)           -2.568338e+00
dtype: float64
```

You can check the optimality of the model using the built-in procedures of the library.

We are most interested in the table of coefficients in the picture below. The coef column shows the effect of each parameter on the time series, and P>|z| - significance. The closer the value P>|z| to zero, the higher the significance.]

result.summary() returns us a table like this

```
[96]: result_AR.summary()
```

[96]:

ARIMA Model Results

Dep. Variable:	D.Consumption	**No. Observations:**	1005
Model:	ARIMA(3, 1, 3)	**Log Likelihood**	268.592
Method:	css-mle	**S.D. of innovations**	0.185
Date:	Wed, 09 Mar 2022	**AIC**	-521.184
Time:	11:40:48	**BIC**	-481.882
Sample:	1	**HQIC**	-506.250

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0002	0.000	0.742	0.458	-0.000	0.001
ar.L1.D.Consumption	-1.4478	0.242	-5.977	0.000	-1.923	-0.973
ar.L2.D.Consumption	-0.4428	0.174	-2.539	0.011	-0.785	-0.101
ar.L3.D.Consumption	0.1390	0.069	2.025	0.043	0.004	0.274
ma.L1.D.Consumption	0.6830	0.239	2.860	0.004	0.215	1.151
ma.L2.D.Consumption	-0.8066	0.057	-14.215	0.000	-0.918	-0.695
ma.L3.D.Consumption	-0.7462	0.205	-3.633	0.000	-1.149	-0.344

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	-1.0938	-0.3774j	1.1571	-0.4471
AR.2	-1.0938	+0.3774j	1.1571	0.4471
AR.3	5.3720	-0.0000j	5.3720	-0.0000
MA.1	1.0396	-0.0000j	1.0396	-0.0000
MA.2	-1.0602	-0.4062j	1.1354	-0.4418
MA.3	-1.0602	+0.4062j	1.1354	0.4418

Fig. 8 - The main Statistics of our ARIMA model

Less the RSS value, the more effective the model is. We will check with ARIMA(3,1,1), etc to look for the smallest values of RSS.

Fig. 9 - Representation of fitted line and actual line with RSS measure

The residual sum of squares (RSS) is **a statistical technique used to measure the amount of variance in a data set that is not explained by a regression model itself**. Here can be represented that RSS value is approximately 34.5, which is a good result for us.

Predictions

Now that we have verified that our model is optimal, we will see how well ARIMA can predict the subsequent values of the series.

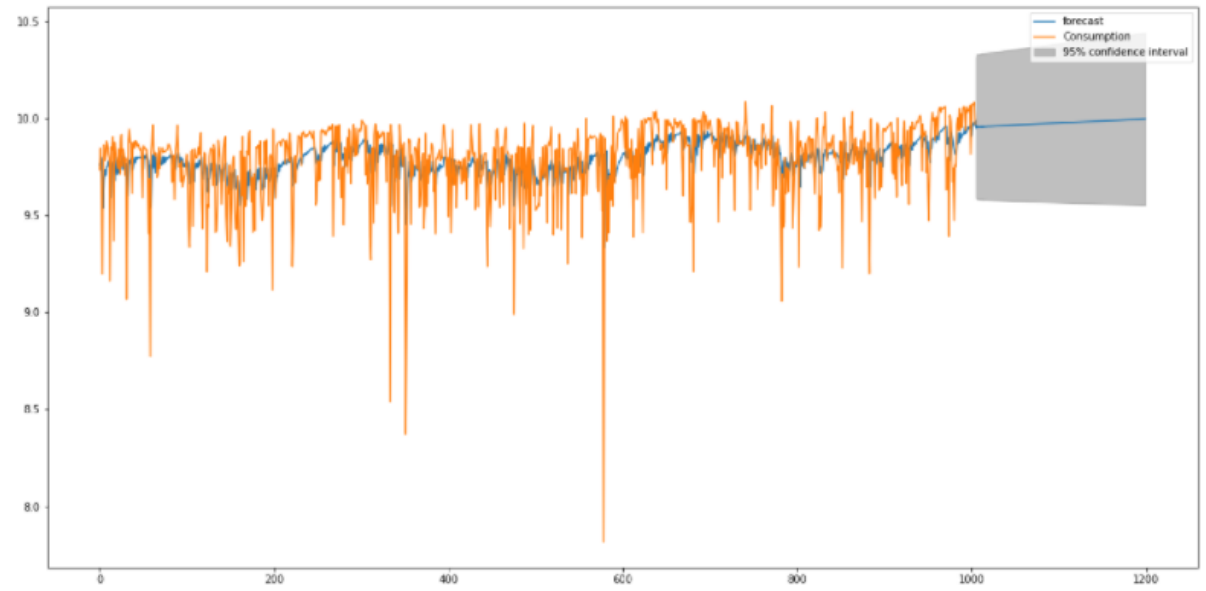

Fig. 10 - The graph of energy consumption with prediction in 3 years period with confidence interval 95%

From the above graph, we calculated the future predictions till next year. The greyed out area is the confidence interval which means the predictions will not cross that area.

We can evaluate the accuracy of the forecast not only with the help of a graph, but also in other ways. For clarity, we will take the average absolute percentage error (MAPE - Mean Absolute Percent Error):

```
In [155]: x=result_AR.forecast(steps=1006)
          forecasted = result.predict()
          actual = df_log['Consumption']
          mape = np.mean(np.abs((actual - x[0])/actual))*100
          mape

Out[155]: 2.814740595444499
```

Fig. 11 - Python script of predicting values and evaluating the performance of the model by MAE

And by the result, we can see that our model shows an error of 2.815 %. The forecast turned out to be quite accurate.

The discussion of the results and Conclusion

Finally, we were able to build an ARIMA model and actually forecast for a future time period. Keep note that this is a basic implementation to get one started with time series forecasting. There are a lot of concepts like smoothening etc and models like ARIMAX, prophet, etc to build your time series models.

References

- *ARIMA model for relationl data*. (n.d.). Www.Kdnuggets.Com. Retrieved March 9, 2022, from https://www.kdnuggets.com/2019/10/automl-temporal-relational-data.html
- Chauhan, N. (2020, January 2). *Predict Electricity Consumption Using Time Series Analysis*. Https://Www.Kdnuggets.Com/. Retrieved March 9, 2022, from https://www.kdnuggets.com/2020/01/predict-electricity-consumption-time-series-analysis.html
- *Growth in global energy demand comes entirely from emerging economies*. (n.d.). Https://Www.Bp.Com/En. Retrieved March 10, 2022, from https://www.bp.com/en/global/corporate/energy-economics/energy-outlook/demand-by-region.html

- Piatetsky, G. (2020, April 9). *24 best (and free) books to understand machine learning*. Https://Www.Kdnuggets.Com. Retrieved March 9, 2022, from https://www.kdnuggets.com/2020/04/top-stories-2020-mar.html
- Popov, I. (2021, June 24). *REGRESSION AND ARIMA PREDICTION IN STATSMODELS*. Https://Newtechaudit.Ru. Retrieved March 9, 2022, from https://newtechaudit.ru/arima-v-statsmodels/
- Prabhakaran, S. (2021, August 22). *ARIMA model – complete guide to time series forecasting in python*. Https://Www.Machinelearningplus.Com/. Retrieved March 9, 2022, from https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/
- scikit-learn developers. (n.d.). *Metrics and scoring: Quantifying the quality of predictions*. Https://Scikit-Learn.Org/. Retrieved March 9, 2022, from https://scikit-learn.org/stable/modules/model_evaluation.html
- Valchanov, I. (2018, November 5). *Sum of squares total, sum of squares regression and sum of squares error*. Https://365datascience.Com/. Retrieved March 9, 2022, from https://365datascience.com/tutorials/statistics-tutorials/sum-squares/
- Огурцов, И. л. ь. я. (2020, January 27). *«Чемпионом» по загрязнению воздуха в Казахстане является Павлодарская область*. Https://Liter.Kz/. Retrieved March 10, 2022, from https://liter.kz/chempionom-po-zagryazneniyu-vozduha-v-kazahstane-yavlyaetsya-pavlodarskaya-oblast/