**Electricity Production forecasting using Time Series Analytics**



**Dheeraj Kashyap Varanasi – wt3631**

**Siddhharth Reddy – fo4203**

**Mohan Srinivas Vendra – us2205**

**Rajashekar Reddy Ayluri – nv7709**

**Sai Rohith Reddy Nagella – tx8218**

**Summary**

In this project, we utilized the Electricity Production dataset obtained from Kaggle, which contains yearly data from 1985 to 2017. The goal was to predict electricity production using time series techniques. To begin, we visually analyzed the data using a historical plot and observed a consistent upward trend from 1985 to 2017. Additionally, we discovered that the data is strongly correlated, as evidenced by the significant autocorrelation coefficients for all 12 lags. We performed basic predictability tests using both the normal approach and the first lag differencing approach and found that the data was predictable.

We then applied several time series techniques to forecast the data, including Two-Level Forecasting (Quadratic trend + seasonality and MA trailing method), Holt-Winter's model with automated selection of error/level, trend, and seasonality, and ARIMA (Autoregressive Integrated Moving Average). We calculated the accuracy measures for each method and determined that ARIMA was the most effective approach for forecasting the data. We compared the accuracy measures for ARIMA with those of the seasonal naïve and naïve forecasts, ultimately selecting ARIMA as the most suitable method for forecasting.

To compare the accuracy of the different methods, we utilized four measures: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Error Percentage (MAPE). We primarily evaluated the models based on their MAPE and RMSE comparisons.

**Introduction**

Modern society is highly reliant on electricity production, which powers everything from homes and businesses to transportation and communication infrastructure. In the United States, electricity is generated using a variety of energy sources, including fossil fuels, nuclear energy, and renewable energy. According to the U.S. Energy Information Administration (EIA), the country produced 4.01 trillion kilowatt-hours (kWh) of energy in 2020, with most of it coming from fossil fuels.

The dataset used for this project contains electricity production data from 1985 to 2017, which was obtained from the Kaggle website. Generally, electricity production has increased over time, with production defined as the kilowatt-hours (kWh) generated in a day, month, or year.

series analysis is a valuable tool for forecasting various types of outcome variables, such as sales or stock volume, using historical time-based data. This is also known as prescriptive analysis. The focus of this project is to forecast electricity production using various time series analysis techniques applied to the historical data collected from Kaggle. This could provide a useful solution for countries to manage their electricity production based on available funds and plan accordingly.

**8- steps of forecasting**

**Step -1: Define Goal:**

The main objective of this research is to predict the power output for the upcoming fiscal year, specifically for the year 2018, using historical data from the dataset. The goal is to develop a prediction model that accurately forecasts data for the next 12 months, while taking into account seasonality and trend patterns. The preferred model is the one with the highest level of accuracy, as it will aid in future data forecasting. To implement the strategies, R was chosen as the programming language for this research.

**Step -2: Get data:**

The dataset used in our project was obtained from Kaggle.com, an online community for sharing and discovering datasets. It covers a period from January 1985 to December 2017 and consists of 396 data points. Our goal is to use this data to forecast the electricity production for the next 12 months, from January 2018 to December 2018.

**Step -3: Explore and visualize series:**

The historical plot of the data indicates an upward trend, with no noticeable decrease in production over the years. However, there is observable seasonality in the data, with repeating patterns, and the peak production value occurred in 2014.

A picture containing text, font, screenshot, line

Description automatically generated

**Trend, seasonality and level plot :**

From the below plot we can understand that there is an upwards trend in the data hence, there is a trend component. Which can be true from the below stl() plot aswell.

**A picture containing text, font, screenshot, line

Description automatically generated**

From the seasonal section we can infer that there is an additive seasonality and there is also a level component in the data. The noise is present in all the data but it is quite high in the end of the data.

**Autocorrelation plot:**

The plot below displays the correlation coefficients for up to 12 lags. It is observed that the autocorrelation coefficients are significantly high for all the lags. Lag 1 and lag 12 have the highest autocorrelation coefficients, both with a value of approximately 0.9, indicating the presence of trend and seasonality components in the data.

**A picture containing text, diagram, line, plot

Description automatically generated**

**Step -4: Data Preprocessing:**

In the data we have two columns, one is date column and other is value one which contains production of electricity. We have used ts() function to convert the whole data into time series data.

A picture containing text, font, screenshot, line

Description automatically generatedA screenshot of a data

Description automatically generated with low confidence

396 observations are contained in the time series data electric.data.ts file.

**Checking the predictability of data:**

We need to check whether the data is predictable or a random walk for this we followed two approaches first is the general approach with the null hypothesis and Ar(1).

* **Approach – ARIMA – AR(1):**

ar1 <- 0.9993

s.e. <- 0.0010

null\_mean <- 1

alpha <- 0.05

z.stat <- (ar1-null\_mean)/s.e.

z.stat p.value <- pnorm(z.stat)

p.value if (p.value < alpha){

“Reject null hypothesis”

} else {

“Accept null hypothesis”

}

A screenshot of a computer program

Description automatically generated with medium confidence

Based on this approach we got p-vale < 0.05 and this results in

**“Reject null hypothesis.”**

This means that the data is predictable. In addition, we can go to another approach which is first lag differencing approach.

* **Approach – 2 ACF with differencing lag 1:**

Below graph represents the auto correlation of the first lag differencing where all the autocorrelation coefficients are significant and are above the horizontal dashed lines, hence the graph shows that the data is predictable.

**A picture containing text, diagram, screenshot, line

Description automatically generated**

**Step -5: Partition series:**

We get 396 data points in total, and when we divide the time series data into two parts, we receive 308.6 as 80% of the training data and 79.2 as 20% of the validation data out of the 396 data points we have. As it is preferred to divide the yearly data into the proper ratio of years , we got the training data of 324 records(26 years) for the training period and 72 records(6 years) for the validation period.

**Training data:** train.ts



**Validation data:** valid.ts



**Step 6 & 7 : Apply forecasting and comparing performance:**

1. **Two level forecasts (quadratic trend and seasonality with moving average for residuals):**

The trailing MA may be used to data with trend or seasonality using two-level forecasting, which combines two forecasting models.

**Level1**: A quadratic trend in regression with seasonality. Additionally, it may be used to remove seasonality and/or patterns from historical data.

Finding residuals (errors): The differences between the forecast using the regression trend with seasonality and actual data points for different time periods.

**Level2**: The residuals (errors) of the regression can be predicted using trailing MA.

We combine these Regression with quadratic trend with seasonality and trailing MA to bring the overall forecasts of the data.

To improve the linear trend and seasonality regression model and anticipate model residuals, a trailing moving average was used. After combining these components, a two-level model and a model over all the data were created and used to anticipate the upcoming 12 months.

**Model trained over training data:**

A screenshot of a computer

Description automatically generated with medium confidence

Looking at Adjusted R – Squared value 0.9641 (96%), we can say that the model is a good fit. Considering overall p – value is less than 0.05 hence the model is significant, and all the coefficients are also significant. May be applied for the forecasting.

**Model Equation: (Regression model with quadratic trend and seasonality)**

**For our case:**

***where, t = 1,2,3……n (n=number of time periods/trends)***

***D2 = binary (1,0), it is 1 if Feb and 0 if otherwise***

***D3 = binary (1,0), it is 1 if Mar and 0 if otherwise***

***.***

***.***

***D12 = binary (1,0), it is 1 if Dec and 0 if otherwise***

***If D2 , D3 ,…, D12 are 0 then it is Jan***

**Selecting K (window width):**

We want to apply the trailing Moving Average method to several window sizes for the training data and select the best window size with statistical backing (by contrasting accuracy metrics for all selected window sizes). This approach will help us predict better.

**To select the window width from all the selected window sizes using the moving average for the residuals, we compare the accuracy measures for various sizes as well:**

**A picture containing text, screenshot, font, number

Description automatically generated**

From the common accuracy measure the moving average with window width 3 and 8 have the similar MAPE and RMSE values for both window widths we can take any one of them, we selected window width = 3.

Window width Graph for all three window widths

A picture containing text, screenshot, diagram, line

Description automatically generated

From the above graph we are choosing either window width as 3 or 8 but, in the project, we are using 3 as the window width in the trailing MA.

**Two-level predictions = regression model with linear trend and seasonality + Trailing MA for residual**

**Two level forecasting for validation data**

A screenshot of a computer

Description automatically generated with low confidence

**Predictions Graph:**

A picture containing text, line, font, plot

Description automatically generated

**Forecasted values for validation data:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Just to check the comparison we used only quadratic trend without seasonality as well for the level – 1:**

**A screenshot of a computer program

Description automatically generated with low confidence**

**The Adjusted – R square value is 0.7439 which is around 74% the output is reliable on the input variables. Comparatively the quadratic trend with seasonality is having high adjusted R – Square value which is a good fit.**

To measure them correctly we are going to use the common accuracy measurements below for the Two – level forecasting.

A screenshot of a computer code

Description automatically generated with low confidence

The Two – Level forecasting with quadratic trend and seasonality with MA trailing for residual has LOW MAPE and RMSE values. Hence, it is a good fit among other two models i.e., seasonal naïve and Two – level with quadratic trend and MA trailing.

**For Entire Dataset (Two – level Forecasting):**

Firstly, we applied quadratic trend with seasonality for entire data and the observations are below from the R – code:

A screenshot of a computer program

Description automatically generated with low confidence

Looking at the image above the adjusted – R square is around 0.9607 (96%). Hence, we can conclude that it is a good fit for the data as the p – value is less than 0.05, the model is significant, and the coefficients are significant as well. Hence, this model may be applied for the time series forecasting.

**Model Equation:**

For this entire dataset based on coefficients model equation is:

***where, t = 1,2,3……n (n=number of time periods/trends)***

***D2 = binary (1,0), it is 1 if Feb and 0 if otherwise***

***D3 = binary (1,0), it is 1 if Mar and 0 if otherwise***

***.***

***.***

***D12 = binary (1,0), it is 1 if Dec and 0 if otherwise***

***If D2, D3 ,…, D12 are 0 then it is Jan***

**Forecast of 12 months after applying two – level forecast:**

After getting the quadratic trend and seasonality predicted mean and residuals, we take the residuals and run it through the MA trailing then we combine both the levels to predict the future values of year 2018 as shown in the image below.

**A screenshot of a computer

Description automatically generated with low confidence**

**Accuracy Measures:**

**A picture containing text, screenshot, font, line

Description automatically generated**

From the above picture it is evident that the MAPE and RMSE values are quite low for Two – level Forecasting with quadratic trend + seasonality and MA trailing method. Hence, we can say that two – level Forecasting with Quadratic trend+ seasonality and MA trailing is better as compared to Seasonal naive and naive methods.

**Holt-winter’s:**

The next technique utilized for the time series analysis is the Holt-Winters model of advanced exponential smoothing. The Holt-Winter's (HW) or simply Winter's model is used for time series that exhibit trend and seasonality. By including a seasonal component, the goal is to enhance Holt's model. This model is appropriate for making predictions since it creates estimates that take into consideration both the trend and seasonality components.

**Advanced Exponential Smoothing:**

The Holt-Winters model of advanced exponential smoothing is the next method used for the time series analysis. Since this model makes projections considering both the trend and seasonality components, it is suitable. The model was initially tested using the training and validation partitions before being run on the complete dataset.

**Automated Holt-Winter Model (Z, Z, Z):**

Ets() function uses model=ZZZ and chooses the best possible parameters for alpha, beta, gamma.

**where:**

**α = smoothing constant for exponential smoothing**

**β = smoothing constant for trend estimate**

**𝜸𝜸 = smoothing constant for seasonality estimate**

**k = periods to be forecasted into future**

**M = number of seasons**

A screenshot of a computer code

Description automatically generated with low confidence

**Plot for training and validation data:**

A picture containing text, screenshot, line, plot

Description automatically generated

**Holts winter Model built over entire data.**

A screenshot of a computer

Description automatically generated with medium confidence

**Forecast for 12 months in 2018:**

A screenshot of a computer

Description automatically generated with low confidence

**Plot of holt’s winter model over entire data:**

A picture containing text, line, font, plot

Description automatically generated

From the above plot we can understand that the holt’s winter model is perfectly fitting into the data for entire dataset and is also able to forecast the results of the future year from January 2018 to December 2018.

**Accuracy measure of this model compared with naïve, seasonal naïve and holts winter:**

A screenshot of a computer code

Description automatically generated with low confidence

By comparing the MAPE and RMSE values the Holt’s winter model is quite better than seasonal naïve and naïve model. Hence, we are using the Holt’s winter for the time series forecasting as the measures are better for the data.

**ARIMA Model:**

ARIMA (Autoregressive Integrated Moving Average) model is a widely used statistical model for time series forecasting. It is a combination of autoregression (AR), differencing (I), and moving average (MA) models.

The AR component represents the linear dependence of the current value on past values, where the number of past values to consider is controlled by the order of the model (p). The I component represents the degree of differencing needed to make the series stationary, where the order of differencing is controlled by (d). The MA component represents the linear dependence of the current value on past errors, where the order of the model is controlled by (q).

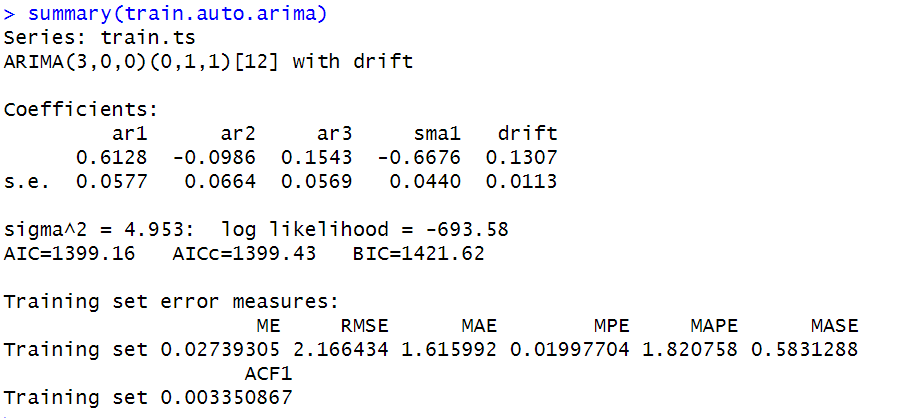
ARIMA models are capable of capturing trends, seasonality, and other complex patterns in the data. They are widely used in various industries for forecasting applications, such as predicting sales, stock prices, weather patterns, and more. The accuracy of ARIMA models depends on the appropriate selection of model parameters (p, d, q), which can be done using statistical techniques or by trial and error.

**Auto.arima():**

`auto.arima` is a function in the R programming language that automatically selects the best ARIMA model for a given time series data. The function uses a stepwise approach to evaluate a large number of possible ARIMA models and selects the one with the lowest Akaike Information Criterion (AIC) value.

The `auto.arima` function is used because selecting the best ARIMA model for a time series can be a difficult and time-consuming process. The function simplifies this process by automating the selection process and choosing the optimal ARIMA model. This can save time and effort for analysts and help ensure that the best model is selected for forecasting.

**Arima model for training data:**



**In this model:**

ARIMA (3,0,0) (0,1,1) [12] with drift

p = 3, order 3 autoregressive model AR(3)

d = 0, order 0 differencing to remove linear trend

q = 0, order 0 moving average MA(0) for error lags

P = 0, order 0 autoregressive model no AR() for seasonality

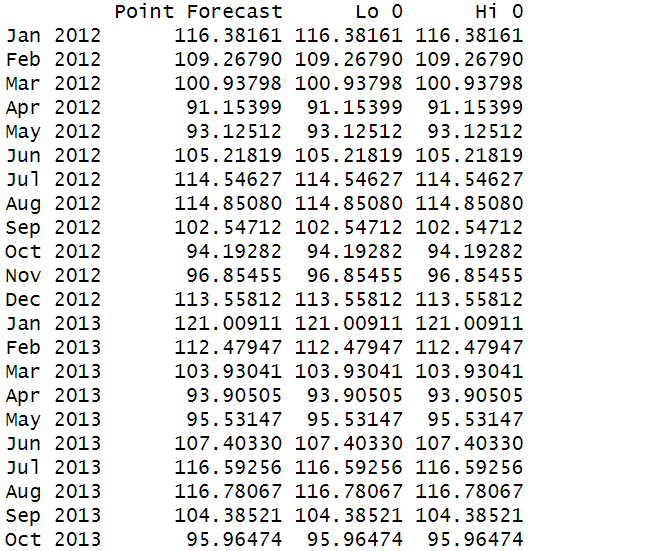
D = 1, order 1 differencing to remove linear trend

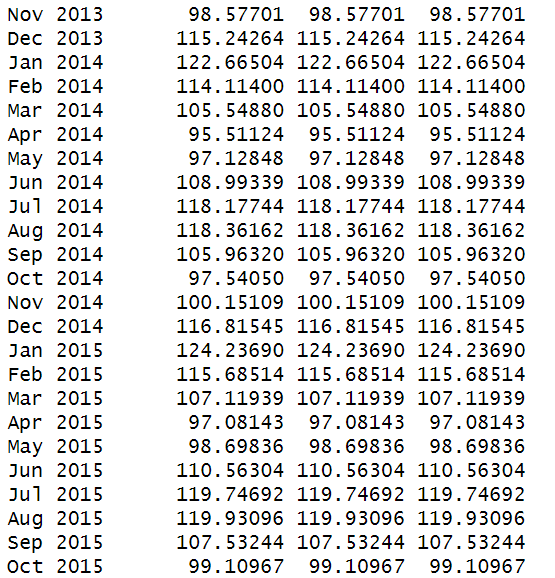
Q = 1, order 1 moving average MA(1) for error lags

m = 12, for monthly seasonality

**Model Equation:**

**Forecast for validation:**



A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, font, screenshot, line

Description automatically generated

**Plot for the auto.arima():**

A picture containing text, screenshot, line, plot

Description automatically generated

**Auto Arima model for entire data:**

A screenshot of a computer code

Description automatically generated with low confidence

**In this model:**

ARIMA (2,1,1) (0,1,2) [12]

p = 2, order 2 autoregressive model AR(2)

d = 1, order 1 differencing to remove linear trend

q = 1, order 1 moving average MA(1) for error lags

P = 0, order 0 autoregressive model no AR() for seasonality

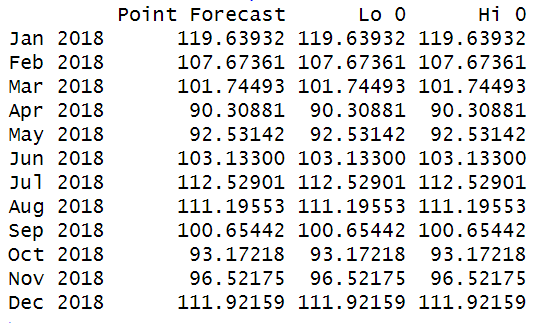
D = 1, order 1 differencing to remove linear trend

Q = 2, order 2 moving average MA(2) for error lags

m = 12, for monthly seasonality

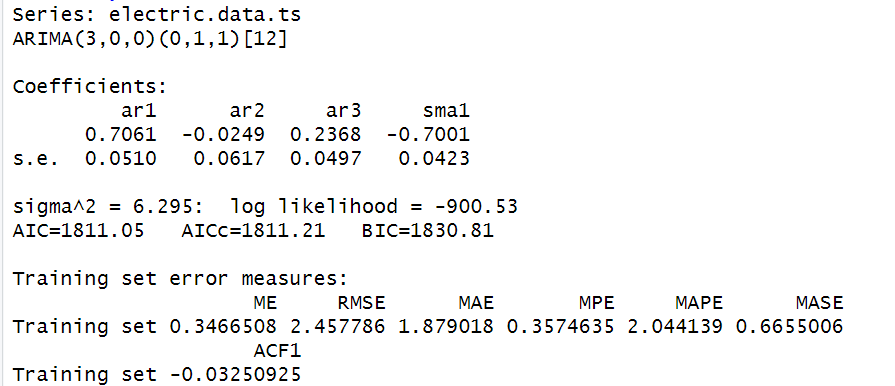
**Model Equation:**

**Forecast:**



**Arima (3,0,0) (0,1,1) [12] model for entire data:**

Summary:



**In this model:**

ARIMA (3,0,0) (0,1,1) [12] with drift

p = 3, order 3 autoregressive model AR(3)

d = 0, order 0 differencing to remove linear trend

q = 0, order 0 moving average MA(0) for error lags

P = 0, order 0 autoregressive model no AR() for seasonality

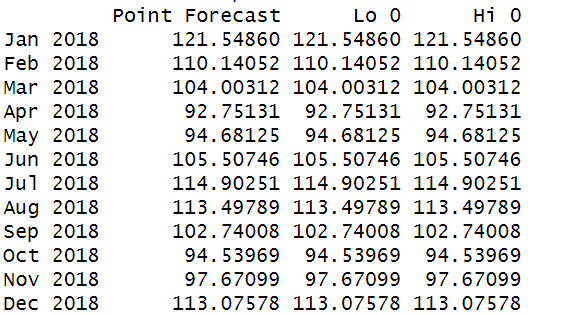
D = 1, order 1 differencing to remove linear trend

Q = 1, order 1 moving average MA(1) for error lags

m = 12, for monthly seasonality

**Model Equation:**

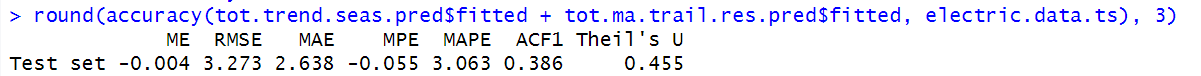
**Forecast**:

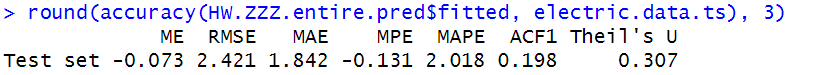


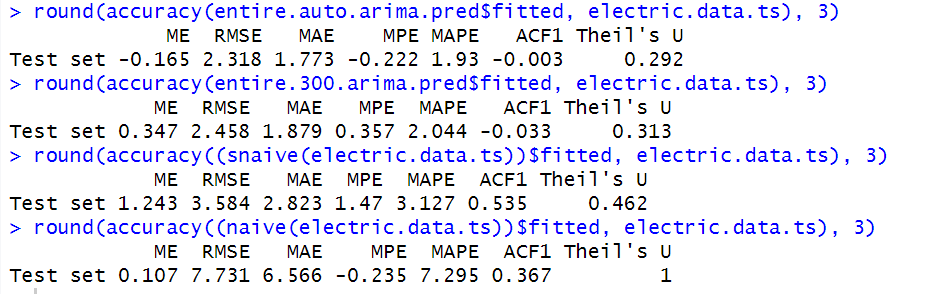
**Plot for future prediction:**A picture containing text, screenshot, line, plot

Description automatically generated

**Step – 8 Implement Forecast:**

****

****



**Below is the comparison table of all methods:**

|  |  |  |
| --- | --- | --- |
|  | MAPE | RMSE |
| Two-level (MA.Trailing Residuals) | 3.063 | 3.273 |
| HoltWinter’s.zzz | 7.295 | 7.731 |
| auto.arima | 1.93 | 2.318 |
| ARIMA(3,0,0)(0,1,1)[12] | 2.044 | 2.458 |
| Seasonal Naïve | 3.127 | 3.584 |

Auto.Arima() is having lowest MAPE value so, we are selecting auto.arima() is our best model to forecast the result.

**Forecast auto.arima() for next 12 months:**

A screenshot of a computer

Description automatically generated with low confidence

**Forecast of ARIMA (3,0,0)(0,1,1)[12] for 12 months:** A screenshot of a computer

Description automatically generated with low confidence

**CONCLUSION**

In this study, we used a number of data analytics approaches (Time series) to acquire some understanding of power production. To analyze its association with the data, we used a variety of models from Regression (two-level), Holt’s winter, and Auto Arima. This concluding analysis showed that the ARIMA models with the lowest MAPE and RMSE are the Auto.Arima() model, which comes in first, followed by the ARIMA (3,0,0) (0,1,1) [12] model, and finally, the forecast for the next year of 2023 utilizing the Auto Arima model.

**Limitations on study:**

* Limitation in study regarding the electric production and economics of them to understand about the trend.
* This analysis is only based on the historical data and do not include factors that impact the electricity production such as weather conditions or low in manpower, anything else.

**Reference for the data:**

<https://www.kaggle.com/datasets/kandij/electric-production>.