ANALYSIS AND FORECASTING ELECTRIC VEHICLE DEVELOPMENT REVENUE TRENDS USING ARIMA MODEL

TRƯƠNG DƯƠNG MINH NHẬT[1*], NGUYỄN THANH THUẬN[1], TRƯƠNG QUỐC BẢO[1], NGUYỄN HẢI TÂM[1], NGUYỄN ĐỨC TÀI[1]

[1]*Faculty of Information Technology, Industrial University of Ho Chi Minh City*
[1*] *tdminhnhat13092003@gmail.com*

Abstract. Nowadays, we are living in a new era in the use of almost all tools, electric vehicles, more specifically, the means of transportation around us. Electric vehicles have only been prominent in the past few years, but recently we have seen electric vehicles developing strongly and it is almost the future trend of the world in recent years. We cannot deny that electric vehicles are the main intention to replace gasoline and crude oil vehicles, at the same time they have great benefits in protecting the environment, but how can we know that electric vehicles will be popular and develop strongly and sustainably or not. Therefore, using the ARIMA model to forecast the development trend of electric vehicles to see if it will continue to develop strongly in the following years or just stagnate in stages or not develop strongly but gradually weaken.
Keywords. ARIMA, Time – Series, Regression, Supervised learning

PHÂN TÍCH VÀ DỰ BÁO XU HƯỚNG PHÁT TRIỂN PHƯƠNG TIỆN ĐIỆN BẰNG MÔ HÌNH ARIMA

Tóm tắt. Ngày nay, chúng ta đang sống ở trong một kỷ nguyên mới trong việc sử dụng hầu hết tất cả các công cụ, phương tiện chạy bằng năng lượng điện, cụ thể hơn đó chính là phương tiện di chuyển xung quanh chúng ta. Gần như phương tiện điện chỉ mới nổi trội ở các thời điểm của vài năm trước nhưng dạo gần đây chúng ta thấy gần như phương tiện điện nó phát triển một cách mạnh mẽ và gần như nó là xu hướng tương lai của thế giới của mấy năm gần đây. Chúng ta không thể phủ nhận rằng, phương tiện điện chính là ý đồ chính để có thể thay thế được các phương tiện chạy bằng xăng; dầu thô sơ, đồng thời chúng có lợi ích to lớn trong việc bảo vệ môi trường nhưng làm sao chúng ta thể biết được rằng phương tiện điện sẽ phổ biến và phát triển mạnh có bền vững hay không. Do đó, nhóm chúng tôi quyết định sử dụng mô hình ARIMA để dự báo xu hướng phát triển phương tiện điện để xem nó có phát triển mạnh nữa cho các năm sau không hay là chỉ ngập ngừng ở các giai đoạn hay là không phát triển mạnh mà yếu dần đi.
Từ khóa. ARIMA, Chuỗi thời gian, Hồi quy, Máy học quan sát

1 INTRODUCTION

Forecasting future development trends using the ARIMA model is one of the popular techniques that mainly relies on the factor of continuous variation between values corresponding to time at past points in time and from there predicting new values corresponding to future points in time. This is considered a suitable model when applying revenue at past points in time to be able to forecast future revenue and from there we can confirm whether the development of electric vehicles is really strong or not? Of course, when training a machine learning model, cannot predict completely accurately, but we can only predict at a relative level.
The steps takes to handle the problem are as follows:
Step 1: Collect data for analysis.
Step 2: Process relevant data after collection.
Step 3: Visualize the data so can analyze and orient what to do before applying the machine learning model.
Step 4: Split the training data set and the test data set.

Step 5: Train the machine learning model using the training data set.
Step 6: Let the model predict on the test set to evaluate the quality.
Step 7: Forecast the future using the model.

2 THEORETICAL BASIC

2.1 Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF)

2.2.1 Autocorrelation Function (ACF)

In ACF, the correlation coefficient is on the x-axis whereas the number of lags (referred to as the lag order) is shown in the y-axis. An autocorrelation plot can be created in Python using plot_acf from the statsmodels library and can be created in R using the ACF function. [1]
Normally, to calculate the correlation between lags, most of them calculate the correlation between the value at this time compared to its past values by calculating the auto-correlation coefficient. As known, the correlation coefficient only fluctuates in the range of -1 to 1, lags with positive correlation show that the lag data at that time has signs of correlation compared to past data and vice versa, legs with negative correlation show that the lag data at that time has no signs of correlation and the correlation is very bad.
To calculate the auto-correlation coefficient using the following definition:

$$r_k = \frac{\sum_{t=k+1}^{T}(y_t - \bar{y}) - (y_t - k - \bar{y})}{\sum_{t=1}^{T}(y_t - \bar{y})^2}$$

In there:

- r: is the lag in this time k
- T: is the length of the time-series
- y: the value of time-series

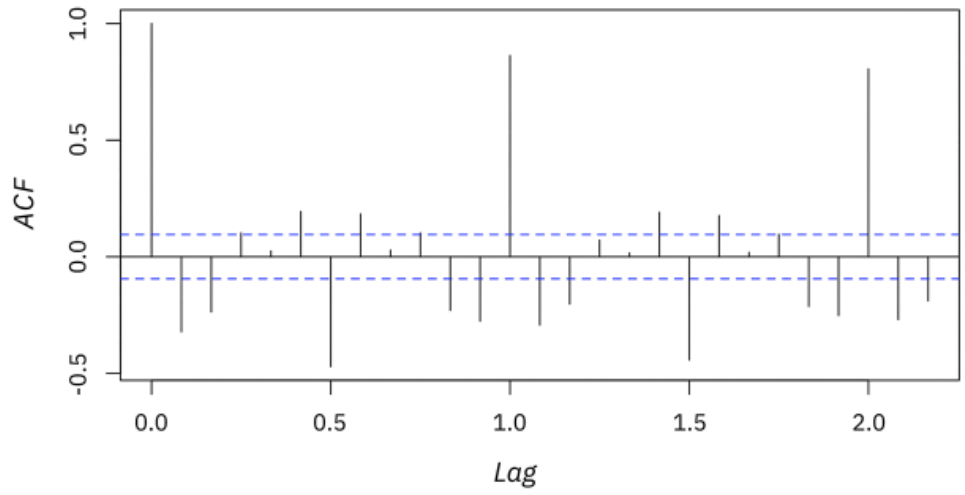

Figure 2.2.1.1: Example visualization of the ACF (Autocorrelation Function) plot

Most of the time the first lag position always has the highest correlation and then gradually decreases or stabilizes between the following lags. Based on graph 2.2.1.1, it can be seen that the 12th lag has a very strong correlation, similarly the 24th lag also gives a strong correlation but also slightly less than the 12th lag.

2.2.2 Partial Autocorrelation Function (PACF)

Another important plot in preparing to use an ARIMA model on time series data is the Partial Autocorrelation Function. An ACF plot shows the the relationship between yt and y_{t-k} for different values

of k. If y_t and y_{t-1} are correlated, then y_{t-1} and y_{t-2} will also be correlated. But it's also possible for yt and yt−2 to be correlated because they are both connected to y_{t-1}, rather than because of any new information contained in y_{t-2} that could be used in forecasting y_t. To overcome this problem, can use partial autocorrelations to remove a number of lag observations. These measure the relationship between y_t and y_{t-k} after removing the effects of lags 1 to k. So the first partial autocorrelation is identical to the first autocorrelation, because there is nothing between them to remove. Each partial autocorrelation can be estimated as the last coefficient in an autoregressive model. [1]

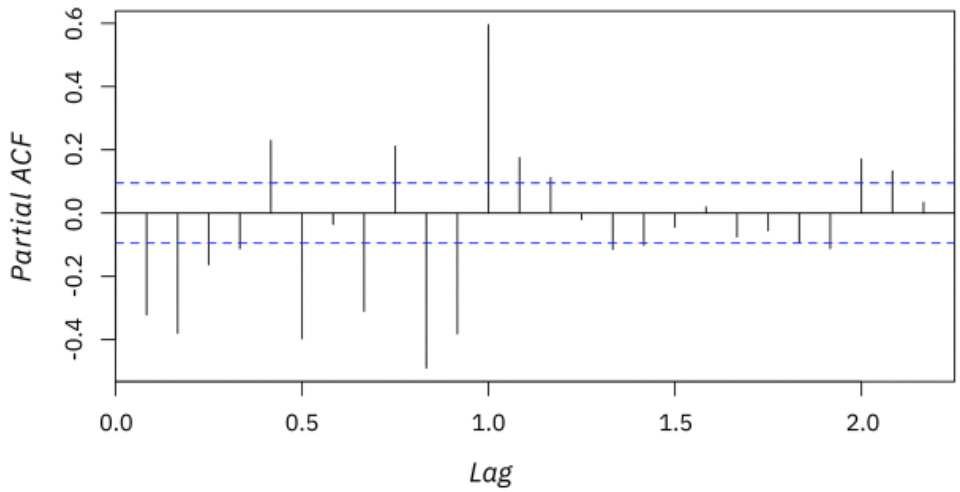

Figure 2.2.2.1: Example visualization of the PACF (Partial Autocorrelation Function) plot

Based on the graph, it can be seen that the value at lag 1 has the highest correlation coefficient instead of starting from lag 0, thus showing that there is a strong direct correlation at this lag position. The PACF values gradually decrease and fall within the confidence interval (blue dash) from lag 2 onwards.

Compare the characteristics between two correlation functions

Characteristic	ACF	PACF
Type Correlation	Both directly and indirectly	Only directly
Role	Find AR (Autoregressive – p)	Find MA (Moving average – q)
Chart form	Gradually decrease or fluctuate with lag	Quickly drops to zero after some lag
Effect to lag	Includes mediating influence	Eliminate intermediate influences

2.2 Stationary and Difference

A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, are not stationary — the trend and seasonality will affect the value of the time series at different times. On the other hand, a white noise series is stationary — it does not matter when observe it, it should look much the same at any point in time. [3]
Thus, a time series is considered stationary if it meets the requirements such as the statistical characteristics, expectation, variance, and autocorrelation between them do not change over time. Therefore, the time series will not have repeated seasonal changes, and the variance will not increase or decrease over time.

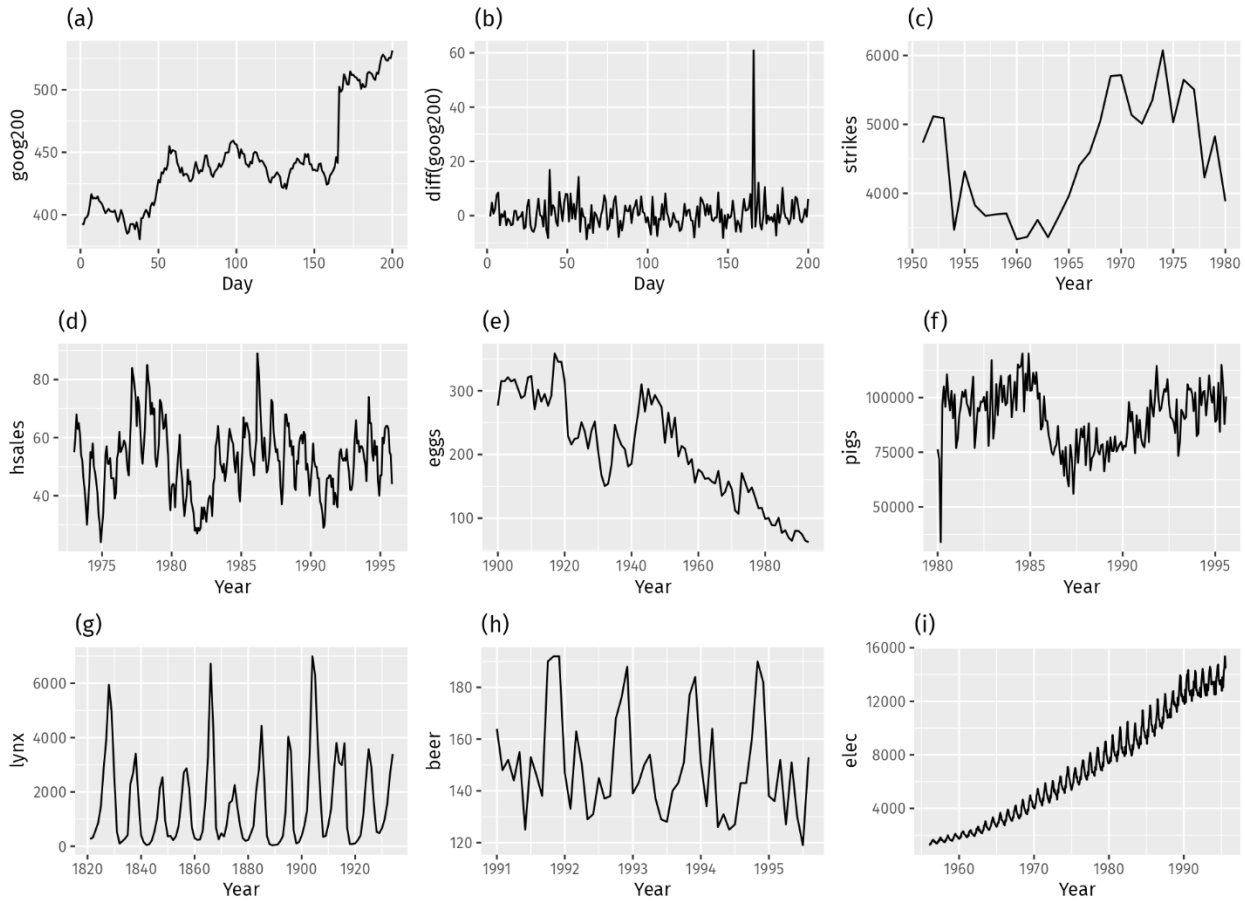

Figure 2.2.1: Illustration of non-stationary time series and stationarity time series

The 9 pictures in figure 2.2.1 are examples of a that time series, but to the naked eye, which time series are stationary and which are not? Thus, from the 9 time series graphs can determine that, when viewed with the naked eye, graphs (d), (h) and (i) are seasonal time series. season, according to the stationary series property, for a time series to be stationary, the first condition is that the series must not have seasonal properties, and the time series graphs (a), (c), (e), (f) and (i) are time series with a strong increasing or decreasing trend, the variances or correlations always change gradually according to the series trend, so those graphs are not stationary. The remaining graphs (b) and {g} are stationary graphs, because their time series always fluctuate around a mean value, and variance, and do not change over time. So in summary, only graphs (b) and (g) are stationary time series, the remaining graphs are non-stationary.

To be able to transform a non-stationary time series into a stationary series, there are many different techniques such as: trend removal, seasonality removal (if the time series is seasonal), non-linear transformation, moving average, and removing non-stationary components by filtering. In the article, only 3 main methods are mentioned: trend removal, seasonality removal, and non-linear transformation because these 3 methods are the 3 most popular and easiest ways to transform a non-stationary time series into a stationary time series.

The case of "trend removal" with 2 methods: "difference taking" and "trend decomposition":

- The "difference taking" method is by subtracting the current value from the previous value to remove the linear trend, if the first difference is not stationary, continue to take the second difference. Take the difference until the series is stationary. But when applying this method, there is a disadvantage which is the loss of information if used excessively.

$$Y_t' = Y_t - Y_{t-1}$$

- The "trend decomposition" method is to apply a regression model to estimate a linear or nonlinear trend and then subtract this trend from the series. (The goal is to remove the $\beta_0 + \beta_1 t$ to obtain a constant graph with fixed values at all times)

$$Y_t = \beta_0 + \beta_1 t + \varepsilon_t$$

In the case of "seasonal removal" for seasonal time series, there are 2 methods of "seasonal difference" and "time decomposition":

- The "seasonal difference" method is similar to the normal difference method but subtracts the current value at the current time from the value of the same time in the previous cycle.

$$Y'_t = Y_t - Y_{t-s}$$

  - In there s is the seasonal cycle, t is the time

- The "time decomposition" method will use these two methods to remove seasonal components from the series to make the series stationary.
  - Additive decomposition:

$$Y_t = T_t + S_t + E_t$$

  - Multiplicative decomposition:

$$Y_t = T_t \,.\, S_t \,.\, E_t$$

The case of "nonlinear transformation" which makes time series with variance changing over time can be handled by applying nonlinear transformations:

- The "Logarithm" method will help reduce the magnitude of the values and smooth out the fluctuations in the ratio. This method is suitable for time series with exponential growth trend data.

$$Y'_t = \log(Y_t)$$

- The "Square Root" method will reduce the influence of large values.

$$Y'_t = \sqrt{Y_t}$$

- The "Box-Cox Transformation" method will help generalize the transformation to reduce the variance, with the parameter:

$$Y'_t = \frac{Y_t^{\lambda}-1}{\lambda} \text{ (if } \lambda \neq 0) \text{ or } Y'_t = \log(Y_t) \text{ (if } \lambda = 0)$$

2.3 Autoregression (AR)

AR (Auto Regression): This emphasizes the dependent relationship between an observation and its preceding or 'lagged' observations [2]. It is understood that the time series is assumed to auto-regress linearly based on its own old (past) value sets. Usually, the autoregression model is often called the symbol p.

The model AR can be written in full equation like this:

$$y_t = c + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \cdots + \Phi_p y_{t-p} + \varepsilon_t$$

In there:

- ε_t: white noise
- c: instance, a value never changes.
- Φ_p: parameter of autoregression
- y_t: the value of the equation calculated at time t.

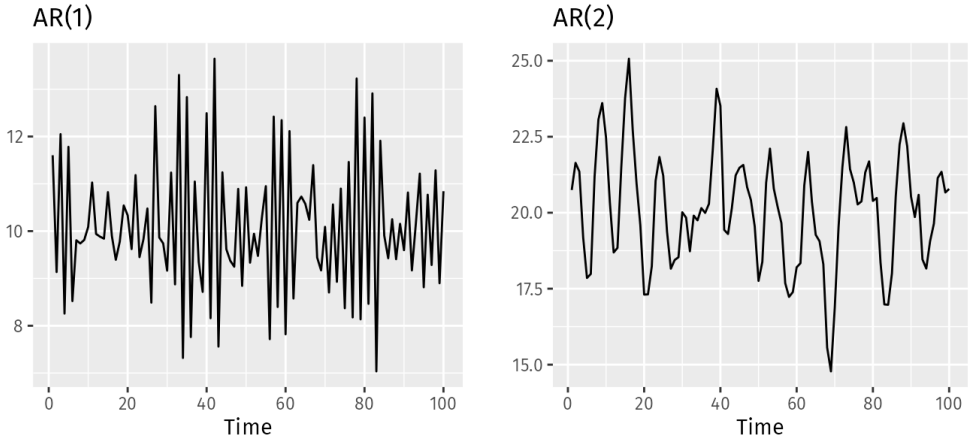

Figure 2.3.1: Two examples of data from autoregressive models with different parameters.

Autoregressive models are remarkably flexible at handling a wide range of different time series patterns [3]. By changing the value of the AR parameter (q), different results will be obtained.

2.4 Moving Average (MA)

MA (Moving Average): This component zeroes in on the relationship between an observation and the residual error from a moving average model based on lagged observations [2]. The number of past forecast errors included in the MA model of the time series is “regressed” on the past forecast error. Usually, the moving average model is often called the symbol q.
The model MA can be written in full equation like this:

$$y_t = c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \cdots + \theta_q\varepsilon_{t-q}$$

In there:

- ε_t: white noise
- c: instance, a value never changes.
- θ_q: parameter of autoregression
- y_t: the value of the equation calculated at time t.

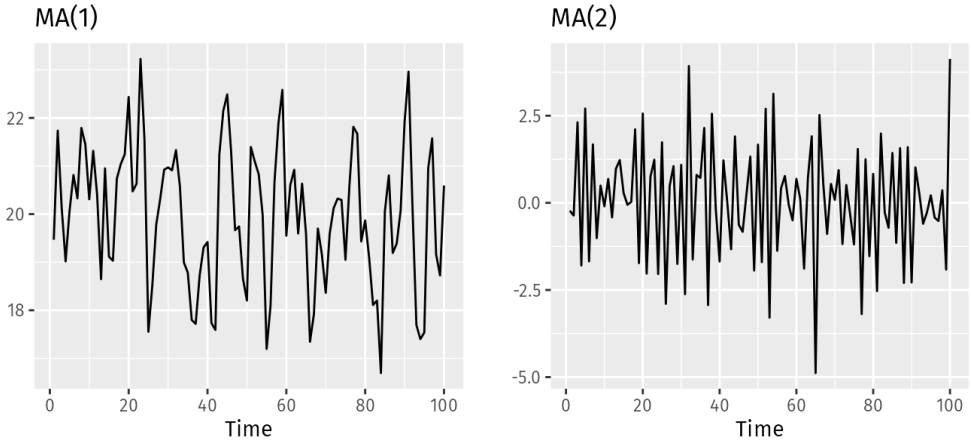

Figure 2.4.1: Two examples of data from moving average models with different parameters

2.5 Integrated (I)

I (Integrated): To achieve a stationary time series that doesn't exhibit trend or seasonality, differencing is applied. It typically involves subtracting an observation from its preceding observation [2]. Simply put, it calculates the number of times the difference between time series occurs until the series is truly stationary. Usually, the integrated model is often called the symbol d.
The model can be written in a full equation like this:

$$y'_t = y_t - y_{t-1}$$

In there:

- y_t: the value of the equation calculated at time t.

2.6 ARIMA Model

According to [1], ARIMA stands for Autoregressive Integrated Moving Average. It's a technique for time series analysis and forecasting possible future values of a time series. This model mainly uses stationary time series for analysis and forecasting. Still, if the time series is not stationary, it is difficult to apply and it needs to be converted to a stationary series before use.
The model ARIMA can be written in full equation like this:

$$y'_t = c + \Phi_1 y'_{t-1} + \cdots + c + \Phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

ARIMA has three components (or parameters) to work: AR (Auto Regressive), I (Integrated) and MA (Moving Average). These three components play an important role in building a Time Series model:

- **AR** (called by Auto-Regressive): *it's p in parameter*. This emphasizes the dependent relationship between an observation and its preceding or 'lagged' observations [2]. It is understood that the time series is assumed to auto-regress linearly based on its own old (past) value sets.
- **I** (called by Integrated): *it's d in parameter.* To achieve a stationary time series, one that doesn't exhibit trend or seasonality, differencing is applied. It typically involves subtracting an observation from its preceding observation [2]. Simply put, it is calculating the number of times the difference between time series occurs until the series is truly stationary.
- **MA** (called by Moving Average): *it's q in parameter.* This component zeroes in on the relationship between an observation and the residual error from a moving average model based on lagged observations [2]. The number of past forecast errors included in the MA model of the time series is "regressed" on the past forecast error.

In general, all three of these parameters affect model training and evaluation, especially the ARIMA model has many different variations from changing the indices in these parameters. As know if ARIMA model is made up of 3 small models (AR, I and MA), but what happens if set the parameter value to 0 for one or more of the ARIMA model components. Then will get a new model that is a variation of ARIMA but without the component whose parameter value is set to 0.
Suppose if set the parameter p to 0, that is, an ARIMA model without an AR component results in the model that applies having only an IMA component.
Suppose if set the parameter p (AR) to 0, it means that an ARIMA model does not have an AR component, leading to the model that applies having only an IMA component. Similarly, if the model only sets the parameter d (I) to 0, it means that the ARIMA model becomes an ARMA model. But what if both parameters are set to 0? The model will become extremely simple because the model will predict with a single constant value, representing the average value of the original data (if the time series is stationary).

Just as the ARIMA model does not have seasonal properties, if the time series intended for analysis does not have seasonal properties, this model is considered suitable and can be applied, but it is a prerequisite because if the time series has seasonal properties, instead of using this model, use the SARIMA model instead, which is considered a model that contains all the working properties of ARIMA but with the

addition of the prefix "S" (also known as Seasonal) before the model name "ARIMA", this SARIMA model allows working with seasonal time series. It is possible to completely apply the seasonal time series model to the ARIMA model, but to get the best performance, quality and optimization in deploying and training the model, it is recommended to use the SARIMA model. In some cases, the seasonal time series, when applied to the ARIMA model, gives a good evaluation, but in some cases, it does not, this also depends on many other factors.

2.7 Residuals and white noise

Residual means described as the difference between the actual value (y_i) and the predicted value ($\widehat{y_i}$), residual is defined as follows:

$$Residual = y_i - \widehat{y_i}$$

Residual is a measure of model error. It shows how well the model matches the actual data. Residual is not the actual error of the model on future data (that's the concept of error); it only measures how well the model matches the observed data.
Compare the difference between residual and error:

Residual	Error
It meaned: the difference between the actual value and the predicted value on training or test data.	It meaned: the difference between the actual and predicted values on future (unseen) data.
Can be calculated immediately when applying the model.	Unknowable (only happens when predicting on unseen data).
Residual depends on the observed values in the current dataset.	Errors related to generalization.

Residual analysis helps check the quality of the model and its underlying assumptions. Some things to check:

- **Average of residual**: Residual should have a mean close to 0, indicating that the model is not biased.
- **Distribution of residual:** Residual should be normally distributed in linear and time series models. If residual does not follow a normal distribution, the model may not be appropriate, or additional variables may need to be added to the model.
- **Pattern of residual:** Residuals should be randomly distributed, with no pattern. If there is a pattern, the model may not be capturing all the information in the data.
- **Homoscedasticity of residual:** Residual should have homoscedasticity. If the variance is heteroscedasticity, the model may not fit well.

White Noise is a random sequence of data in which: the expectation (mean) of the series is zero, constant variance and there is no autocorrelation between values at different times.
In time series analysis and model building, residuals are expected to behave like white noise if the model is well-fitted. This means that:

- **No more patterns in residuals:** If the residuals are white noise, it means that all the patterns or relationships in the data have been fully captured by the model. This proves that the model is well-fitted and there is no useful information left in the residuals.
- **Random distribution:** Residuals should be distributed around a mean of zero and there should be no correlation between data points.
- **Constant variance:** Residuals need to have homoscedasticity, which ensures that the model's errors do not change over time.
- **No autocorrelation:** Residuals should not have significant autocorrelation. If residuals still have autocorrelation, it means that the model does not capture all the relationships in the data.

To check the White Noise, there are some ways to do:

- **Using ACF of residuals plot:** If the residuals are white noise, the ACF plot of the residuals will have no significant autocorrelation values (all values fall within the confidence interval).

- **Ljung-Box test:** The Ljung-Box test is used to test whether there is significant autocorrelation in the residuals. Null hypothesis is residuals have no autocorrelation (white noise) and hypothesis 1 is residual have autocorrelation. If the p-value of test smaller significance level (normally is 0.5), then residuals are not white noise.
- **Analysis residuals plot:** Residuals should be randomly distributed around a mean value of zero, with no trend or pattern.

3 PYTHON SOURCE CODE

Link GitHub: https://github.com/TDMinhNhat/forecast_electric_vehicle_arima

4 IMPLEMENTATION

4.1 Data

The dataset used to analyze and forecast includes 12654 rows and 9 columns. This dataset talks about the revenue information of vehicles worldwide over the years. The dataset has information such as region, category, parameter (revenue type), mode (type of electric vehicle), powertrain, year, unit, value, and percentage. In this information, only focuses on two main data columns: year and value, because these two data columns are the premise for being able to apply the model.
Before visualizing the dataset, conducted EDA of the dataset by reformatting the data of the "value" column, forcing the data column to a numeric data type so that the model could be applied. After forcing the data type to a numeric type, some data was corrupted during the formatting process and became "null" data. To handle the "null" data, then deleted those null data rows to get a dataset that did not contain null data.
Finally, grouped the data of the "year" and "value" columns according to the "year" group and accumulated the "value" values after being grouped.

	year	value
0	2011	0.25
1	2014	0.50
2	2015	0.25
3	2016	3.50
4	2017	1.25
5	2018	0.75
6	2019	1.25
7	2020	1.75
8	2021	3.00
9	2022	1.50
10	2023	0.50
11	2030	0.75
12	2035	0.25

Figure 4.1.1: The dataset time – series before applying the training model

4.2 Visualize data

In the ARIMA model, it is especially important before applying model training to visualize the data to see whether the time series data set is truly stationary or not as well as the fluctuations in high and low values over time periods. Applying a data set with a truly stationary time series is essential for the ARIMA model to be able to predict as accurately as possible.

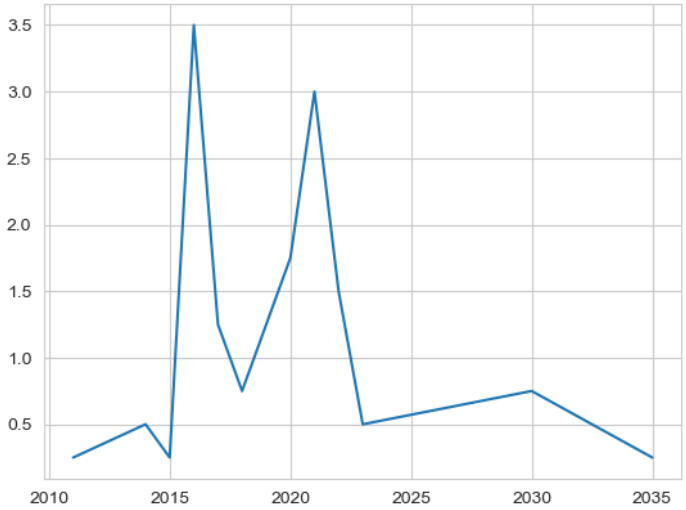

Figure 4.2.1: Visualize the time – series of the EV Sales for each year from 2011 to 2035

Based on the visual chart, it can be seen that the time series data set shows signs of strong ups and downs, but after 2025, the time series shows signs of increasing again, but a little until 2030, then it decreases quite deeply, which leads to the time series data set showing signs of a decreasing trend for the following years. After visualizing the model, divided the data set into a train set and a test set according to the conditions; for the train set, took data with years less than 2023 and before, and the test set took data from 2023 and after. From there, the model on the time period of the test set after training the model with the train set, from which could compare and evaluate the quality of the model after application.

4.3 Check time-series stationary and make the time-series stationary

Before applying the machine learning model, one needs to check whether the training data set is really a stationary time series or not? To be able to check the time series that is used, decided to use the "adfuller" function from the "statsmodels.tsa.stattools" library package.
Using statistical hypotheses to be able to test the results of testing the stationarity of the time series. Suppose set the null hypothesis that the graph series is not stationary and hypothesis 1 that the graph series is stationary. If the statistical value p-value is greater than or equal to 0.5, it means we reject hypothesis 1 and take the null hypothesis and vice versa, if the statistical value p-value is less than 0.5, it means then rejects the null hypothesis and accept hypothesis 1 (choose the significance level of 5%).

```
(-2.781185328699488,
 0.061011975232483,
 1,
 9,
 {'1%': -4.473135048010974,
  '5%': -3.28988060356653,
  '10%': -2.7723823456790124},
 19.272979508238105)
```

Figure 4.3.1: Checking stationary of time-series dataset in the first time

In figure 4.3.1, it can be seen that when using the "adfuller" function to check whether the time-series is stationary, the result returned by a function is a list of value information, especially in the p-value information located in the first element (understood as the index in the tuple) gives an approximate result of 0.061, which shows that the p-value is now greater than 0.05, which means that the test rejects hypothesis

1 and accepts the null hypothesis. This leads to the fact that training data set is not really at the stationary level, it is necessary to improve to turn a graph from a non-stationary series into a stationary series.
To turn a non-stationary series into a stationary series, takes the first-order difference and then conducts the test.

```
(-3.322376869678028,
 0.013907381219126266,
 3,
 6,
 {'1%': -5.354256481481482,
  '5%': -3.6462381481481483,
  '10%': -2.901197777777778},
 13.386246362519806)
```

Figure 4.3.2: Checking stationary of time-series dataset after taking the first-order difference

In figure 4.3.2, The information about the statistical value of p-value at this time gives the result of 0.01 which is completely smaller than 0.05, from which concludes that after taking the first difference of the time series data set, then now have a new time series data set that is stationary after taking the difference. From there, proceeded to use this differenced data set to train the model.

4.4 Training model and compare between the forecast dataset and the test dataset

Before training the model, it is necessary to determine the parameters for the small models in the ARIMA model such as AR (p), I (d) and MA (q). First, determine I (parameter d) by relying on the number of times the difference is taken in the training data set. In the previous section, to make this data set stationary, it was taken only at the first difference, which made the model stationary, so the parameter d was assigned the value 1. In the case of determining AR (parameter p) and MA (parameter q), it is necessary to rely on a type of graph, which is the autocorrelation graph and the partial autocorrelation graph, from which it is possible to determine the appropriate values to assign to the remaining two parameters.

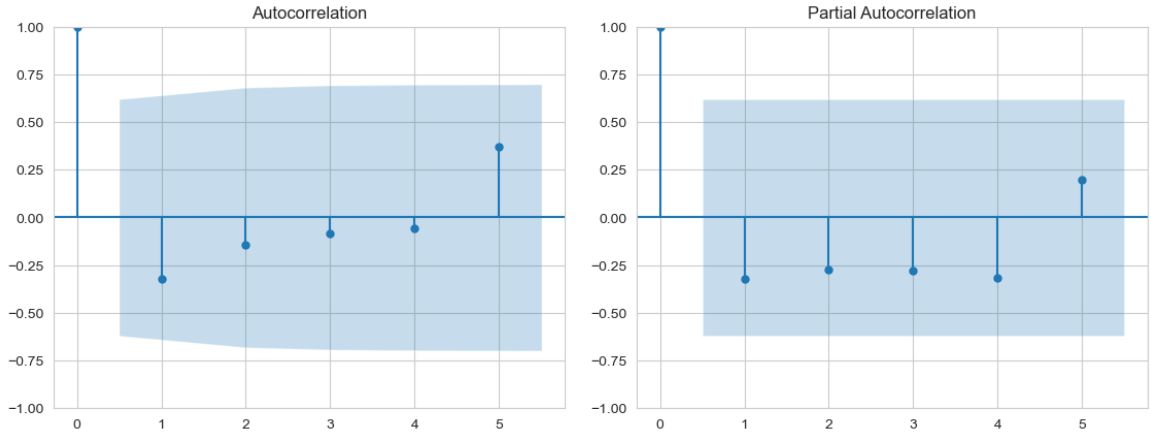

Figure 4.4.1: The ACF (Autocorrelation Function) plot and PACF (Partial Autocorrelation Function) plot

In figure 4.4.1 (left), the AR model (ie the parameter p) will rely on the autocorrelation graph (ACF) to be able to determine, it can be determined that the lags have signs of strong decay and the correlation relationship between the lags is not there, only occurring in the first lag (ie position 0 on the graph) and the 6th lag (ie position 5 on the graph) has quite strong correlation signs. It can be determined from the graph that the value of the AR model (ie the parameter p) will be 0 or 5.
Similarly, for the MA model (ie the q parameter) will rely on the partial autocorrelation graph (PACF) to be able to determine, it can be determined that the lags have signs of strong decay (like the autocorrelation graph) and the correlation relationship between the lags is not there, only occurs in the first lag (ie position

0 on the chart) and the 6th lag (ie position 5 on the chart) has signs of quite strong correlation. It can be determined from the graph that the value of the MA model (ie the q parameter) will also be 0 or 5.
After selecting the parameters, the next step is to train the ARIMA machine learning model, let the model learn data on the train set, from which the model will predict the results and from that result can be compared with the test set, to see if the difference between the two forecast data sets and the test data set is really too far apart?

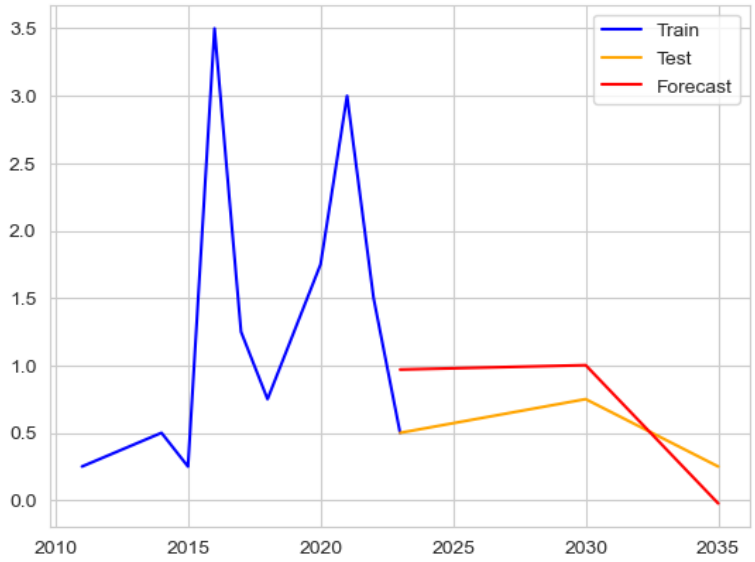

Figure 4.4.2: Training model on train dataset and compare the predict and test dataset

Follow the figure 4.4.2, after training the model based on the train set and predicting the model on the time period of the test set, and visualizing it with a diagram, it can be seen that the model has a fairly good fit, not too deviated when both sets predict a downward trend.
After applying the model to forecast the feature for the time test dataset, let's check the residuals

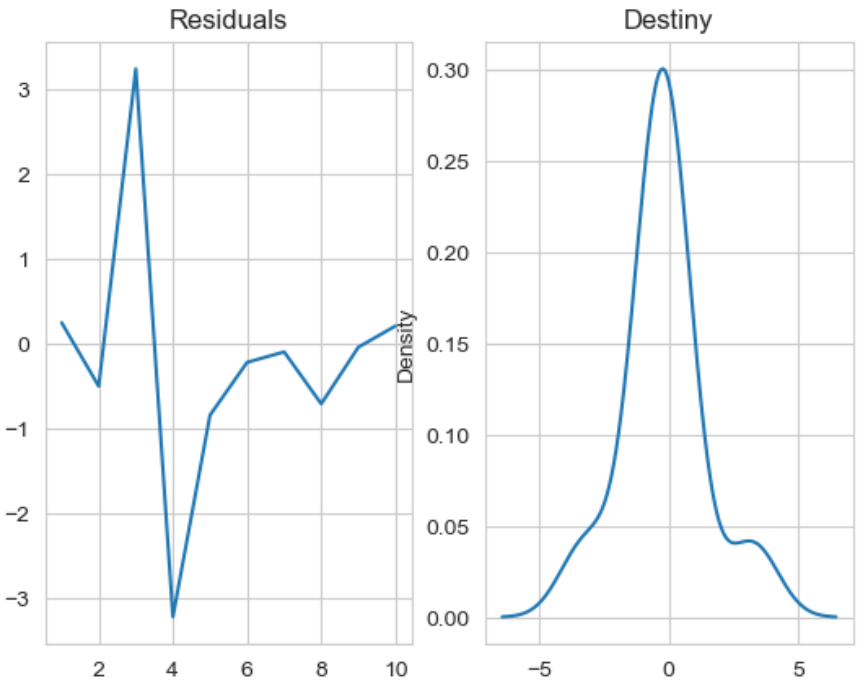

Figure 4.4.3: Visualize the residual dataset plot

For the residuals plot (left image). About the Oscillation shape: the residuals fluctuate around the 0 axis, which is normal if the model is performing well and there is no clear trend. However, the fluctuations are quite large at some points (for example, the 4th value has the largest residual, which is negative), indicating that some points have significant errors.
For the destiny plot (right image): The density plot shows that most of the residuals are concentrated near zero, indicating that the model has good predictive ability on most of the data. However, some residuals are quite far from the center, making the distribution plot not completely normal. This could be due to outliers or problems in the data/model.

4.5 Forecast

The final step is to predict the future using the model. In this step, instead of running the model over the time period of the test set, the model predicts further, specifically the next 5 years of the latest year of the test set, from which it is possible to evaluate the future development trend of electric vehicles.

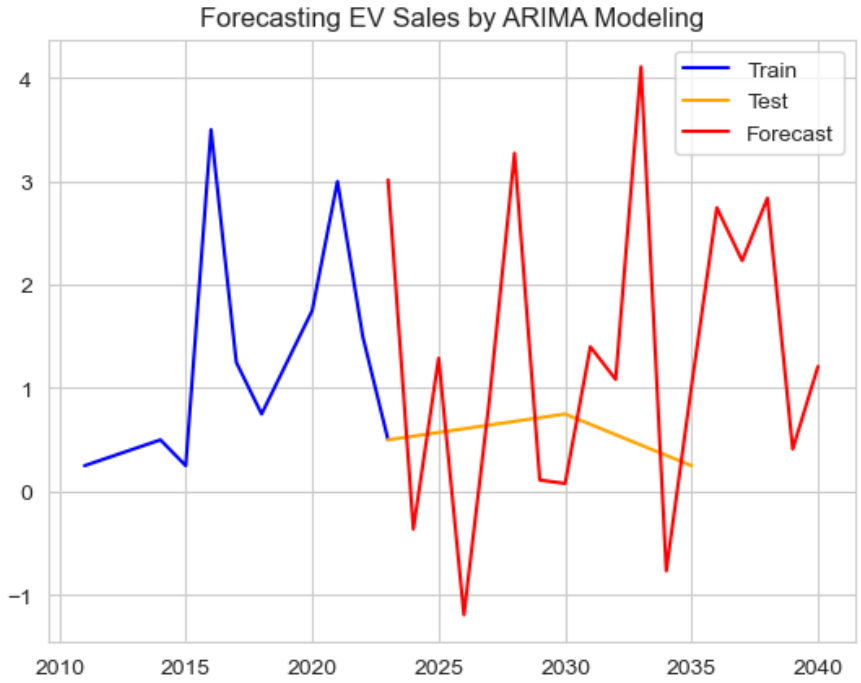

Figure 4.5.1: Visualize the forecast dataset after the training model

In figure 4.5.1, It can be seen that the forecasting model shows that the future trend of electric vehicles is fluctuating, fluctuating continuously, not following any specific direction.
After applying the model to forecast, let's check the residuals

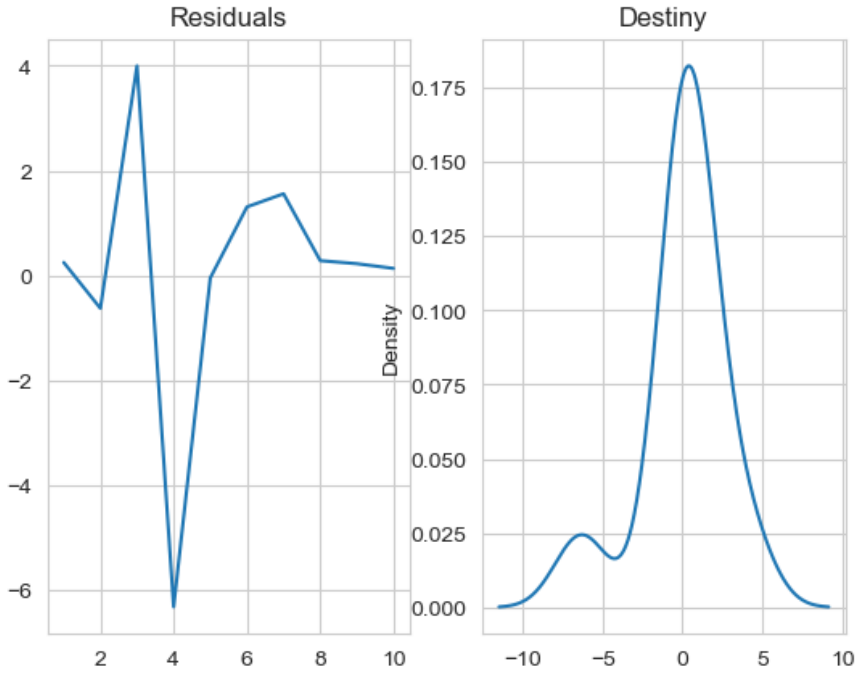

Figure 4.5.2: Visualize the residual dataset after forecasting

For the residuals plot (left image). Larger range of variation - Residuals vary more widely (between -6 and 4), indicating that there are many data points where the model is significantly off. The point with the largest residual is around the 4th value, indicating that the model is having difficulty with some specific data. Uneven trend - Residuals tend to be irregular, not regularly lying around the zero axis. In particular, large and sudden variations at some points indicate the possibility of outliers or complex features in the data that the model does not capture.
For the destiny plot (right image). Non-normal distribution - The distribution of residuals is not symmetrical, with many values far from the center (between -10 and 10), indicating many large residuals. A peak at 0 indicates that most of the residuals are close to the center, but the two sides of the distribution have longer tails, indicating outliers or instability in the prediction. Multi-peak phenomenon - The density plot shows more than one peak, suggesting that the data may belong to different groups or that the model is not good enough to accurately predict all groups.

5 CONCLUSION AND DEVELOPMENT DIRECTION

Through the research process as well as the application and realization of the ARIMA machine learning model in predicting future development trends, in general, the model has not predicted completely as expected, but the model has provided an insight into the development of electric vehicles, which is almost constantly fluctuating up and down erratically, making it difficult to maintain progress and excellence over the years, and there will be times when it will decline. However, to achieve the most effective and complete model, more in-depth research is needed, using advanced techniques to be able to bring about the most optimal expected results possible.

REFERENCES

[1] Joshua Noble. (2024) Introducing ARIMA models [Online]. Available: https://www.ibm.com/topics/arima-model

[2] Jason Brownlee. (2023) How to create an ARIMA Model for Time Series Forecasting in Python [Online]. Available: https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python

[3] Rob J Hyndman and George Athanasopoulos. (2018) Forecasting: Principles and Practice (2nd ed) [Online]. Available: https://otexts.com/fpp2/