Time Series Analysis of UK-South Korea Exports

1. Introduction

- **Objective**: To forecast South Korea's monthly exports to the UK using historical data and time series analysis techniques. This will help in understanding export trends and making informed decisions.
- **Background:** The main products that South Korea exported to the United Kingdom were Cars ($2.25B), Broadcasting Equipment ($563M), and Platinum ($295M).[1]

2. Methodology

- **Data Loading and Pre-processing**:
    - Load the data from a CSV file, filter relevant entries using pattern recognition to only obtain monthly export data, convert date formats and set the date as the index of the dataframe.
    - Handle missing or infinite values by forward and backward filling.

```
                Value
Time Period
1997-01-01   109.0
1997-02-01    98.0
1997-03-01    90.0
1997-04-01   101.0
1997-05-01   118.0
...            ...
2024-01-01   290.0
2024-02-01   352.0
2024-03-01   329.0
2024-04-01   270.0
2024-05-01   532.0

[329 rows x 1 columns]
```

Figure 1: Summary of Filtered Dataframe

[1] https://oec.world/en/profile/bilateral-country/kor/partner/gbr

- **Time Series Analysis**:
  - **Visualisation**: Plot the time series data to visualise the trends. The graph shows an overall increase in exports over time.

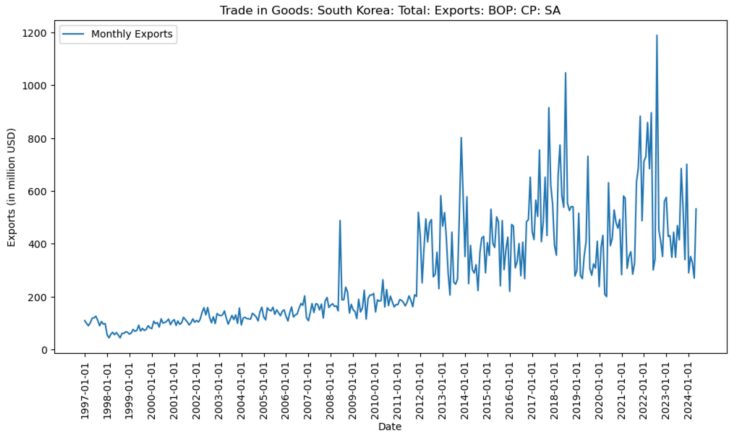

Figure 2: Visualisation of Time Series

  - **Decomposition**: Decompose the time series into trend, seasonal, and residual components using a multiplicative model. As the seasonal variations in the data change proportionally with the level of the time series (as shown by increasing variance over time), the multiplicative model seems to be more appropriate as compared to the addictive model.
    - The multiplicative model seems more appropriate for this data because the deseasonalized plot using the multiplicative model maintains the overall structure of the original data, indicating that the model accurately adjusts for the proportional seasonal effect.

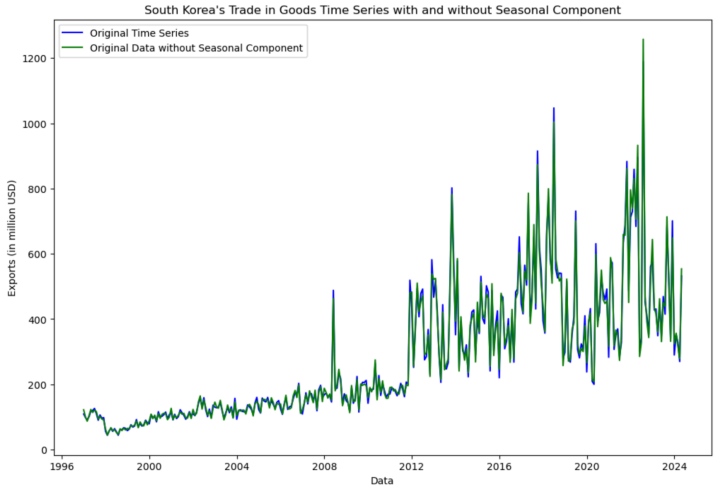

Figure 3: Time Series with and without Seasonal Component (Multiplicative)

- The addictive model does not seem appropriate as the green line appears to have a more stable trend component, indicating that seasonality has been effectively isolated.

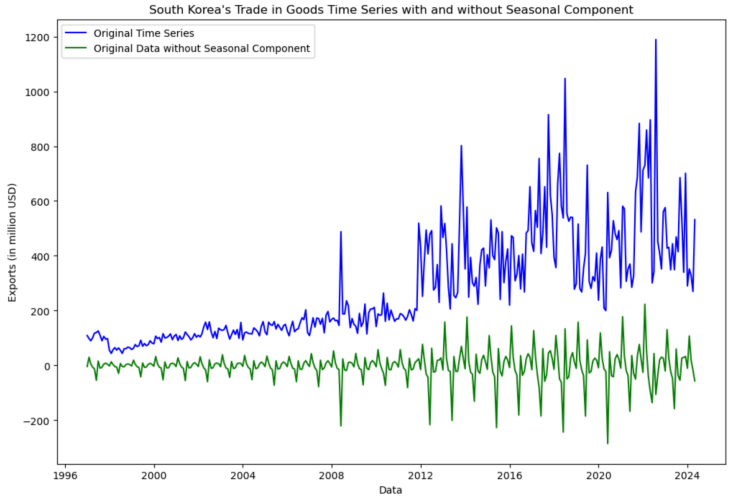

Figure 4: Time Series with and without Seasonal Component (Addictive)

- **Deseasonalisation**: Remove the seasonal component to focus on the trend and residual components.

Time Period	Value
1997-01-01	121.862383
1997-02-01	99.237113
1997-03-01	87.057439
1997-04-01	104.313516
1997-05-01	122.809699
...	...
2024-01-01	324.221019
2024-02-01	356.443507
2024-03-01	318.243305
2024-04-01	278.857914
2024-05-01	553.684404

Figure 5: Deseasonalised Data

- **Calculating Percentage Change:** Building a model based on percentage changes instead of raw price data to simplify achieving stationarity and improve model performance by focusing on relative movements rather than absolute levels. This approach reduces noise and enhances interpretability, making forecasts more robust and insightful.

Time Period	Value	Pct_change
1997-01-01	121.862383	NaN
1997-02-01	99.237113	-0.185662
1997-03-01	87.057439	-0.122733
1997-04-01	104.313516	0.198215
1997-05-01	122.809699	0.177313
...	...	...
2024-01-01	324.221019	-0.500535
2024-02-01	356.443507	0.099384
2024-03-01	318.243305	-0.107170
2024-04-01	278.857914	-0.123759
2024-05-01	553.684404	0.985543

[329 rows x 2 columns]

Figure 6: Deseasonalised Data with Percentage Change

- **First Stationarity Testing**: Conduct Augmented Dickey-Fuller (ADF) and KPSS tests on percentage changes to check for stationarity.
  - The ADF test recorded a p-value of 1.29e^-22 (<0.05), which suggests that the data is stationary. However, the KPSS test returned a p-value of 0.01 (<0.05), which suggests that the data is non-stationary.

```
ADF Statistic: -12.190829111445796
p-value: 1.2869516713121955e-22
Critical Values:
   1%, -3.450695263332383
Critical Values:
   5%, -2.87050218926466
Critical Values:
   10%, -2.5715449066453284

KPSS Statistic: 1.0454362080799753
p-value: 0.01
Critical Values:
   10%, 0.347
Critical Values:
   5%, 0.463
Critical Values:
   2.5%, 0.574
Critical Values:
   1%, 0.739
```

Figure 7: Results of First Stationarity Tests

- **Differencing**: Apply first-order differencing to the percentage change to obtain percentage change difference (Pct_change_Diff) and to achieve stationarity.

Time Period	Value	Pct_change	Pct_change_Diff
1997-03-01	87.057439	-0.122733	0.062929
1997-04-01	104.313516	0.198215	0.320948
1997-05-01	122.809699	0.177313	-0.020901
1997-06-01	113.790125	-0.073444	-0.250757
1997-07-01	120.832934	0.061893	0.135336
...	...	...	...
2024-01-01	324.221019	-0.500535	-1.457496
2024-02-01	356.443507	0.099384	0.599920
2024-03-01	318.243305	-0.107170	-0.206555
2024-04-01	278.857914	-0.123759	-0.016588
2024-05-01	553.684404	0.985543	1.109302

Figure 8: Deseasonalised Data with Percentage change and Percentage change Difference

- **Second Stationarity Testing**: Conduct Augmented Dickey-Fuller (ADF) and KPSS tests again on percentage change difference to check for stationarity.
  - Both tests suggest that the data (Pct_change_Diff) is stationary. The ADF test recorded a p-value of 3.88e^-15 (<0.05) and the KPSS test returned a p-value of 0.1 (>0.05).

```
ADF Statistic: -9.088554994428192
p-value: 3.8800657076006085e-15
Critical Values:
   1%, -3.451552879535732
Critical Values:
   5%, -2.8708786756338407
Critical Values:
   10%, -2.571745666091128

KPSS Statistic: 0.07922086727643882
p-value: 0.1
Critical Values:
   10%, 0.347
Critical Values:
   5%, 0.463
Critical Values:
   2.5%, 0.574
Critical Values:
   1%, 0.739
```

Figure 9: Results of Second Stationarity Tests

- **Model Building and Forecasting**:
  - **Splitting into Train and Test Sets:** Divided the dataset into training (291) and testing (36) sets based on Pct_change_Diff.
  - **ARIMA Model**: Utilised `auto_arima` to determine the best ARIMA parameters (p, d, q) and fit the model.
  - **Rolling Walk Forward Forecast**: Implemented a rolling forecast approach where the model is updated with ARIMA parameters each loop to predict the next month's exports.

```python
def rolling_walk_forward(data, model_function, forecast_function, steps):
    predictions = []
    history = [x[0] for x in data]
    for t in range(steps):
        # using auto_arima to determine p,d,q for each time I forecast for the next month using ARIMA
        auto_model = auto_arima(history, start_p=1, start_q=1,
                      max_p=10, max_q=10,
                      d=1,
                      seasonal=False,
                      start_P=0,
                      D=None,
                      trace=True,
                      error_action='ignore',
                      suppress_warnings=True,
                      stepwise=True)
        p,d,q = auto_model.get_params().get("order")
        model_fit = model_function(history,p,d,q)
        forecast = forecast_function(model_fit, steps=1)
        predictions.append(forecast[0])
        history.append(forecast[0]) # adding forecast to history to prepare for next forecasting
        history.pop(0) # removing first month of history
    return predictions
```

Figure 10: Code for Rolling Walk Forward Forecast

- **Inverse Transformation**: Converted the ARIMA predictions back to percentage change and then to price form, and subsequently 'un-deseasonalised' the price form values for evaluation of model.

Time Period			
2021-06-01	376.276498	2022-12-01	750.474228
2021-07-01	406.702349	2023-01-01	632.451481
2021-08-01	395.524915	2023-02-01	725.447213
2021-09-01	363.512562	2023-03-01	793.473718
2021-10-01	442.232221	2023-04-01	764.418611
2021-11-01	445.984877	2023-05-01	791.208422
2021-12-01	469.373538	2023-06-01	897.946333
2022-01-01	442.071168	2023-07-01	918.166518
2022-02-01	489.790801	2023-08-01	868.208046
2022-03-01	500.143160	2023-09-01	910.297311
2022-04-01	504.568845	2023-10-01	1029.374037
2022-05-01	507.990828	2023-11-01	1046.506051
2022-06-01	588.226782	2023-12-01	1137.999542
2022-07-01	602.095586	2024-01-01	980.148858
2022-08-01	554.053432	2024-02-01	1121.338117
2022-09-01	599.435580	2024-03-01	1214.955290
2022-10-01	671.756246	2024-04-01	1180.734539
2022-11-01	677.861227	2024-05-01	1212.903451

Figure 11: Forecasted values in price form (un-deseasonalised)

- **Evaluation**:
  - Comparing the forecasted values with actual values, the Mean Absolute Error (MAE) value was 373.256 and R-squared (R^2) value was -3.339.
    - A negative R-squared value suggests that this model indicates that the model is performing worse than a simple horizontal line (i.e., the mean of the data). This suggests that the model's predictions are not capturing the patterns in the data well.
  - Plot actual vs. predicted values for visual comparison.

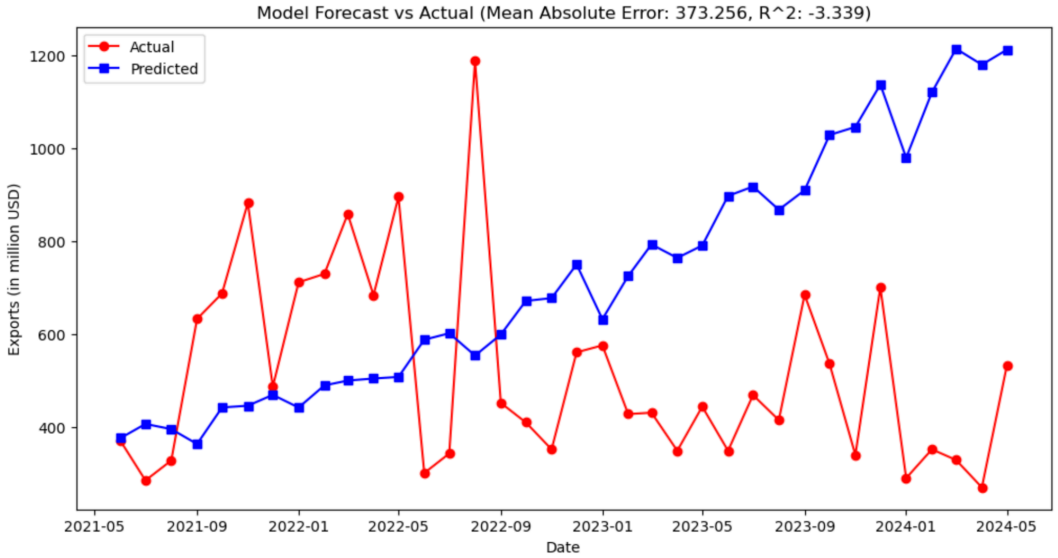

Figure 12: Visualisation of Forecast vs Actual value in price form by ARIMA Model

3. Key Takeaways

- **Model Performance**: Given that the values range from 80 to 600, an MAE of 373.256 represents a significant error, as it is more than half of the maximum value. Coupled with a negative R^2 value of -3.339, it suggests that the model's predictions are not very accurate.
- **Trends and Patterns**: The analysis revealed that while seasonal effects were effectively modelled using a multiplicative approach, the model struggled to account for other underlying factors influencing exports. This suggests that besides seasonality, additional elements such as economic shocks, policy changes, or market dynamics might be affecting export trends.
- **Alternative Modelling**: Based on the results of the model, alternative models should be considered to forecast this data.
    - ARIMA Model on Absolute Percentage Change: By taking the absolute value of Pct_change, it removes the 'directional' (ie. positive or negative) component of the data and solely focuses on the amplitude of change month on month. The hypothesis is that this alternate ARIMA model will be more accurate in predicting future values than the above ARIMA model.
        - Using the same functions as the ARIMA model above, the alternate ARIMA model performed be slightly more accurate with a MAE of 0.272 , and R^2 value of -0.081. However, predictions were relatively flat with no significant changes month on month.

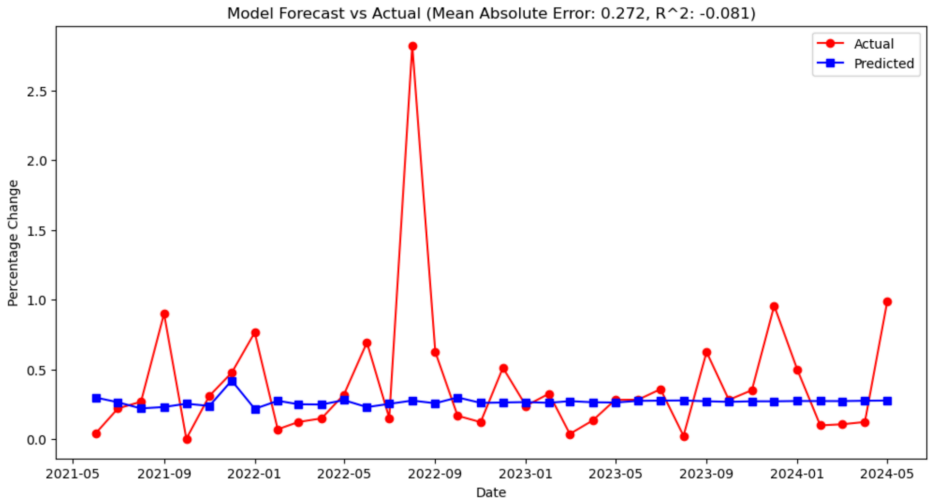

Figure 13: Visualisation of Forecast vs Actual value in percentage change by ARIMA Model

- Simple Regression Model: The regression model used for predictions seems to capture the seasonality better than the two ARIMA models above, and has a greater accuracy in predicting the values with a MAE of 0.363 and R^2 value of 0.055.

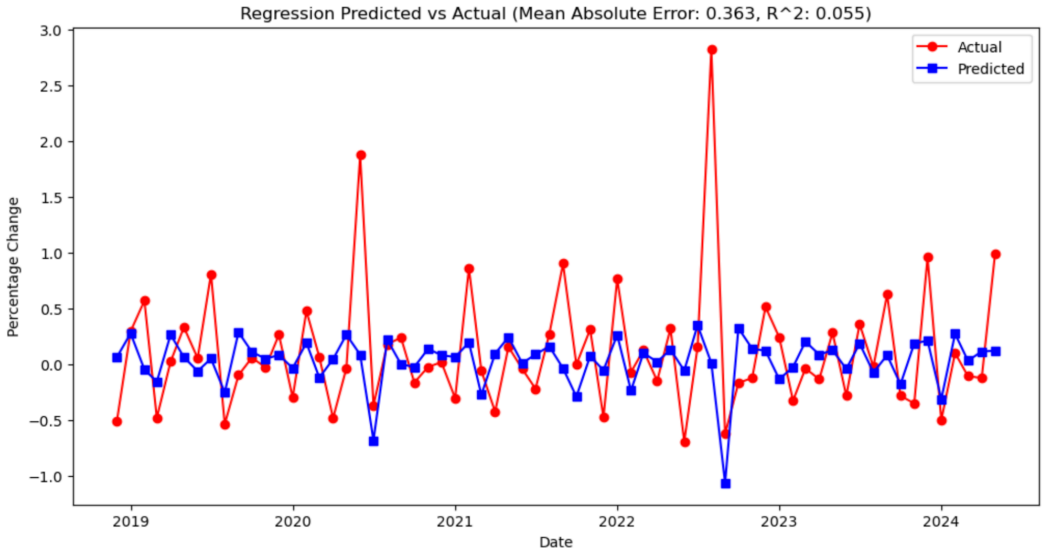

Figure 14: Visualisation of Predicted vs Actual value in percentage change by Regression Model

4. Overall Conclusions

- **Model Limitations:** Although the ARIMA models captured the seasonality of the data, it showed limitations in capturing the other complex patterns of the data, as evidenced by high MAE and poor R^2 value.
- **Alternative Approaches:** Exploring other models, including regression with additional features or advanced machine learning techniques, might offer improved forecasting performance.