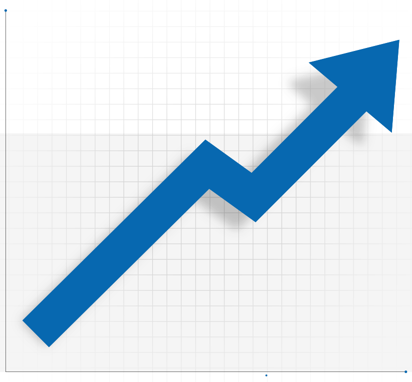
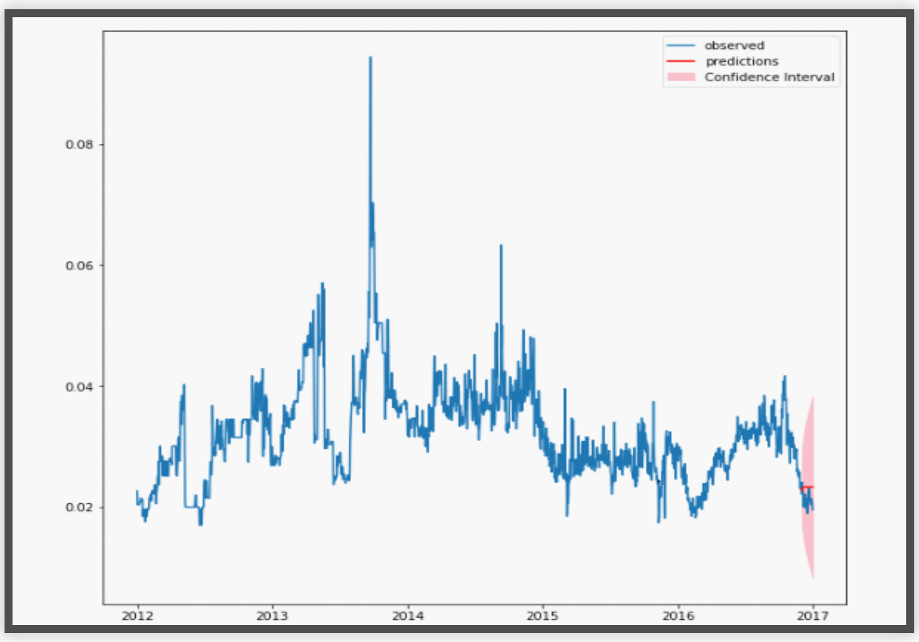
nikoways



Daily input SKU level price forecasts with robust Econometric Time Series Modelling

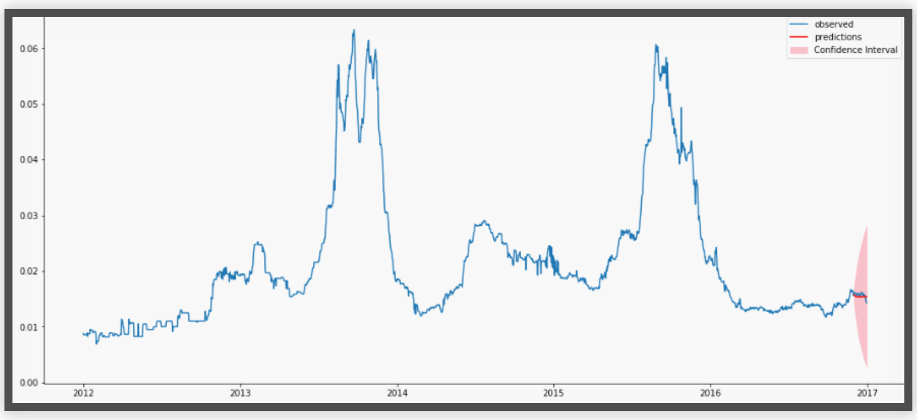
Problem Statement

- Having a forecast of input prices on a daily basis is key to success for our ecommerce client from operating cash flow, pricing, and inventory standpoint.



Our Solution

- We developed an accurate and robust time series econometric forecasting model using past years' daily data for multiple SKUs that forecasts 30/60/90 day ahead input price forecast at SKU level.
- With the accurate forecast, the client was able to optimize cash flow and inventory, and improve bottom line profitability.



Client Benefits

Cost
Reduce uncertainty in input costs.

Easy to deploy
Easy to use, our models can be easily deployed by clients.

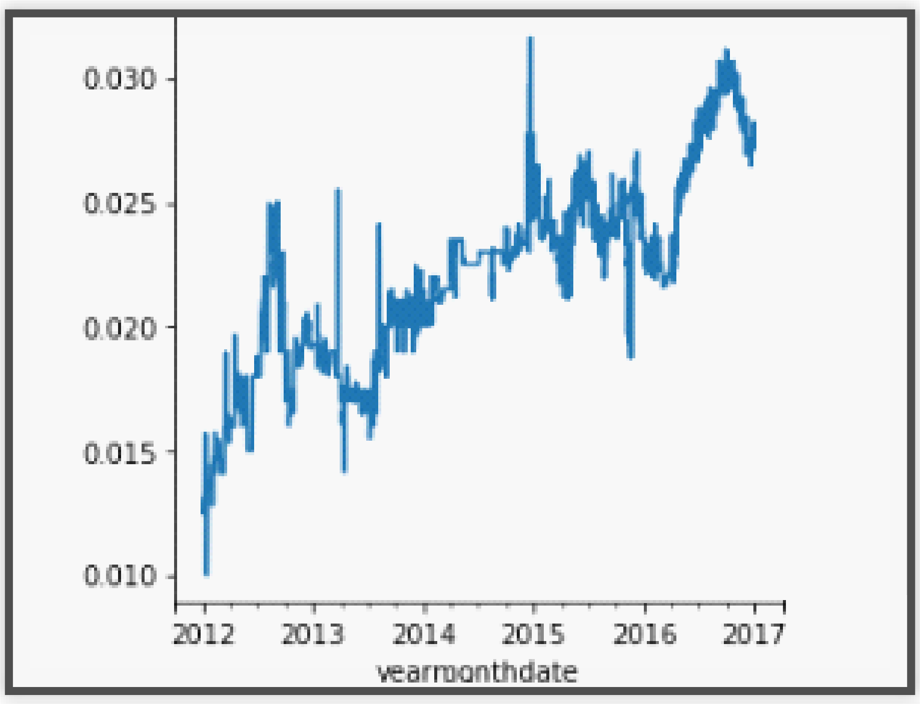
Accuracy
Achieve as high as **90%** accuracy with our optimized forecasting models.

What we can do

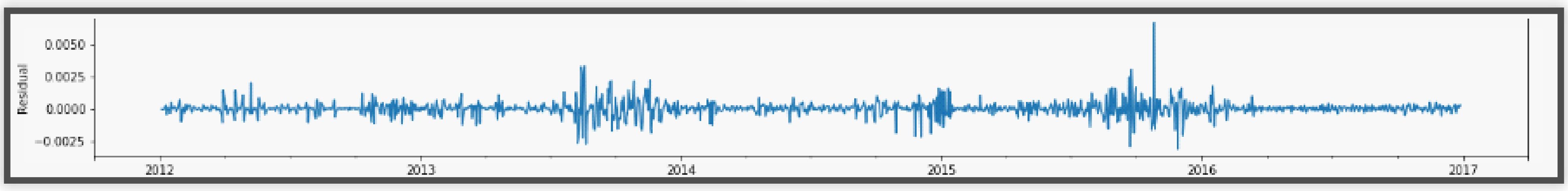
Develop customized time series forecasting models for one or more quantities of interest.

Deliver full forecasting solution for deployment either in-cloud or on-premises.

Maintain and periodically re-train the forecasting solution.

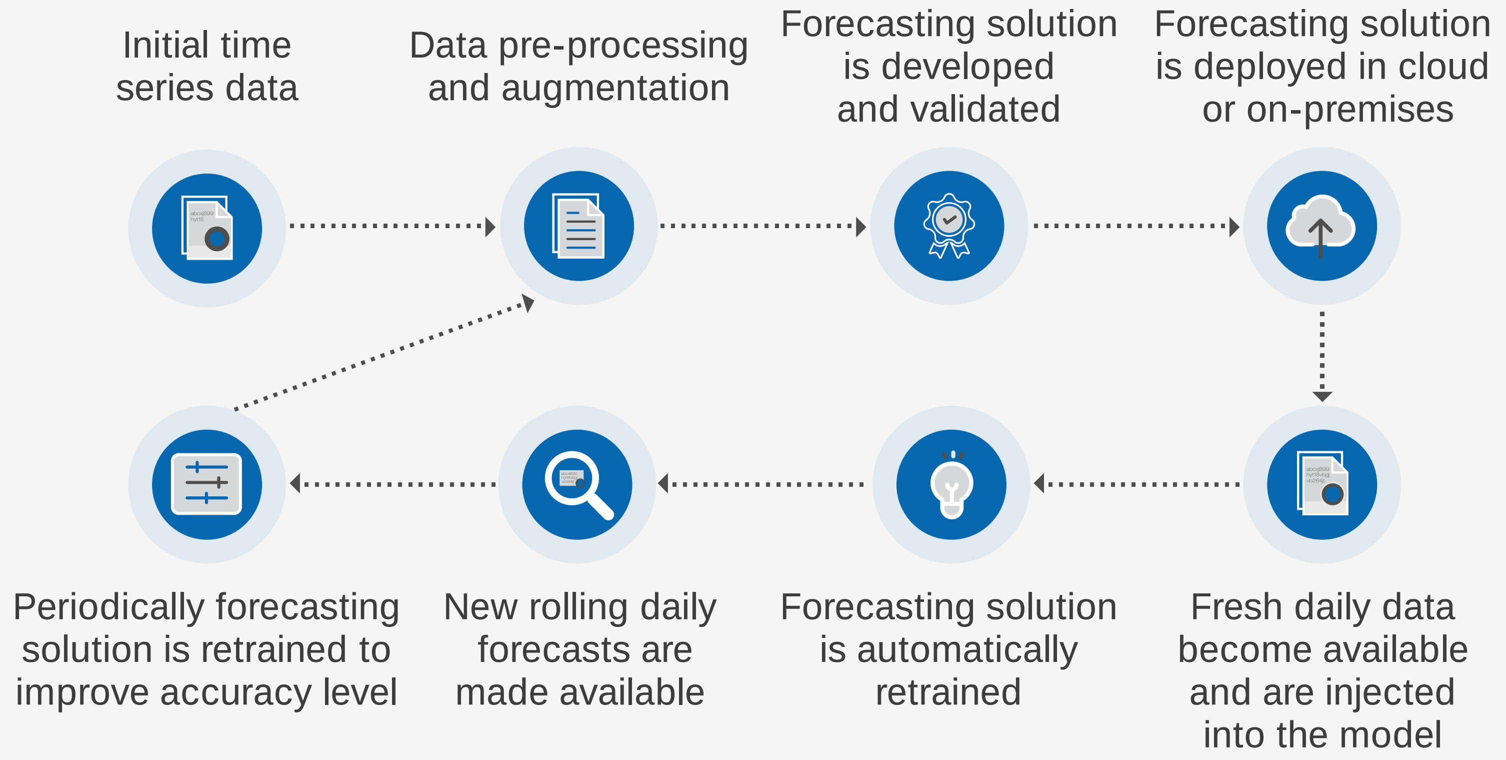



```
def get_forecast(selected_sku):
    #Obtaining the normalizing factor for the SKU
    normalizing_factor_sku = dict_prod_factor[selected_sku]
    print("Normalizing factor:",normalizing_factor_sku)

    #Obtaining the product group
    sku_prod_group = get_prod_group(selected_sku)
    print("Product Group:",sku_prod_group)

    p,d,q = get_pdq(sku_prod_group)

    if p==0 and d==0 and q==0:
        print("No model at the moment")
        return False

    print("Model Parameters:")
    print("p:",p)
    print("d:",d)
    print("q:",q)

    temp_df_exog = df_final[df_final['PROD_GROUP']==sku_prod_group].loc[:,columns_needed].copy()
    #Resampling
    temp_df_exog = temp_df_exog.resample('D').mean()

    #Calling function to plot the decomposition plot
    plot_decomposition(temp_df_exog['Normalized Price'],sku_prod_group)

    #Obtaining ADF Results
    adf(temp_df_exog['Normalized Price'])

    predictions = forecast_model(temp_df_exog,p,d,q,sku_prod_group)

    cost_price_predictions = np.array(predictions)/normalizing_factor_sku
    return cost_price_predictions
```



yearmonthdate										
2012-01-01	142.036391	148.771552	133.040738	127.442807	127.683826	124.047226	119.9	57.8	15.2	24.5
2012-01-01	142.036391	148.771552	133.040738	127.442807	127.683826	124.047226	119.9	57.8	15.2	24.5
2012-01-01	142.036391	148.771552	133.040738	127.442807	127.683826	124.047226	119.9	57.8	15.2	24.5
2012-01-01	142.036391	148.771552	133.040738	127.442807	127.683826	124.047226	119.9	57.8	15.2	24.5
2012-01-01	142.036391	148.771552	133.040738	127.442807	127.683826	124.047226	119.9	57.8	15.2	24.5

Process Flow



www.nikoways.com   +1 (650) 285-1473   info@nikoways.com