Case Study

Ta-Feng grocery dataset model to forecast the sales of any item in the inventory

HELLO!

I am Gurpreet Singh

I have strong passion for data, ML and predictive modeling

gurpreet.sachdeva@gmail.com

- Problem Statement
- Test Harness
- Persistence
- Data Analysis
- Model Validations

Problem Statement

Analyse the point-of-sale data from Ta-Feng Grocery and train a model to predict the monthly sale of any item.

Assumptions:

- Since the data is very less (4 months) this case would be a good representation of the approach rather than best performing model
- The data looked like multivariate time series but then the correlation between different data attributes were not known.
- The effect of external environmental factor such as marketing, seasonality and trend was not very clear
- Demographics of the users were not clear except their age and area
- Since the dataset is small, instead of considering months, I have taken week as the cyclic period

Training Flow

Technical Environment

- The code should work with both Python 2.x and 3.x, I have tested it with 2.7.12
- Required Python packages
  - Numpy
  - Scipy
  - Matplotlib
  - Pandas
  - Scikit
  - Statsmodels

Test Harness

- I have splitted the training and testing data set in 2:1 ratio.
- That means 66% of Ta-Feng data was used to train the model and rest 33% to test.
- Train-Test split was done such that they respect the temporal order of observations.
- For validation I used Walk-Forward Validation.
- I used RMSE to evaluate the performance of predictions.
- This gives more weight to predictions that are extremely wrong.

Training a baseline prediction model

- Model used for making predictions is Autoregressive Integrated Moving Average (ARIMA)
- Analysis of the time series data shows that it is non-stationary data
- I tried to make it stationary by subtracting the observation from the same time in the previous cycle
- Ran the augmented Dickey-Fuller test to verify that the series is stationary
- A plot of the differenced dataset is also created and it seems to be a good starting point for modeling
- Used Grid Search to find the optimal values of p,d and q (ARIMA Hyperparameters)

Data Analysis – Data Snapshot

```
In [177]: pos_data = read_data()
          pos_data.head(10)
```

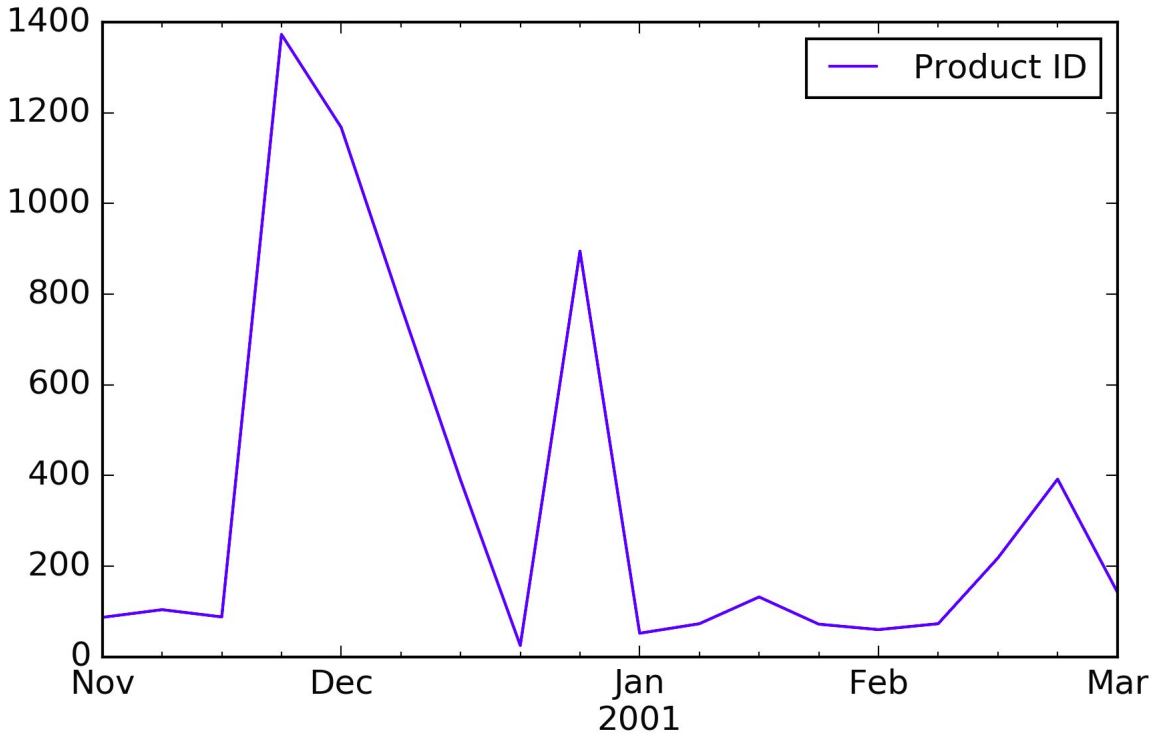
Out[177]:

Transaction Date	Customer ID	Age	Residence Area	Product Subclass	Product ID	Amount	Asset	Sale Price
2000-11-01	46855	D	E	110411	4710085120468	3	51	57
2000-11-01	539166	E	E	130315	4714981010038	2	56	48
2000-11-01	663373	F	E	110217	4710265847666	1	180	135
2000-11-01	340625	A	E	110411	4710085120697	1	17	24
2000-11-01	236645	D	H	712901	8999002568972	2	128	170
2000-11-01	1704129	B	E	110407	4710734000011	1	38	46
2000-11-01	841528	C	E	110102	4710311107102	1	20	28
2000-11-01	768566	K	E	110401	4710088410382	1	44	55
2000-11-01	217361	F	E	130401	4711587809011	1	76	90
2000-11-01	2007052	D	E	110504	4710323168054	1	17	20

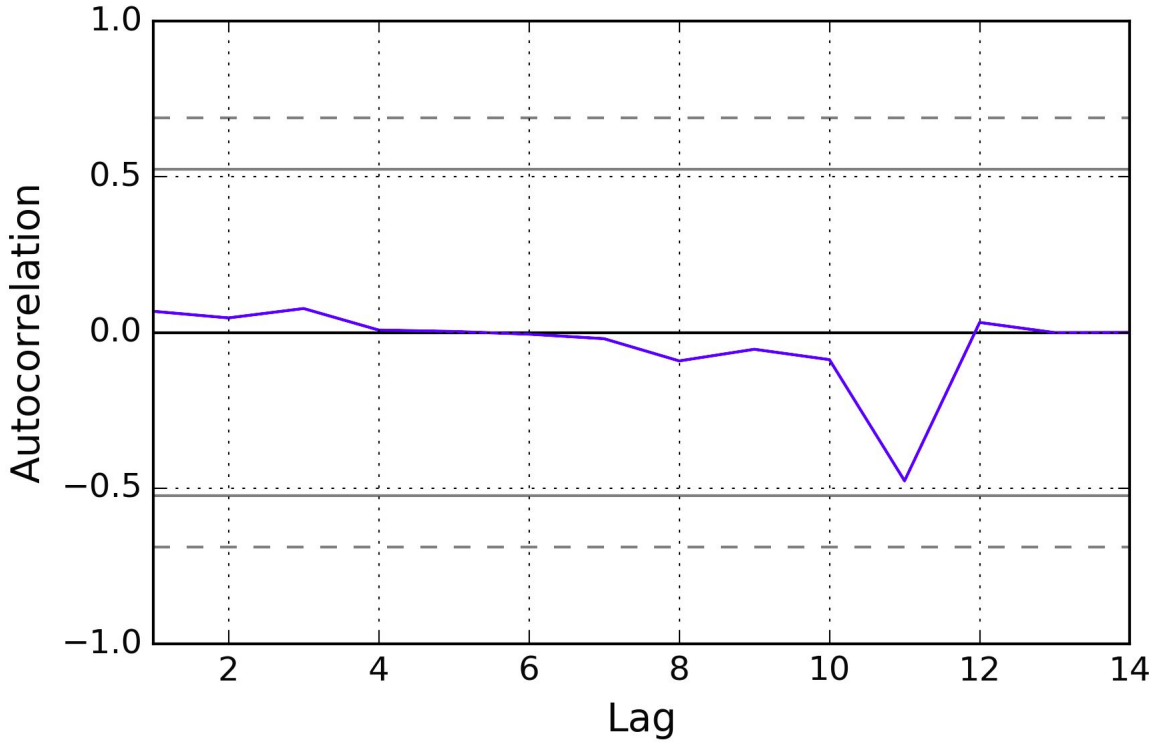
Data Analysis – Data Info

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 817741 entries, 2000-11-01 to 2001-02-28
Data columns (total 8 columns):
Customer ID        817741 non-null int64
Age                817741 non-null object
Residence Area     817741 non-null object
Product Subclass   817741 non-null int64
Product ID         817741 non-null int64
Amount             817741 non-null int64
Asset              817741 non-null int64
Sale Price         817741 non-null int64
dtypes: int64(6), object(2)
memory usage: 56.1+ MB
```

Data Analysis – Data Plot of a Product

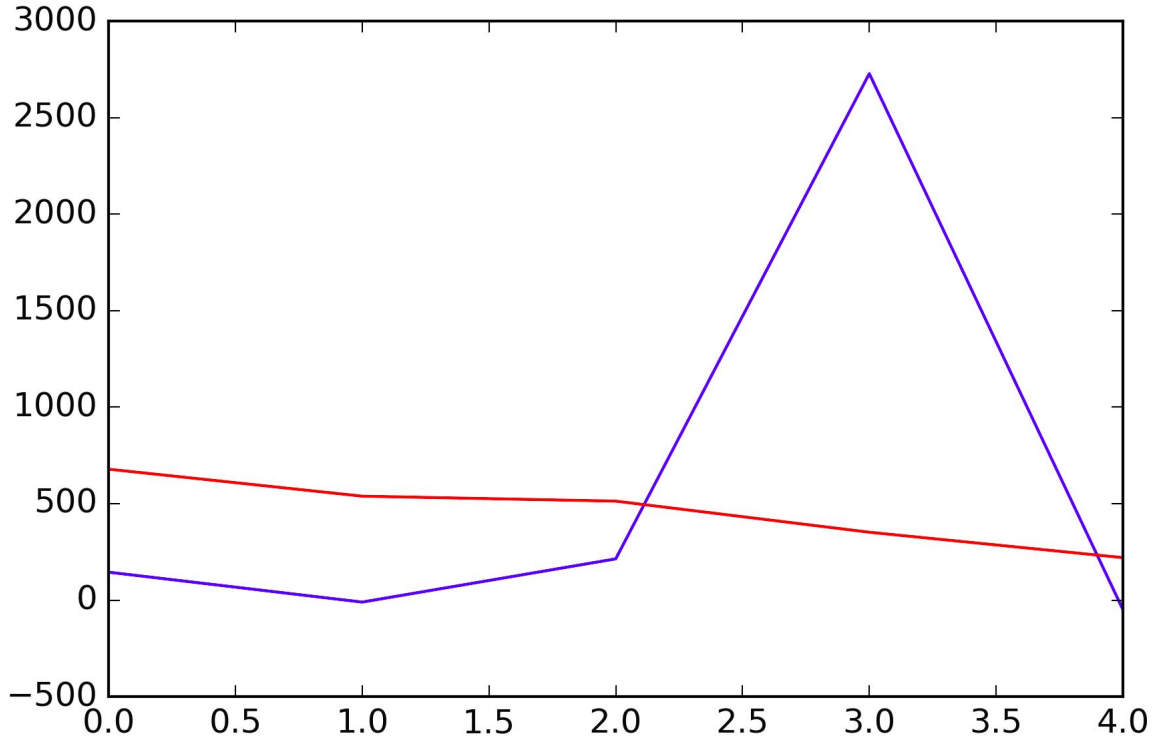


Data Analysis – Autocorrelation Plot of the Product



Model Validation – Test vs Prediction Plot

Best ARIMA hyperparameters (2, 0, 0) MSE=18056.005



THANKS!

gurpreet.sachdeva@gmail.com
https://www.linkedin.com/in/gurpreetsachdeva