CFO Forecasting - Mobile Market - Sandesh Brand 1 - Discovery challenge

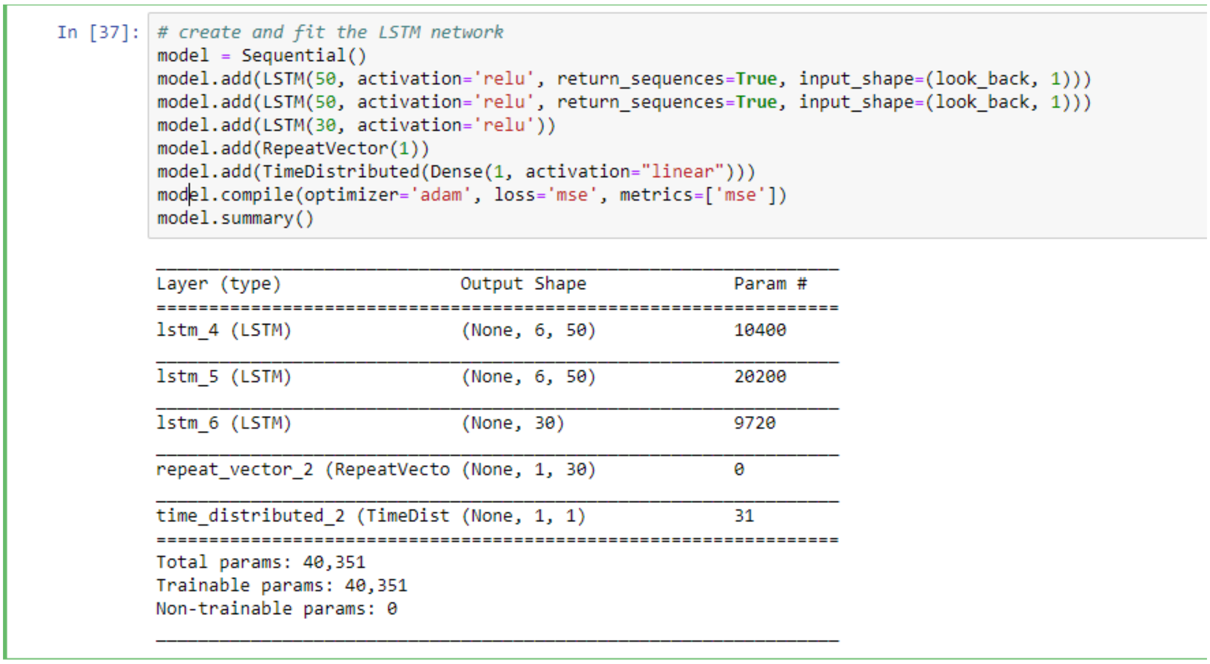
Challenge Objective

The objective of this challenge is to generate time-series forecasts with the highest accuracy predictions possible for the financial variables.

Revenue - the revenue generated by the subscriber base per month for the ‘airtime and data’ service for all products within Sandesh Brand 1. Revenue is therefore the aggregate of all three products - Leopard, Panther and Lion and should be predicted as a single variable.

Training File - Time Series- Revenue.ipynb

- Model –
  Stacked LSTM with time distributed layer
- Data Analysis –
  Data was normalized in the feature range of [0-1] to make it follow normal distribution.
- Model Details–



In [37]:

```
# create and fit the LSTM network
model = Sequential()
model.add(LSTM(50, activation='relu', return_sequences=True, input_shape=(look_back, 1)))
model.add(LSTM(50, activation='relu', return_sequences=True, input_shape=(look_back, 1)))
model.add(LSTM(30, activation='relu'))
model.add(RepeatVector(1))
model.add(TimeDistributed(Dense(1, activation="linear")))
model.compile(optimizer='adam', loss='mse', metrics=['mse'])
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, 6, 50)             10400
_________________________________________________________________
lstm_5 (LSTM)                (None, 6, 50)             20200
_________________________________________________________________
lstm_6 (LSTM)                (None, 30)                9720
_________________________________________________________________
repeat_vector_2 (RepeatVecto (None, 1, 30)             0
_________________________________________________________________
time_distributed_2 (TimeDist (None, 1, 1)              31
=================================================================
Total params: 40,351
Trainable params: 40,351
Non-trainable params: 0
_________________________________________________________________
```

A look back period of 6 was used to train the model but could be re-trained by changing the lookback parameter in the model training code.

- Cross Validation results –

The last 6 values were used as cross-validation to calculate RMSE and MAPE score based on the challenge description.
MAPE score on revenue through cross-validation was noted as – 94.

- Variable Importance –
  Univariate Model so model dependency is only on the variable being considered.

- Additional Inputs –

  no_of_pred parameter in the training file is the forecast_horizon which should be equal to the look back period and any look back other than 6 would need re-training of the model based on the new look back period on the training data.

  The final model in the models directory was trained on the whole dataset of 42 datapoints and the trained model with the scaler was saved in the models directory under the name –

  scaler-Total Revenue-Panther - Leopard - Lion.sav

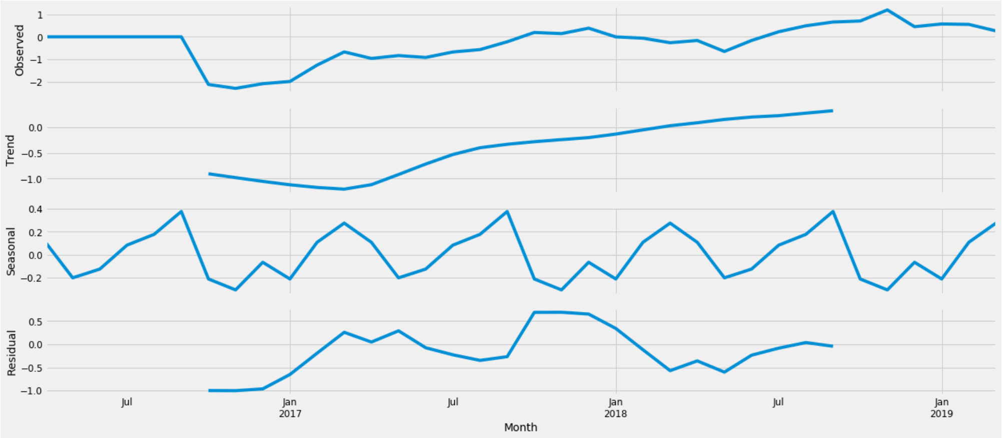
  Total Revenue-Panther - Leopard - Lion.h5

Leavers - the number of subscribers per individual product who terminated service with the brand during that month.

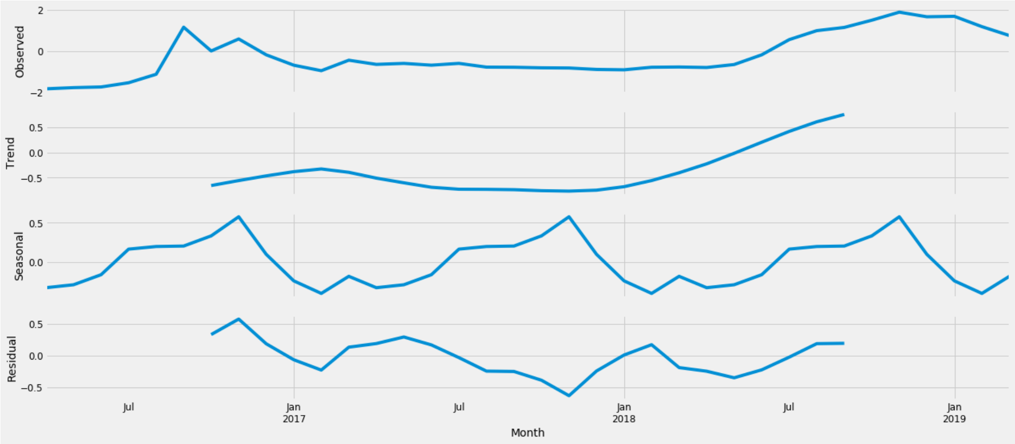
Training File - Time Series- Leavers.ipynb

- Model –
  Sarimax Model with lowest parameter configuration
- Data Analysis –
  The trend and seasonality were captured in the following graphs for each product –

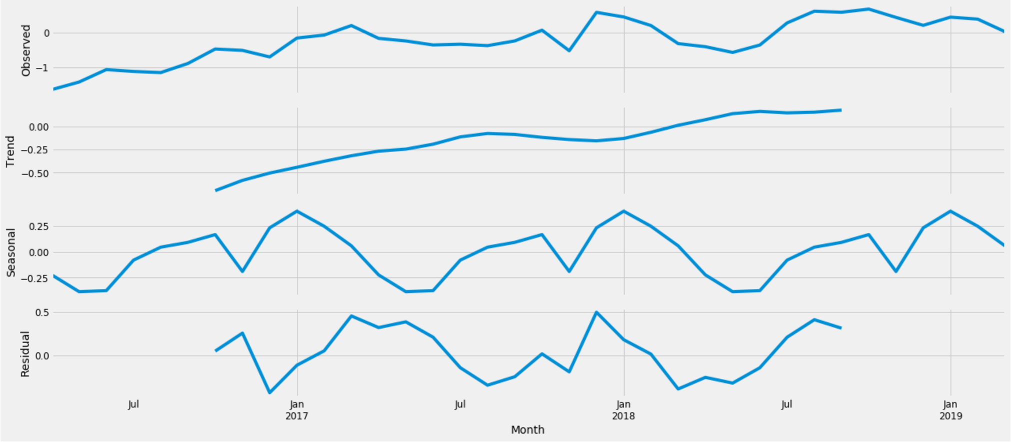
LION-



LEOPARD –



PANTHER –



- Model Details –
  Various models were trained to select the optimal parameter values for our ARIMA(p,d,q)(P,D,Q)s time series model. A “grid search” to iteratively explore different combinations of parameters. For each combination of parameters, we fit a new seasonal ARIMA model with the SARIMAX() function from the statsmodels module and assess its overall quality.

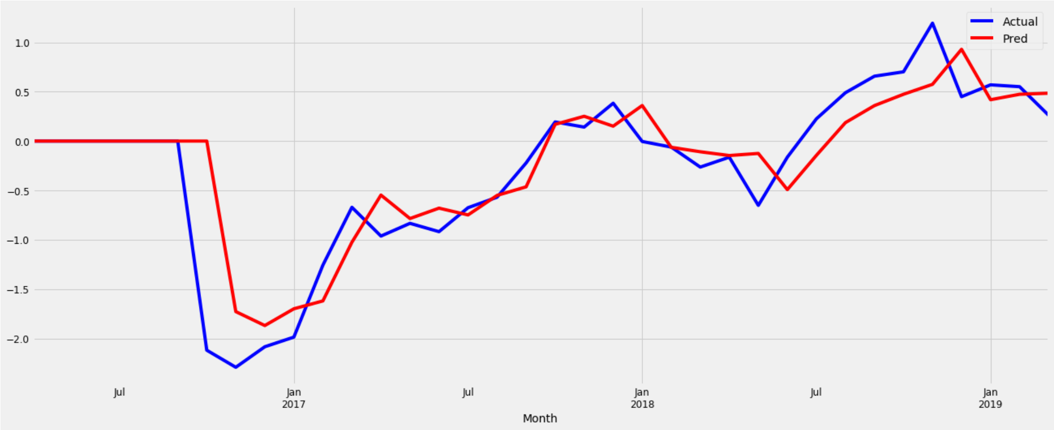
  The various parameters for each product were –
  Format - (p,d,q)(P,D,Q)s
  Lion- (1, 0, 0), (1, 0, 0, 12)
  Panther- (0, 1, 1), (1, 1, 0, 12)
  Leopard- (1, 1, 0), (1, 0, 0, 12)

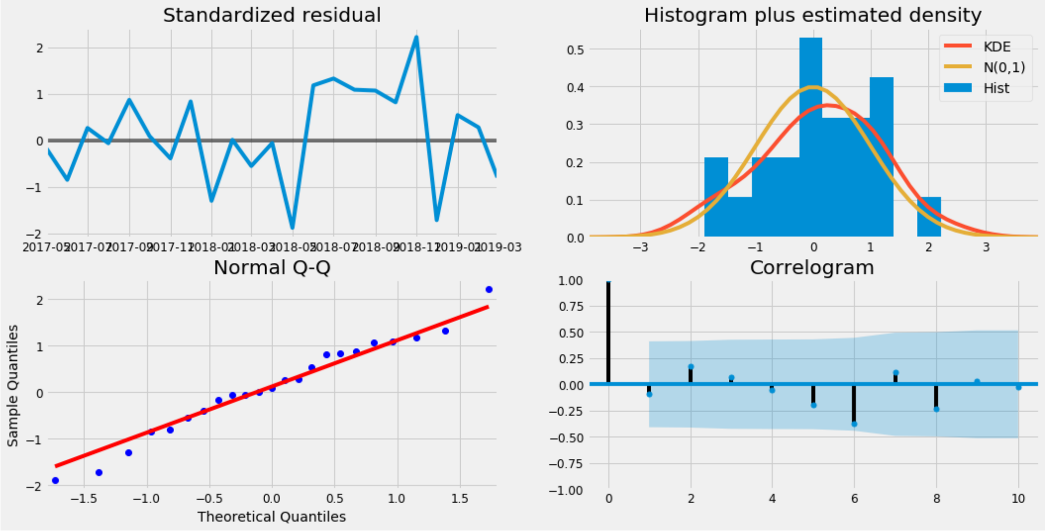
- Cross Validation results –
  Last 6 values were used for cross-validation and the results for each product is shared below. RMSE was also considered as a matrix while evaluating the models.

  MAPE SCORE –
  Lion – 11.709413551522282
  Panther- 0. 600363288366943
  Leopard - 42.57574174452768

- Variable Importance – Univariate Model so model dependency is only on the variable being considered.
- Additional Inputs –
  The primary concern is to ensure that the residuals of our trained model are uncorrelated and normally distributed with zero-mean. Which was considered from the graphs given below –

  Lion-





- In the top right plot, we see that the red KDE line follows closely with the N(0,1) line (where N(0,1)) is the standard notation for a normal distribution with mean 0 and standard deviation of 1). This is a good indication that the residuals are normally distributed.
- The qq-plot on the bottom left shows that the ordered distribution of residuals (blue dots) follows the linear trend of the samples taken from a standard normal distribution with N(0, 1). Again, this is a strong indication that the residuals are normally distributed.
- The residuals over time (top left plot) don’t display any obvious seasonality and appear to be white noise. This is confirmed by the autocorrelation (i.e. correlogram) plot on

the bottom right, which shows that the time series residuals have low correlation with lagged versions of itself.

Such graphs were considered for other products also which can be seen by running the training pipelines.

Additionally, any forecast horizon can be given for prediction.

Lastly, the model were trained for each product with all their datapoints and saved in models directory with names –

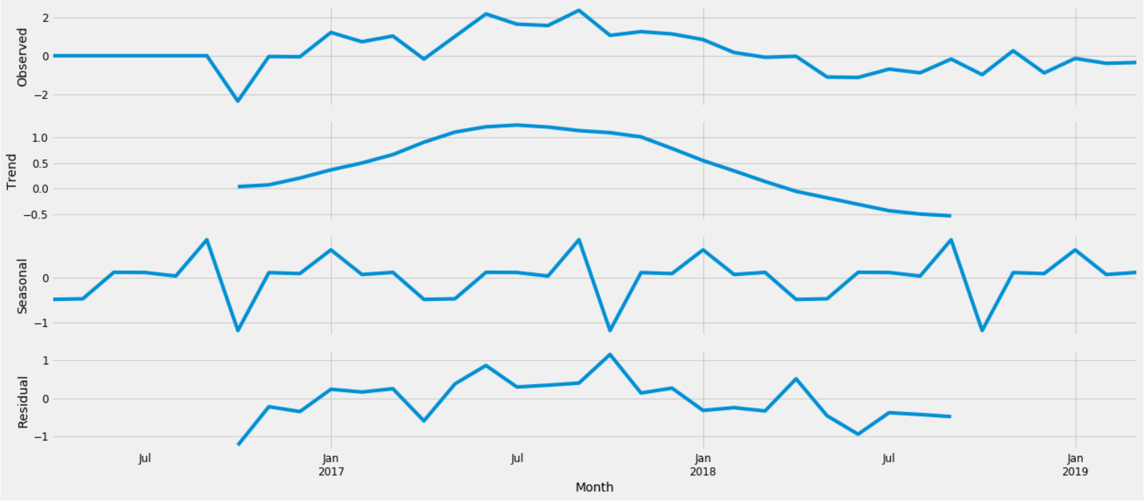
Leavers(Norm)-Leopard.h5
Leavers(Norm)-Lion.h5
Leavers(Norm)-Panther.h5

Gross adds - the number of new subscribers to each individual product joining the brand during a month.

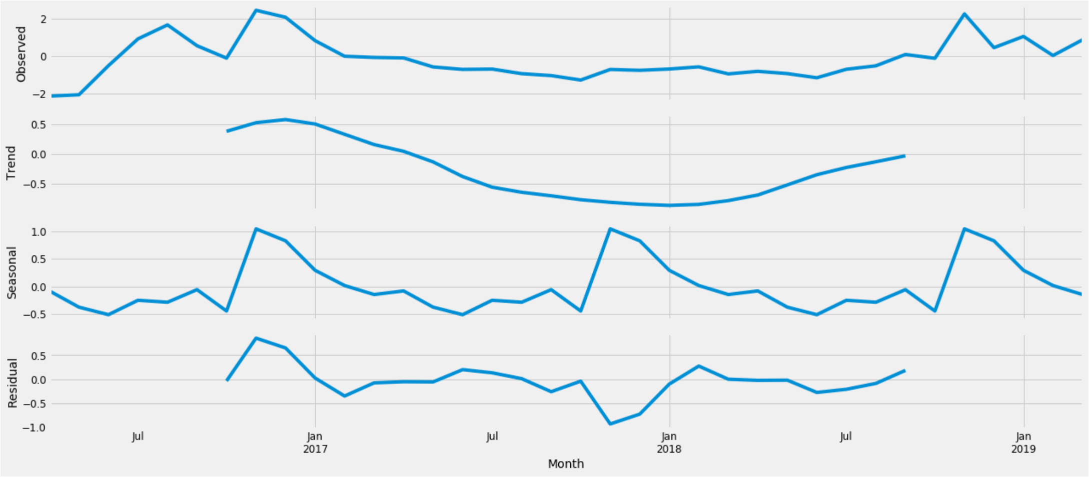
Training File - Time Series-Gross Adds.ipynb

- Model –
  Sarimax Model with lowest parameter configuration
- Data Analysis –
  The trend and seasonality were captured in the following graphs for each product –

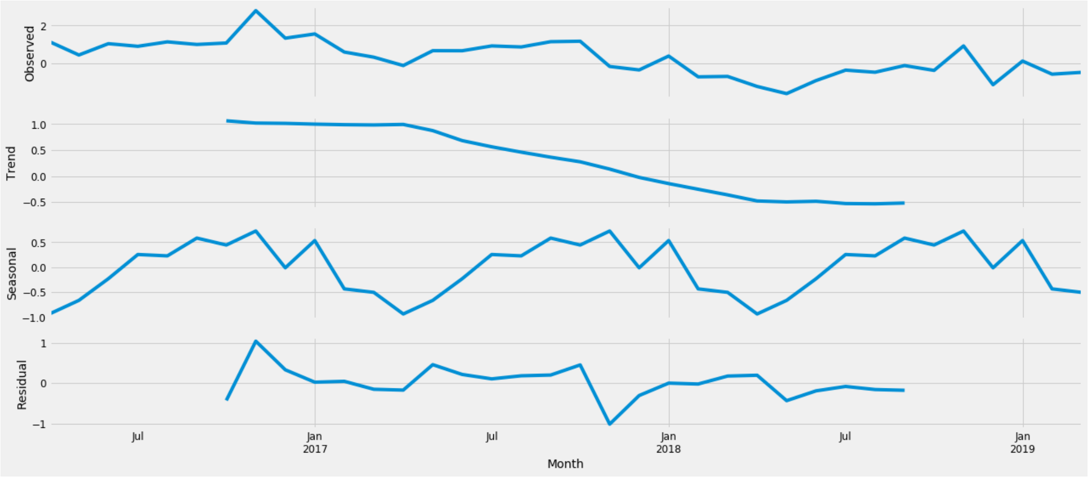
LION-



LEOPARD –



PANTHER –



- Model Details –
  Various models were trained to select the optimal parameter values for our ARIMA(p,d,q)(P,D,Q)s time series model. A "grid search" to iteratively explore different combinations of parameters. For each combination of parameters, we fit a new seasonal ARIMA model with the SARIMAX() function from the statsmodels module and assess its overall quality.

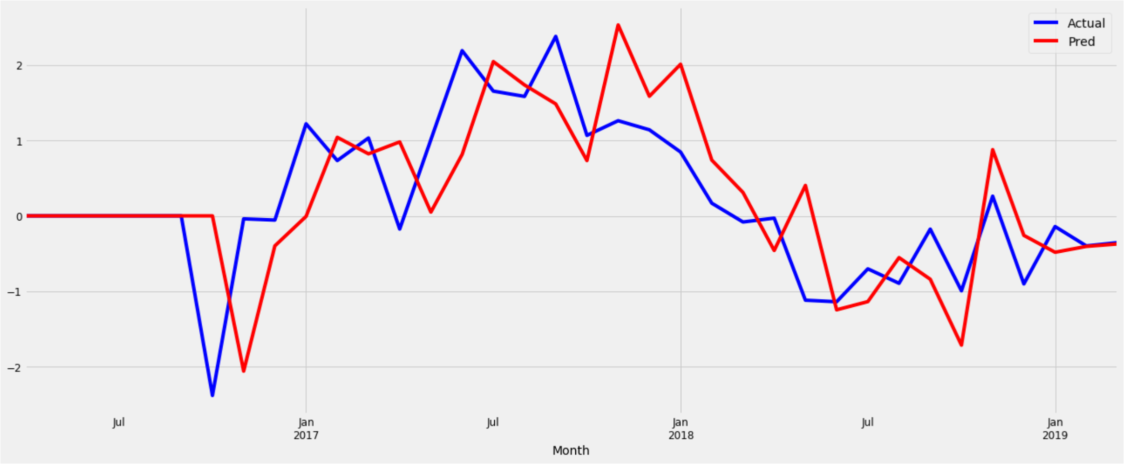
  The various parameters for each product were –
  Format – (p,d,q)(P,D,Q)s
  Lion- (1, 1, 1), (1, 1, 0, 12)
  Panther- (1, 0, 1), (1, 1, 0, 12)
  Leopard- (1, 1, 1), (1, 1, 0, 12)

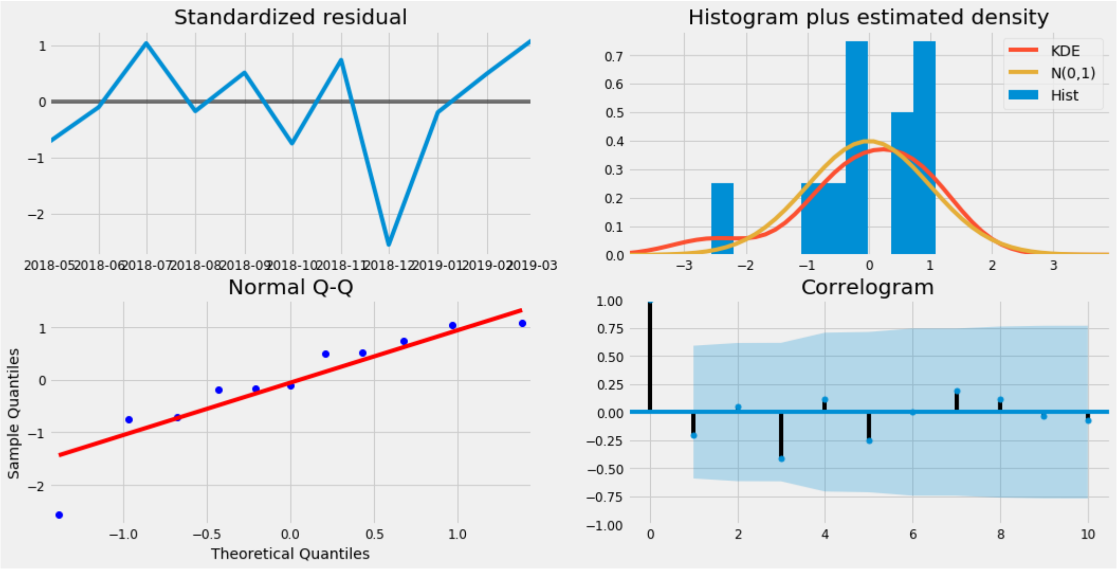
- Cross Validation results –
  Last 6 values were used for cross-validation and the results for each product is shared below. RMSE was also considered as a matrix while evaluating the models.

MAPE SCORE -
Lion - 53.1306169500626
Panther- 50.5180511836907
Leopard - 18.350625982956963

- Variable Importance - Univariate Model so model dependency is only on the variable being considered.
- Additional Inputs -
The primary concern is to ensure that the residuals of our trained model are uncorrelated and normally distributed with zero-mean. Which was considered from the graphs given below -

Panther-





- In the top right plot, we see that the red KDE line follows closely with the N(0,1) line (where N(0,1)) is the standard notation for a normal distribution with mean 0 and standard deviation of 1). This is a good indication that the residuals are normally distributed.

- The qq-plot on the bottom left shows that the ordered distribution of residuals (blue dots) follows the linear trend of the samples taken from a standard normal distribution with N(0, 1). Again, this is a strong indication that the residuals are normally distributed.
- The residuals over time (top left plot) don’t display any obvious seasonality and appear to be white noise. This is confirmed by the autocorrelation (i.e. correlogram) plot on the bottom right, which shows that the time series residuals have low correlation with lagged versions of itself.

Such graphs were considered for other products also which can be seen by running the training pipelines.

Additionally, any forecast horizon can be given for prediction.

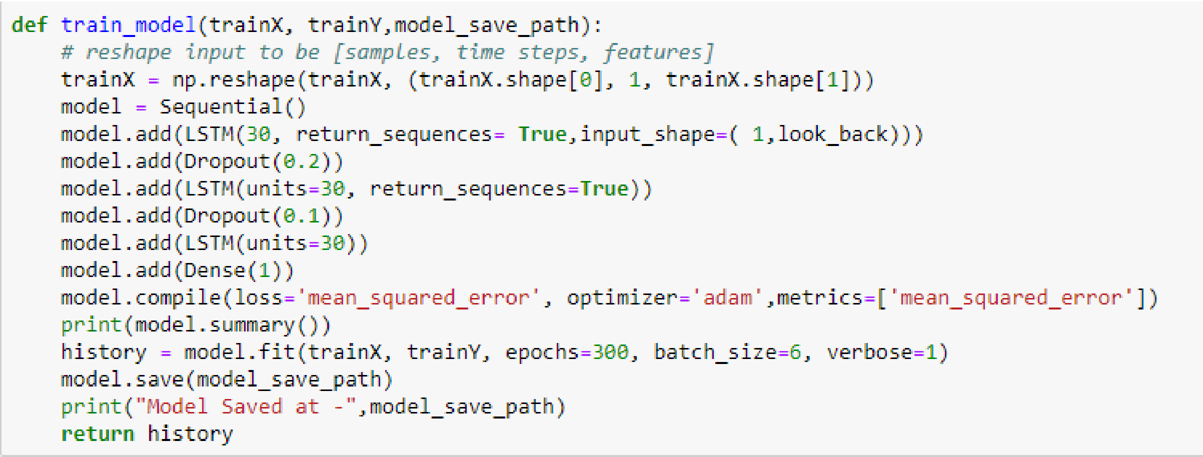
Lastly, the model were trained for each product with all their datapoints and saved in models directory with names –

Gross Adds(Norm)-Leopard.h5
Gross Adds(Norm)-Lion.h5
Gross Adds(Norm)-Panther.h5

Net Migrations - the net number of subscribers moving onto the individual product per month from an alternative product within Sandesh Brand 1. Net Migrations for all three products should add up to Zero (or very close to Zero).

Training File - Time Series- Net Migrations.ipynb

- Model –
  3 layers stacked LSTM with 2 drop out layer.
- Data Analysis –
  Data was normalized in the feature range of [0-1] to make it follow normal distribution.
- Model Details –
  The model was derived from the foundation model architecture given in the challenge description which was tuned on the cross-validation results.
  The architecture of the model is –



```
def train_model(trainX, trainY,model_save_path):
    # reshape input to be [samples, time steps, features]
    trainX = np.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
    model = Sequential()
    model.add(LSTM(30, return_sequences= True,input_shape=( 1,look_back)))
    model.add(Dropout(0.2))
    model.add(LSTM(units=30, return_sequences=True))
    model.add(Dropout(0.1))
    model.add(LSTM(units=30))
    model.add(Dense(1))
    model.compile(loss='mean_squared_error', optimizer='adam',metrics=['mean_squared_error'])
    print(model.summary())
    history = model.fit(trainX, trainY, epochs=300, batch_size=6, verbose=1)
    model.save(model_save_path)
    print("Model Saved at -",model_save_path)
    return history
```

Mean Squared error was used for metric evaluation with adam optimizer. The model was given a batch size of 6 and was trained for 300 epochs.

- Cross Validation results –
  The forecast horizon taken was 6 with a look back period of 1 for this model. The last 6 values were taken as cross-validation set.

  MAPE SCORE for each product model recorded were –

  -Net Migration(lion)- 7.476546141458027

  -Net Migration(panther)- 13.689690333959822

  -Net Migration(leopard) - 13.427593439825712

- Variable Importance –
  Univariate Model so model dependency is only on the variable being considered.
- Additional Inputs –
  no_of_pred parameter in the training file is the forecast_horizon which should be equal to the look back period which is 6 currently.
  The new forecat horizon can be any value.

  The final model in the models directory was trained on the whole dataset of 42 data points for each product for this variable and the trained model with the scaler were saved in the models directory under the name –

  Net Migrations(Norm)-Leopard.h5
  Net Migrations(Norm)-Lion.h5
  Net Migrations(Norm)-Panther.h5
  scaler-Net Migrations(Norm)-Leopard.sav
  scaler-Net Migrations(Norm)-Lion.sav
  scaler-Net Migrations(Norm)-Panther.sav

Closing Base - the number of subscribers to an individual product at the end of each month. This change in Closing Base from one month to the next is therefore the sum of Gross Adds and Net Migrations, minus Leavers.

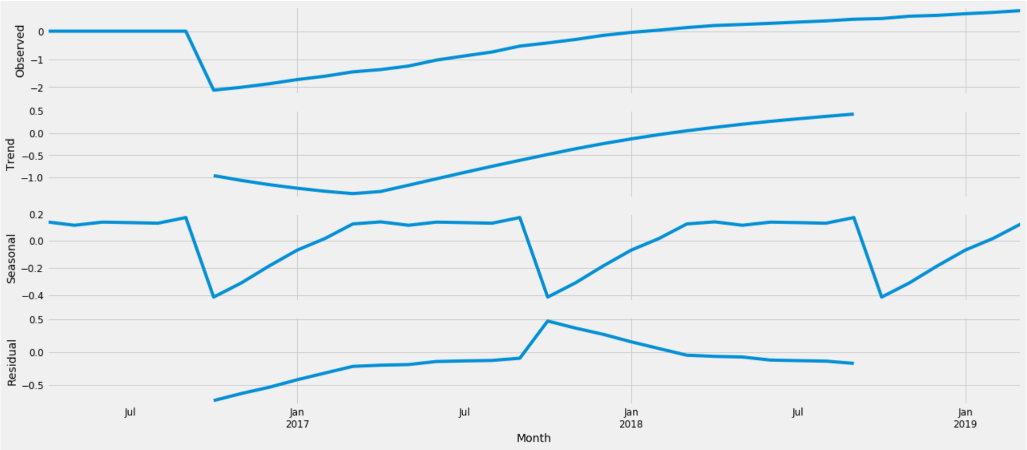
Training File - Time Series-Closing Base.ipynb

- Model –

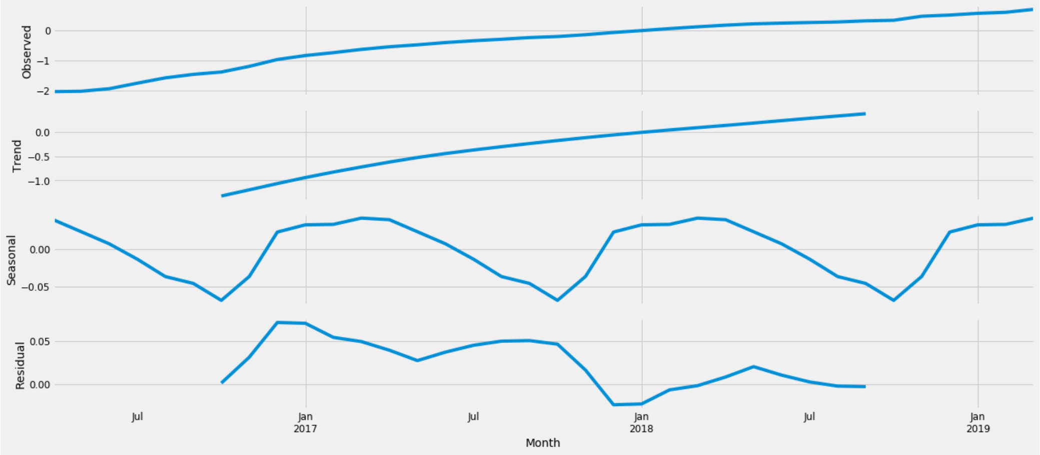
  Sarimax Model with lowest parameter configuration

- Data Analysis –
  The trend and seasonality were captured in the following graphs for each product –

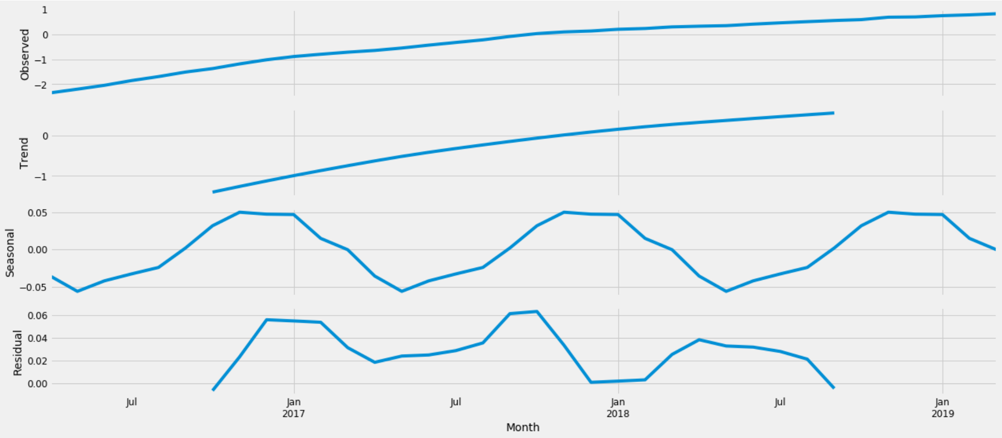
LION-



LEOPARD –



PANTHER –



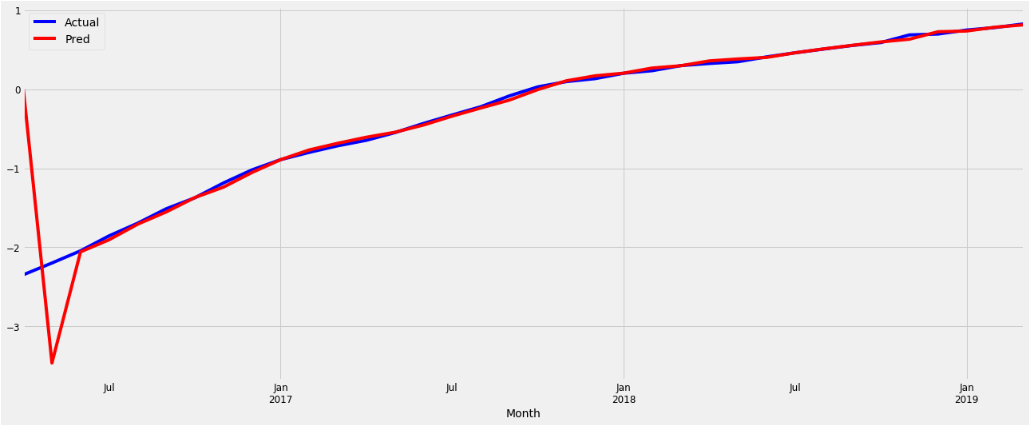
- Model Details –
  Various models were trained to select the optimal parameter values for our ARIMA(p,d,q)(P,D,Q)s time series model. A "grid search" to iteratively explore different combinations of parameters. For each combination of parameters, we fit a new seasonal ARIMA model with the SARIMAX() function from the statsmodels module and assess its overall quality.

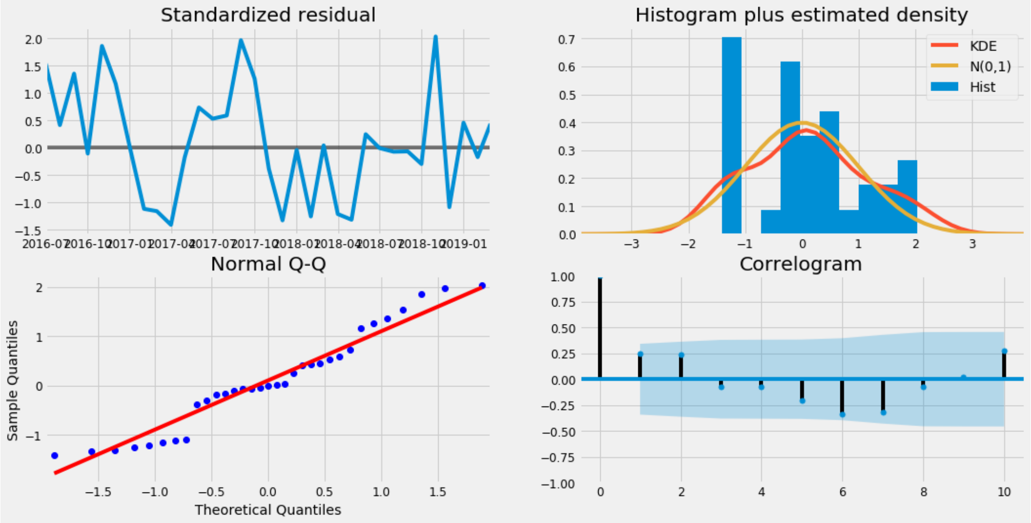
  The various parameters for each product were –
  Format – (p,d,q)(P,D,Q)s
  Lion- (1, 1, 1), (1, 0, 0, 12)
  Panther- (1, 1, 1), (0, 0, 0, 12)Leopard- (1, 1, 0), (0, 0, 0, 12)
- Cross Validation results –
  Last 6 values were used for cross-validation and the results for each product is shared below. RMSE was also considered as a matrix while evaluating the models.

  MAPE SCORE –
  Lion – 95.87542514651317
  Panther- 96.01003483162944
  Leopard - 96.20292265274038

- Variable Importance –
  Univariate Model so model dependency is only on the variable being considered.
- Additional Inputs –
  The primary concern is to ensure that the residuals of our trained model are uncorrelated and normally distributed with zero-mean. Which was considered from the graphs given below –

Panther-





- In the top right plot, we see that the red KDE line follows closely with the N(0,1) line (where N(0,1)) is the standard notation for a normal distribution with mean 0 and standard deviation of 1). This is a good indication that the residuals are normally distributed.
- The qq-plot on the bottom left shows that the ordered distribution of residuals (blue dots) follows the linear trend of the samples taken from a standard normal distribution with N(0, 1). Again, this is a strong indication that the residuals are normally distributed.
- The residuals over time (top left plot) don’t display any obvious seasonality and appear to be white noise. This is confirmed by the autocorrelation (i.e. correlogram) plot on the bottom right, which shows that the time series residuals have low correlation with lagged versions of itself.

Such graphs were considered for other products also which can be seen by running the training pipelines.
Additionally, any forecast horizon can be given for prediction.

Lastly, the model were trained for each product with all their data points and saved in models directory with names –

Closing Base-Leopard.h5
Closing Base-Lion.h5
Closing Base-Panther.h5

MAKING PREDICTIONS –

File - MakePredictions.ipynb

Use this file to load a dataset and provide the variable specific dataset to the various models that can be easily loaded from here to make prediction.

Using this a forecast excel spreadsheet was created containing the time period and the generic variable key names with the forecast values from each model variable and its respective products.