Suresh Gourigolla

DEMAND FORCASTING

**INTRODUCTION**

Demand Forecast is the art and science of forecasting customer demand to drive holistic execution of such demand by corporate supply chain and business management and is...

* Built on top of Apache Spark in and written in Scala inside Databricks Platform
* Uses a machine learning Random Forest implementation to find the most important attribute to a demand forecast of interest.

Includes a dataset with a subset of simulated for store transaction

**Step1: Ingest Demand Forecast Data**

* The demand forecast data consist of 4 generated datasets. They are menu, sales, transaction and location.
* Runs the "data Setup" Databricks Notebook to populate the temp table to be used in this notebook. This notebook is kept under the same folder as this notebook.

Just read the parquet files from the storage containers and write into as TEMP SQL table.

And Sample data of Menu,

|  |
| --- |
| Name |
| Sausage/Bacon, Egg and Cheese Biscuit |
| Sausage/Bacon, Egg and Cheese Muffin |
| Grilled Chicken Cool Wrap |
| Grilled Nuggets |
| Grilled Chicken Sandwich |
| Grilled Market Salad |
| Cobb Salad |
| Spicy Southwest Salad |
| Chicken Salad Sandwich |

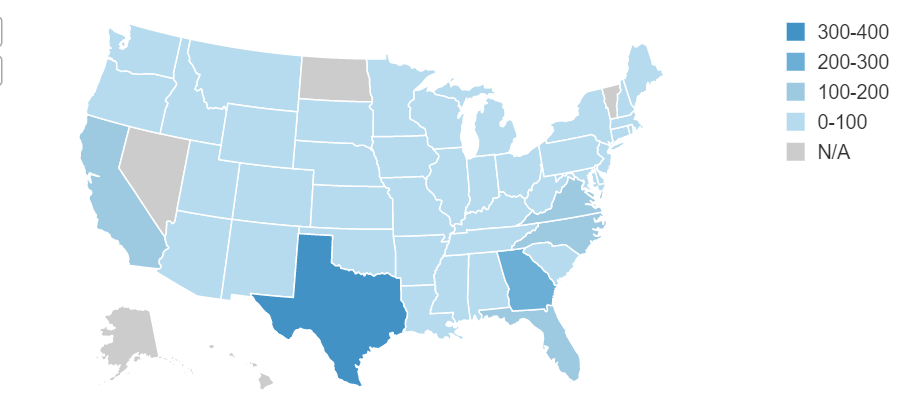
### Step2: Enrich the data by joining the datasets to get complete view of the sales transaction

* Joined the dataset for transaction, location and menu to give a snapshot view of the transaction at any given time

And hold the data into persist layer.

### Step3: Explore demand forecast Data by capturing transaction into a Table and Analyse Data

### select state, count (\*) from locations group by state order by count (\*)



**Demand by Store**

Based on the region population figuring the sales of good and beverages.

**Visualization:**

* Visualizing and find outliers

**Demand Distribution - All Products:**

select a.locationid,a.SKU,b.name,a.sale

from (

select locationid,SKU,count(\*) as sale

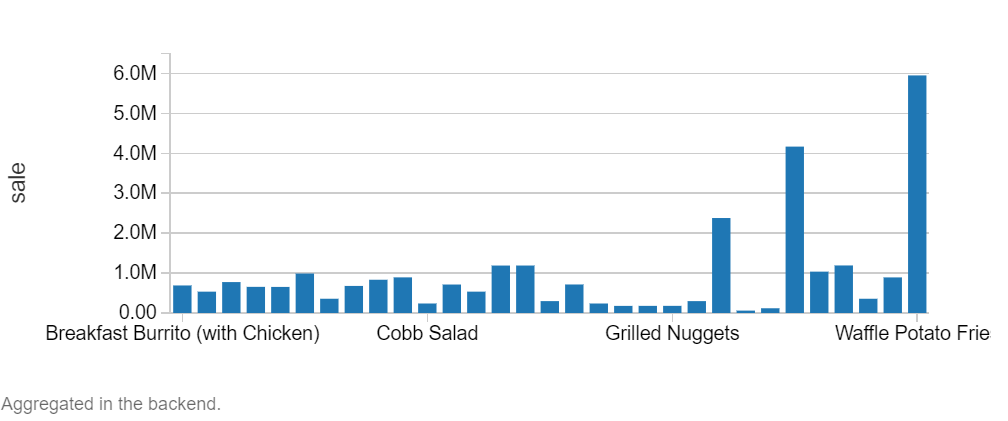
from sales

group by locationid,SKU

order by locationid,SKU asc) a

,menu b

where a.SKU = b.SKU



**Demand Distribution – Entrees:**

select a.SKU,b.name,a.sale

from (

select SKU,count(\*) as sale

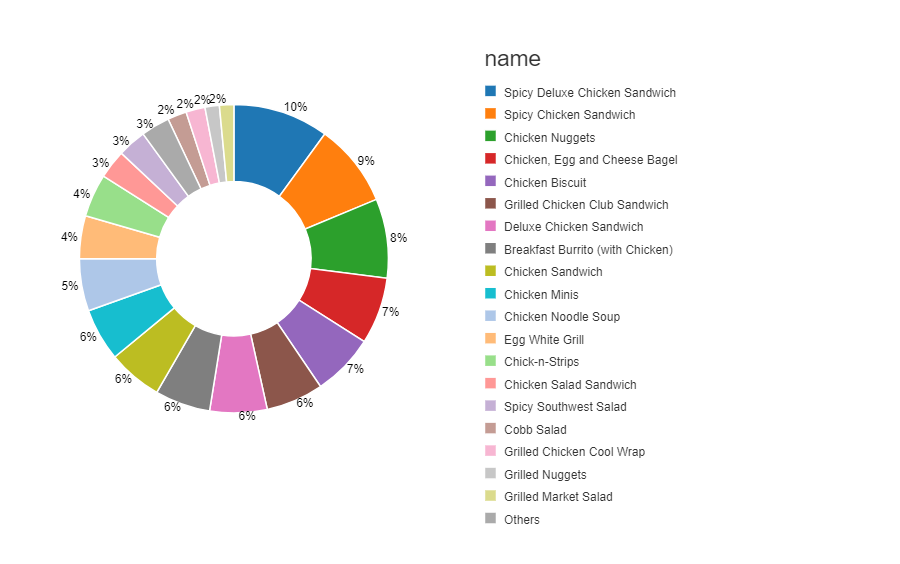
from sales

where sku >= 90000 and sku <= 90021

group by SKU order by SKU asc) a

,menu b where a.SKU = b.SKU

order by a.sale desc



**Demand Distribution – Drinks**

select a.SKU,b.name,a.sale

from (

select SKU,count(\*) as sale

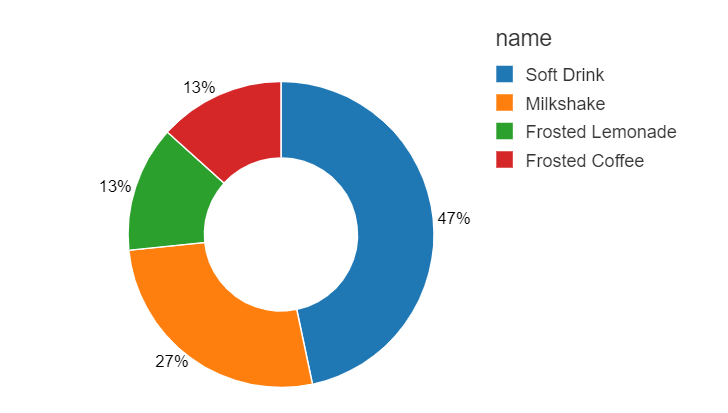
from sales

where sku between 90022 and 90025

group by SKU order by SKU asc) a

,menu b where a.SKU = b.SKU

order by a.sale desc



**Demand Distribution – Sides**

select a.SKU,b.name,a.sale

from (

select SKU,count(\*) as sale

from sales

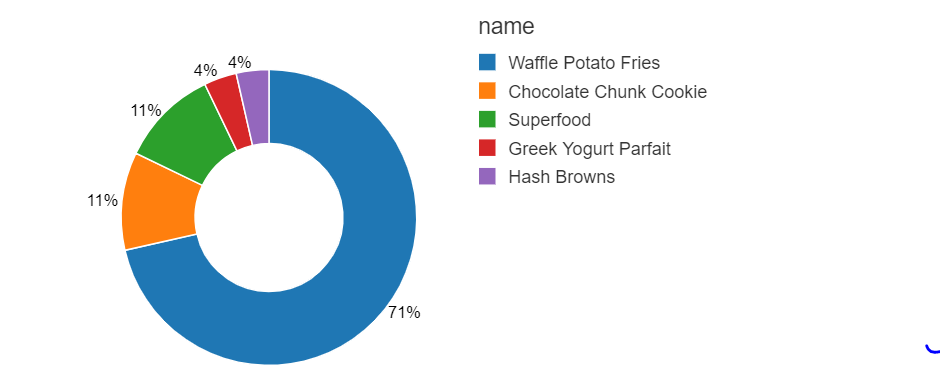
where sku between 90026 and 90030

group by SKU order by SKU asc) a

,menu b where a.SKU = b.SKU

order by a.sale desc

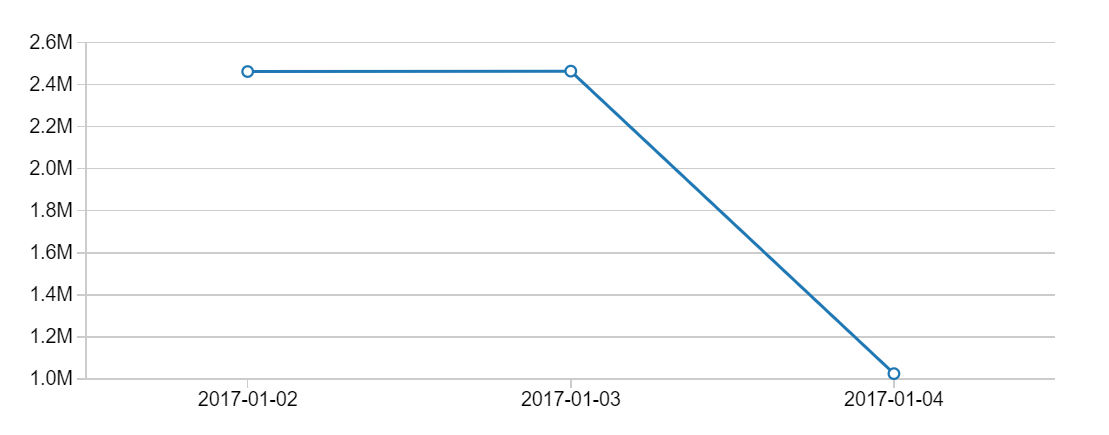
Create a Widget for Customizable Insights



**Create a Widget for Customizable Insights**

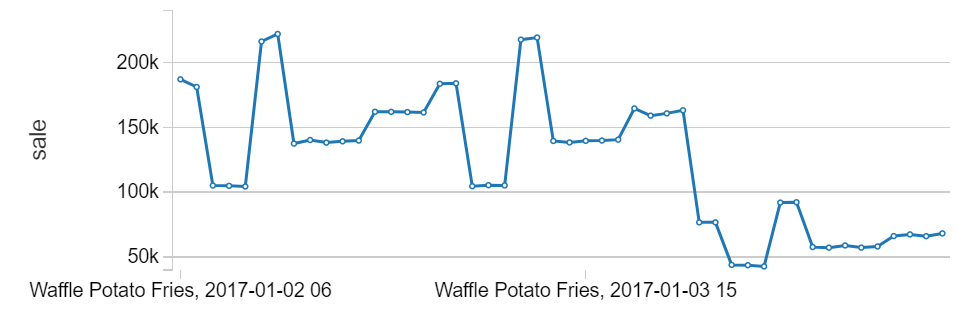
**CREATE WIDGET DROPDOWN Product DEFAULT "Waffle Potato Fries" CHOICES select distinct name from menu**

To DATE, SKU, NAME and Sale information table, just passing as a external parameter value product into the table to extract the results of particular DATE and SKU.

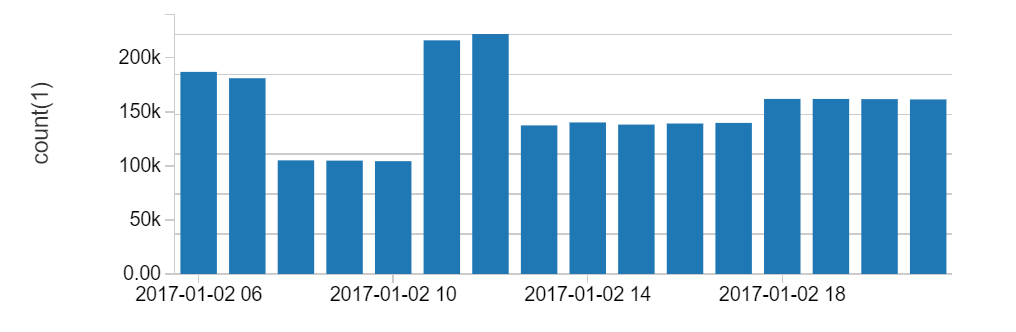


**Weekly Product Demand**

Passing a week list of days as an argument into the function, Thus extract the information on particular product sales on day wise.

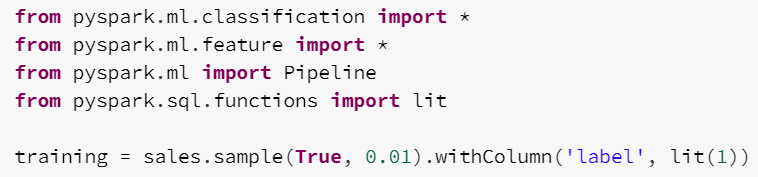


**Hourly Demand for Product**

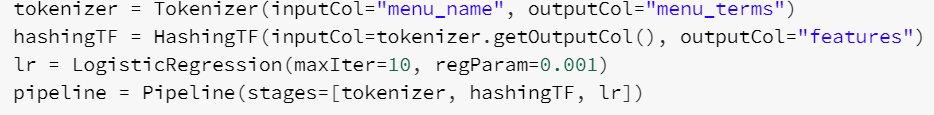


#### **Step 5: Model creation**

Importing the libraries of Pyspark of Classification and Feature.



**Define the pipeline using transformers and a classifier**



Train the model using the historical data

model = pipeline.fit(training)

Make predictions

prediction = model.transform(training)

Persist the trained model to disk for (real-time) scoring

display(prediction.select("label", "menu\_terms", "features", "prediction"))

#### **Make predictions using the model trained using the historical data**

display(streaming\_predictions.select("label", "menu\_terms", "features", "prediction"))

## Results interpretation

The plot above shows that demand index for each ot the items on the menu.



1. For the Entrees we can both in the batch as well as streaming data that the most popular item is Spicy chicken sandwich
2. For the Sides we can both in the batch as well as streaming data that the most popular item is Waffle potato fries. This also happens to be the Most poular item on the menu.
3. For the Drinks we can both in the batch as well as streaming data that the most popular item is Soft drinks. This also happens to be the Second Most poular item on the menu.