Sales Prediction System for supermarkets

Team Matrix - EE0005

The Problem

Profit is higher when turnover is higher

Need to know sales pattern before hand

Order inventor efficiently

Manage Manpower and staff

An accurate forecasting model can greatly increase supermarket revenue and is generally of great importance to the organization as it improves profit as well as provides insights into the way customers can be better served.

The project

- Forecasting Revenue of Store
- For the next month
- Using data of other stores
- Segmented by product category

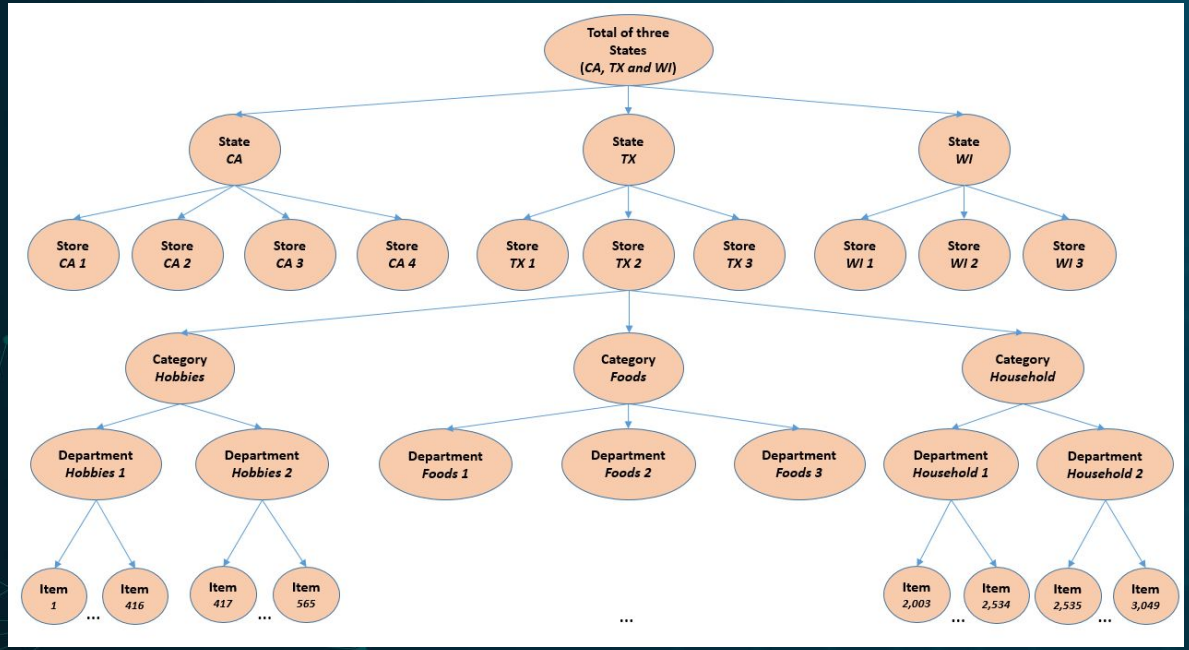
Our Working

Data Processing	**Pandas, numpy**
Seaborn , Plotly	Data Visualization and analysis
Random Forest	**Sci-kit Learn**
Tensorflow, keras	RNN networks

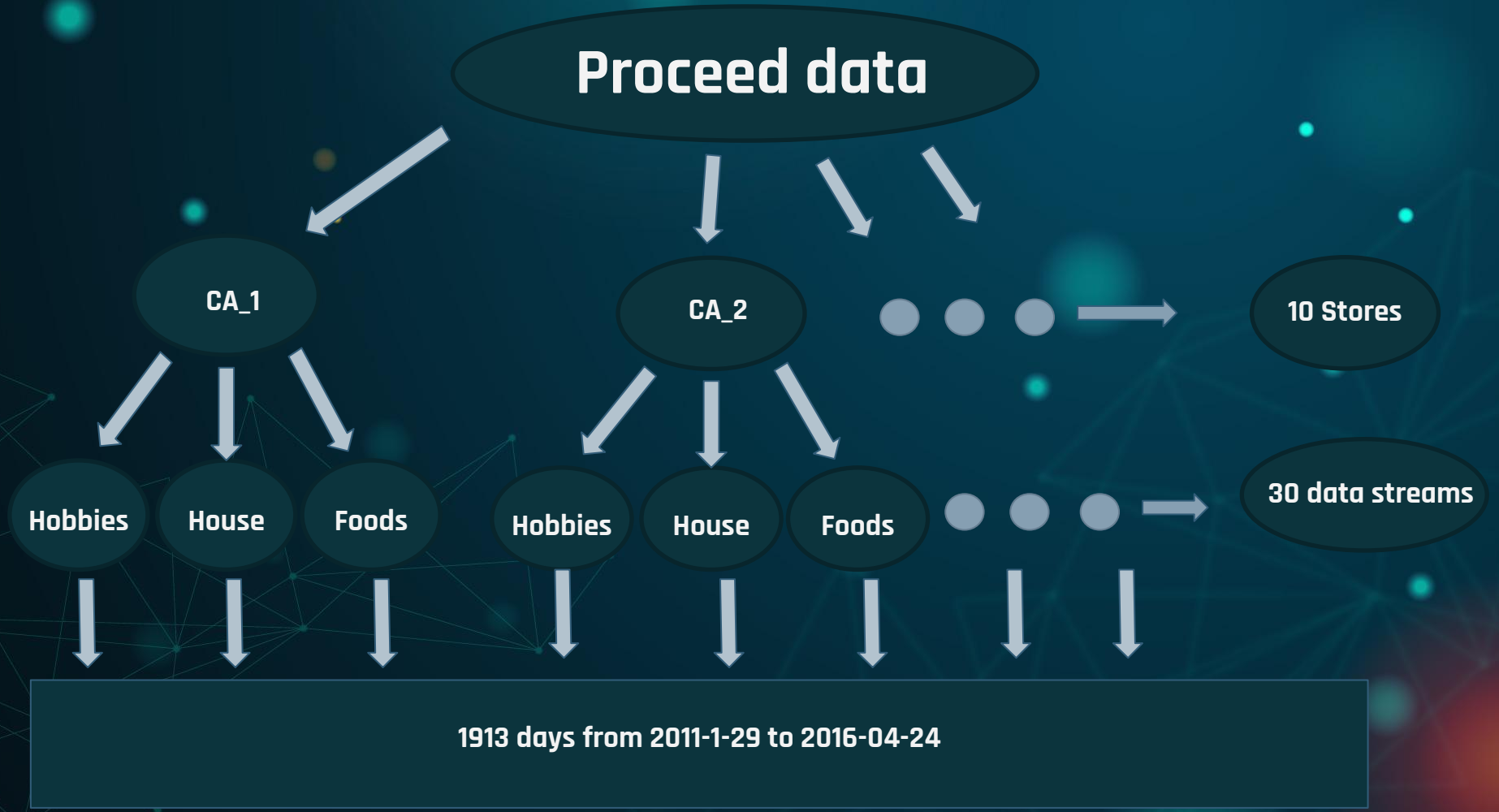
Data

Walmart M5 Accuracy Data

Raw data

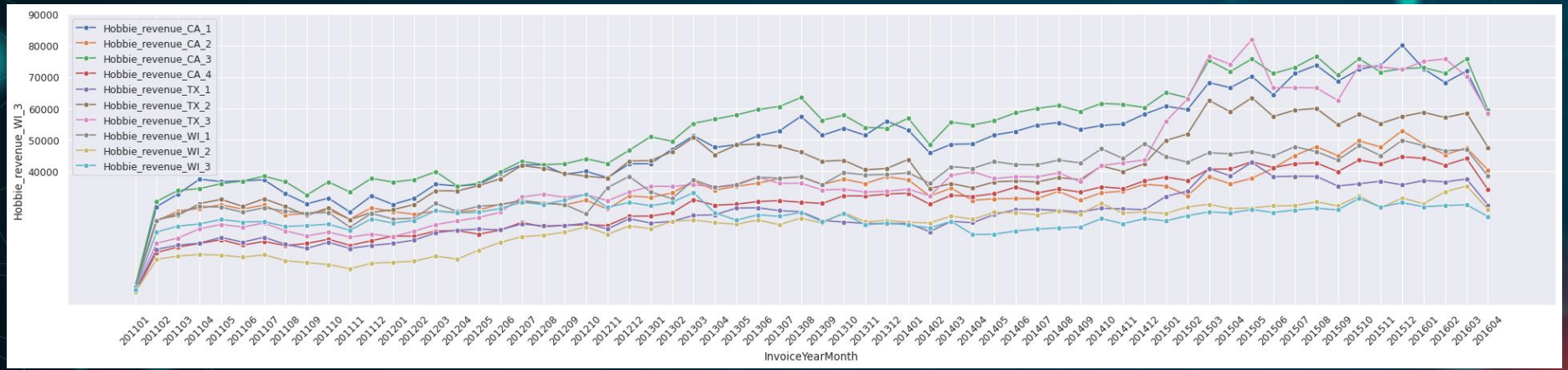


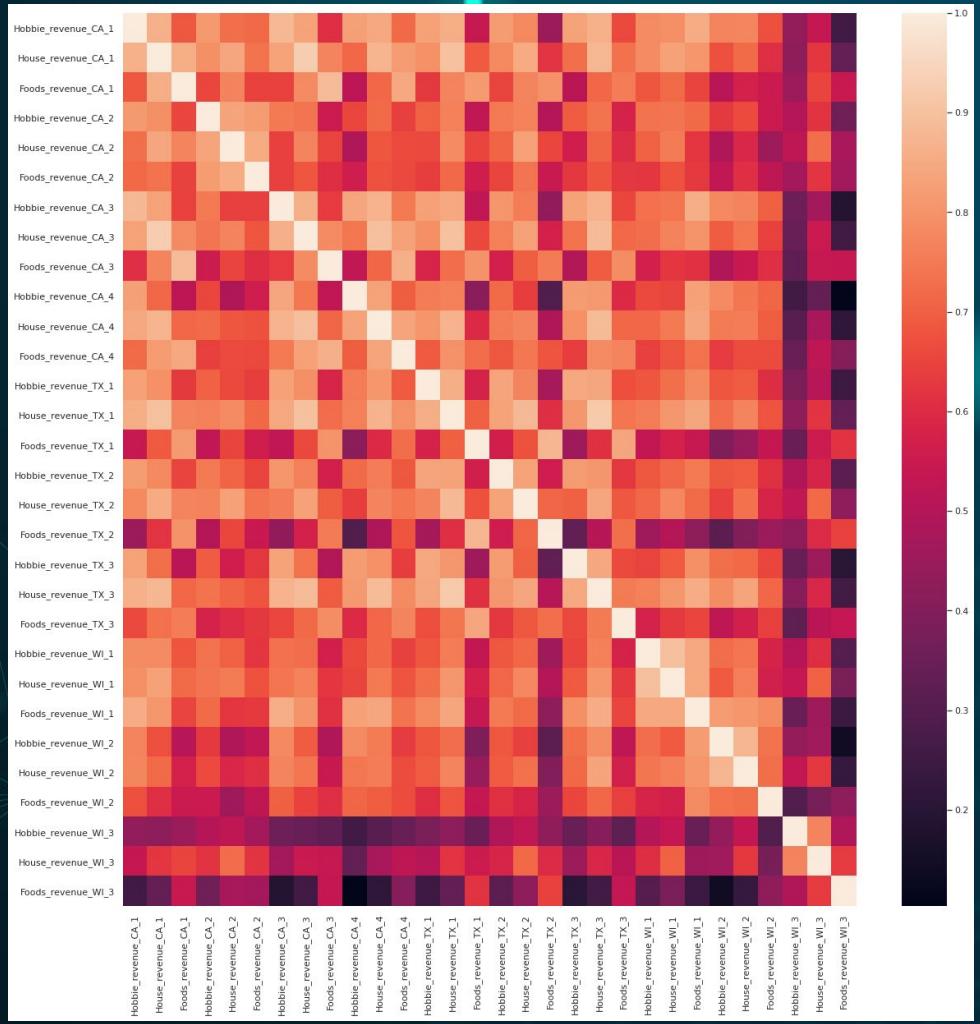
Sales and selling price data inventory-wise


Proceed data
CA_1
CA_2
10 Stores
Hobbies
House
Foods
Hobbies
House
Foods
30 data streams
1913 days from 2011-1-29 to 2016-04-24

	Hobbie_revenue_CA_1	House_revenue_CA_1	Foods_revenue_CA_1	Hobbie_revenue_CA_2	House_revenue_CA_2	Foods_revenue_CA_2	Hobbie_revenue_CA_3	House_revenue_CA_3	Foods_revenue_CA_3	H
count	1913.000000	1913.000000	1913.000000	1913.000000	1913.000000	1913.000000	1913.000000	1913.000000	1913.000000	
mean	1651.024909	3126.688777	6994.523863	1130.609320	3499.558819	4470.465112	1770.127052	5500.156539	9514.429603	
std	568.778895	1034.309162	1782.857772	343.722533	1185.994560	1570.008395	546.017406	1555.736262	2004.472300	
min	0.000000	0.000000	0.000000	0.000000	0.000000	3.160000	0.000000	0.000000	0.000000	
25%	1231.060000	2368.070000	5785.320000	879.940000	2632.260000	3442.770000	1313.200000	4347.270000	8289.990000	
50%	1582.090000	3002.100000	6758.370000	1075.370000	3176.660000	4017.580000	1740.950000	5371.240000	9445.230000	
75%	2028.540000	3667.160000	8077.970000	1310.280000	4191.010000	5226.680000	2158.720000	6438.420000	10807.020000	
max	3985.270000	6603.990000	12279.280000	2510.150000	8595.570000	11747.110000	3485.500000	10845.120000	15586.450000	

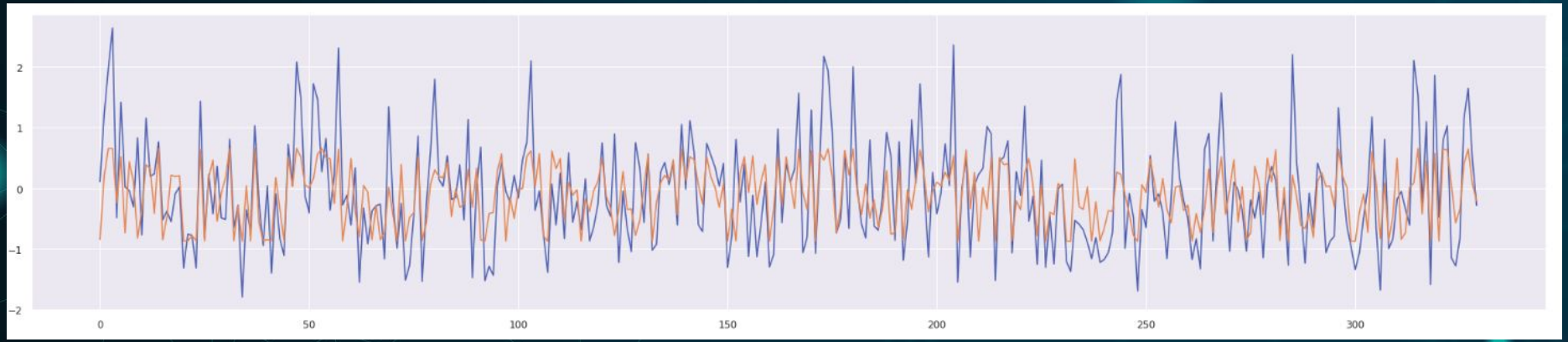
8 rows × 30 columns




Hobbie_revenue_CA_1
House_revenue_CA_1
Foods_revenue_CA_1
Hobbie_revenue_CA_2
House_revenue_CA_2
Foods_revenue_CA_2
Hobbie_revenue_CA_3
House_revenue_CA_3
Foods_revenue_CA_3
Hobbie_revenue_CA_4
House_revenue_CA_4
Foods_revenue_CA_4
Hobbie_revenue_TX_1
House_revenue_TX_1
Foods_revenue_TX_1
Hobbie_revenue_TX_2
House_revenue_TX_2
Foods_revenue_TX_2
Hobbie_revenue_TX_3
House_revenue_TX_3
Foods_revenue_TX_3
Hobbie_revenue_WI_1
House_revenue_WI_1
Foods_revenue_WI_1
Hobbie_revenue_WI_2
House_revenue_WI_2
Foods_revenue_WI_2
Hobbie_revenue_WI_3
House_revenue_WI_3
Foods_revenue_WI_3
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2

- Random Forest is an ensemble machine learning technique capable of performing both regression and classification tasks using multiple decision trees
- Removed outliers before training
- from sklearn.ensemble import RandomForestRegressor
- Test_size = 0.2
- from sklearn.metrics import mean_squared_error

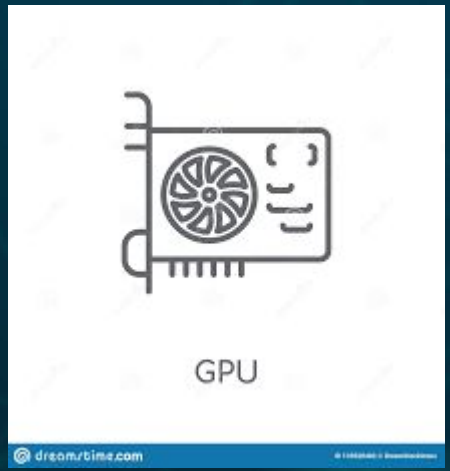
```
Hobbie_revenue_CA_1 0.4270913320673209
House_revenue_CA_1 0.36253245428116315
Foods_revenue_CA_1 0.4611795862249466
Hobbie_revenue_CA_2 0.48152002716754067
House_revenue_CA_2 0.4271971157397212
Foods_revenue_CA_2 0.36787340266724894
Hobbie_revenue_CA_3 0.3992739942173556
House_revenue_CA_3 0.3922743276993896
Foods_revenue_CA_3 0.48007461600469514
Hobbie_revenue_CA_4 0.4041720486977423
House_revenue_CA_4 0.4132930524547144
Foods_revenue_CA_4 0.43638139223266964
Hobbie_revenue_TX_1 0.41046707501653834
House_revenue_TX_1 0.3705262139154396
Foods_revenue_TX_1 0.47302570562846813
Hobbie_revenue_TX_2 0.4675841880450109
House_revenue_TX_2 0.5040668710102293
Foods_revenue_TX_2 0.5186702169249429
Hobbie_revenue_TX_3 0.4773264767380756
House_revenue_TX_3 0.37397094987853396
Foods_revenue_TX_3 0.4930324074857432
Hobbie_revenue_WI_1 0.42813984473785854
House_revenue_WI_1 0.4123062299065458
Foods_revenue_WI_1 0.3468869691794831
Hobbie_revenue_WI_2 0.4508340094604097
House_revenue_WI_2 0.44553607542325385
Foods_revenue_WI_2 0.4534678547567083
Hobbie_revenue_WI_3 0.6385074314274615
House_revenue_WI_3 0.5484290797721061
Foods_revenue_WI_3 0.677706374286333
```


2
1
0
−1
−2
0
50
100
150
200
250
300

colab

kaggle

jupyter

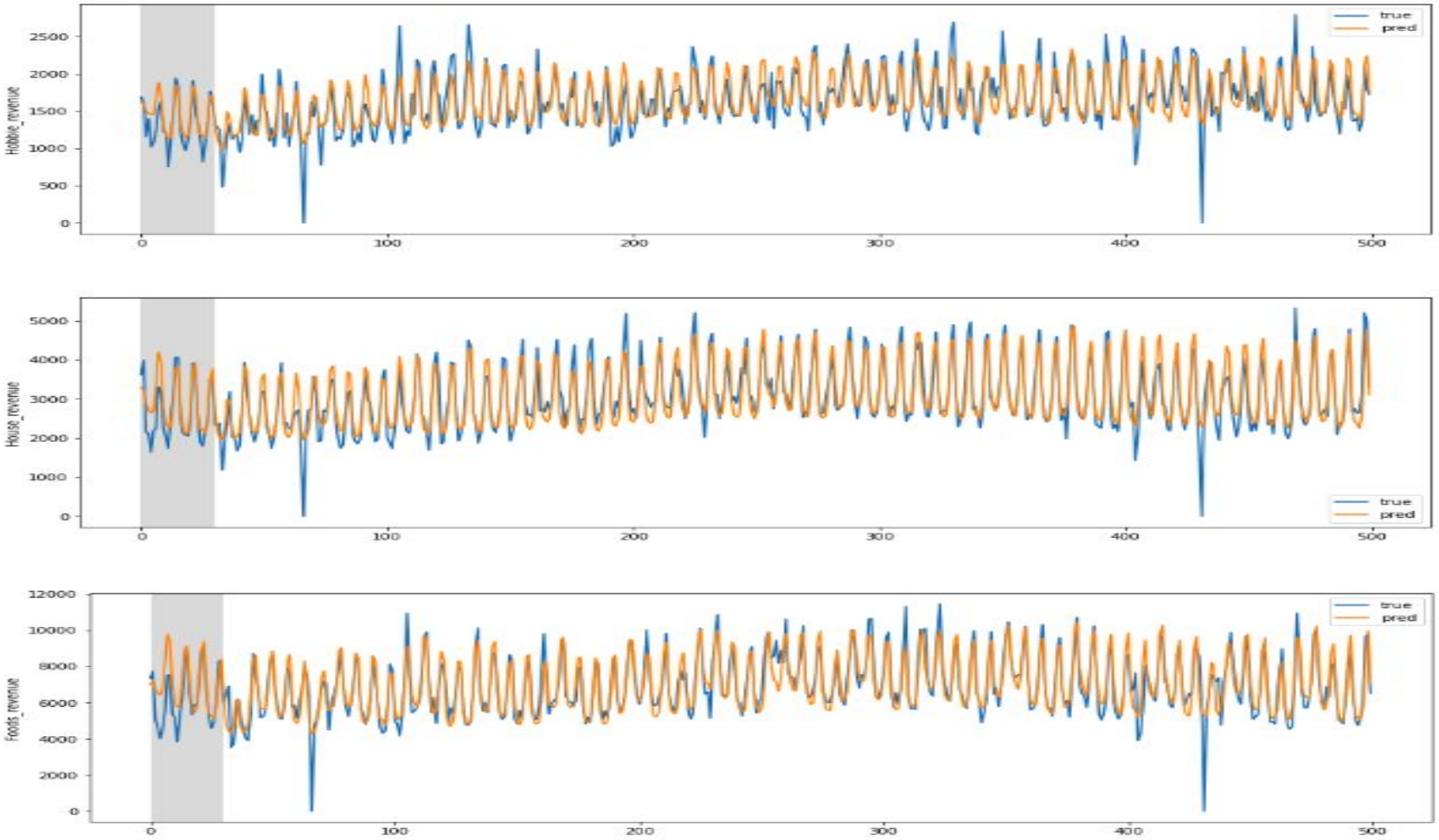


RNN Prediction

- We want to predict the Revenue 1 month into the future, we shift the data 30 time-steps
- Convert data to numpy arrays
- Train_split = 0.9
- Dense with 2 layers
- 31 input and 3 output signals
- Loss function = MSE
- Warmup problem
- Using GRU - dense network with 512 nodes to 3 nodes
- Batch usage
- Epoch

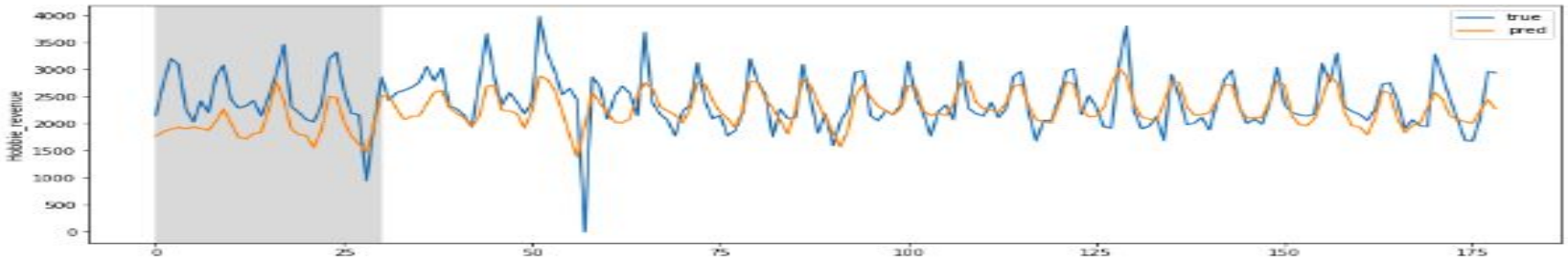
```
Train for 100 steps, validate on 1 samples
Epoch 1/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0127
Epoch 00001: val_loss improved from inf to 0.01733, saving model to 23
_checkpoint.keras
100/100 [==============================] - 44s 440ms/step - loss: 0.01
26 - val_loss: 0.0173
Epoch 2/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0071
Epoch 00002: val_loss improved from 0.01733 to 0.01198, saving model t
o 23_checkpoint.keras
100/100 [==============================] - 39s 389ms/step - loss: 0.00
71 - val_loss: 0.0120
Epoch 3/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0058
Epoch 00003: val_loss did not improve from 0.01198

Epoch 00003: ReduceLROnPlateau reducing learning rate to 0.00010000000
474974513.
100/100 [==============================] - 39s 392ms/step - loss: 0.00
58 - val_loss: 0.0154
Epoch 4/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0041
Epoch 00004: val_loss improved from 0.01198 to 0.01138, saving model t
o 23_checkpoint.keras
100/100 [==============================] - 39s 391ms/step - loss: 0.00
41 - val_loss: 0.0114
Epoch 5/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0038
Epoch 00005: val_loss improved from 0.01138 to 0.01102, saving model t
o 23_checkpoint.keras
100/100 [==============================] - 39s 389ms/step - loss: 0.00
38 - val_loss: 0.0110
Epoch 6/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0036
Epoch 00006: val_loss improved from 0.01102 to 0.01084, saving model t
o 23_checkpoint.keras
100/100 [==============================] - 39s 391ms/step - loss: 0.00
36 - val_loss: 0.0108
Epoch 7/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0035
Epoch 00007: val_loss did not improve from 0.01084

Epoch 00007: ReduceLROnPlateau reducing learning rate to 0.0001.
100/100 [==============================] - 39s 392ms/step - loss: 0.00
```

```
Epoch 8/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0034
Epoch 00008: val_loss improved from 0.01084 to 0.01080, saving model t
o 23_checkpoint.keras
100/100 [==============================] - 39s 390ms/step - loss: 0.00
34 - val_loss: 0.0108
Epoch 9/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0033
Epoch 00009: val_loss did not improve from 0.01080
100/100 [==============================] - 39s 391ms/step - loss: 0.00
33 - val_loss: 0.0110
Epoch 10/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0032
Epoch 00010: val_loss did not improve from 0.01080
100/100 [==============================] - 39s 391ms/step - loss: 0.00
32 - val_loss: 0.0114
Epoch 11/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0031
Epoch 00011: val_loss did not improve from 0.01080
100/100 [==============================] - 39s 391ms/step - loss: 0.00
31 - val_loss: 0.0126
Epoch 12/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0031
Epoch 00012: val_loss did not improve from 0.01080
100/100 [==============================] - 39s 391ms/step - loss: 0.00
31 - val_loss: 0.0126
Epoch 13/20
 99/100 [============================>.] - ETA: 0s - loss: 0.0030
Epoch 00013: val_loss did not improve from 0.01080
100/100 [==============================] - 39s 390ms/step - loss: 0.00
30 - val_loss: 0.0134
Epoch 00013: early stopping
CPU times: user 7min 18s, sys: 2min 37s, total: 9min 56s
Wall time: 8min 32s
```

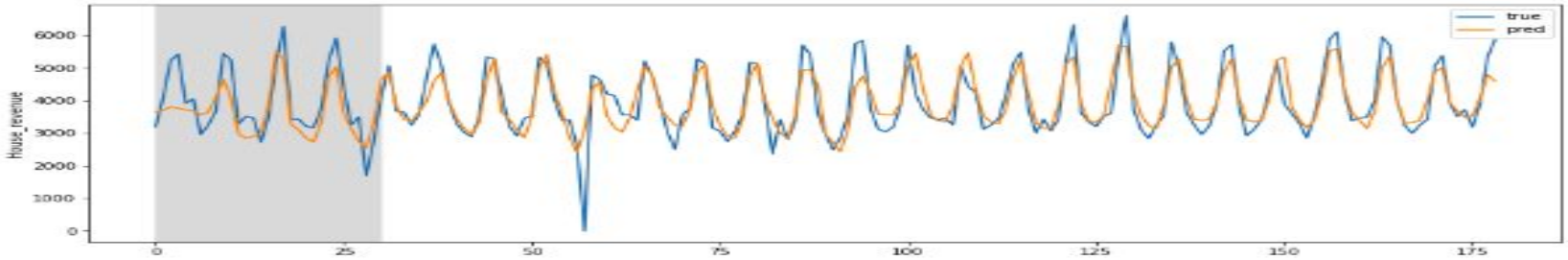

Hobbies_revenue
2500
2000
1500
1000
500
0
0
100
200
300
400
500
true
pred
House_revenue
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
true
pred
Foods_revenue
12000
10000
8000
6000
4000
2000
0
0
100
200
300
400
500
true
pred

0.009594 MSE

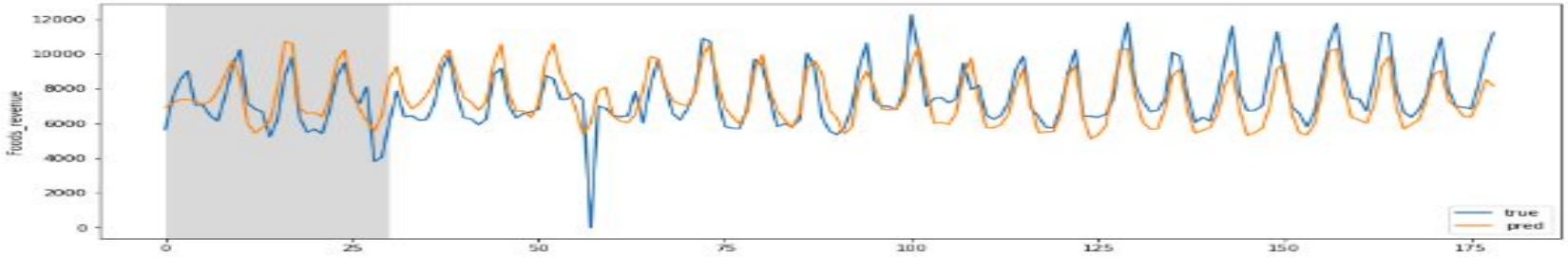
RNN usage




```
566.2322860647323
```




```
1183.485632666087
```



Thus sales forecasted with extreme accuracy

Our Team

[Neo You Shun helped with some parts]

Data Processing — **Joe V Akkara**

James Le Ngoc Canh — Data Visualization and analysis

Random Forest — **James Le Ngoc Canh**

Joe V Akkara — RNN networks

Thank you

Team Matrix

Quote from “The Matrix”

"There is a difference between knowing the path and walking the path."