Sales Amount Forecasting

by AI Masters Group

Why is it important to solve the problems?

- Global retail enterprises require detailed data analysis to understand market dynamics, optimize inventory, enhance customer satisfaction, and develop effective marketing strategies.
- Forecasting can help businesses optimize inventory improve supply chain management, and stay ahead of market trends.
- Machine learning can further improve forecast accuracy, driving more informed and effective decisions.

Project Goals

- Our project is designed to assist Walmart with accurate daily sales forecasting.
- Our goal is to create models to predict the sales of individual items over the next 28 days.

UNIVERSITY OF NICOSIA · FEATURED PREDICTION COMPETITION · 5 YEARS AGO

Late Submission

M5 Forecasting - Accuracy

Estimate the unit sales of Walmart retail goods

Overview

Start
Mar 3, 2020

Close
Jul 1, 2020

Merger & Entry

Description

Note: This is one of the two complementary competitions that together comprise the M5 forecasting challenge. Can you estimate, as precisely as possible, the point forecasts of the unit sales of various products sold in the USA by Walmart? If you are interested in estimating the uncertainty distribution of the realized values of the same series, be sure to check out its companion competition

How much camping gear will one store sell each month in a year? To the uninitiated, calculating sales at this level may seem as difficult as predicting the weather. Both types of forecasting rely on science and historical data. While a wrong weather forecast may result in you carrying around an umbrella on a sunny day, inaccurate business forecasts could result in actual or opportunity losses. In this competition, in addition to traditional forecasting methods you're also challenged to use machine learning to improve forecast accuracy.

Competition Host
University of Nicosia

Prizes & Awards
$50,000
Awards Points & Medals

Participation
31,968 Entrants
7,022 Participants
5,558 Teams
88,741 Submissions

Tags
Time Series Analysis
Custom Metric

Table of Contents

Columns in the dataset

Our Dataset recorded order data of Walmart's USA markets across 1969 days since 29/1/2011.

sales_train_validation.csv
sales_train_evaluation.csv

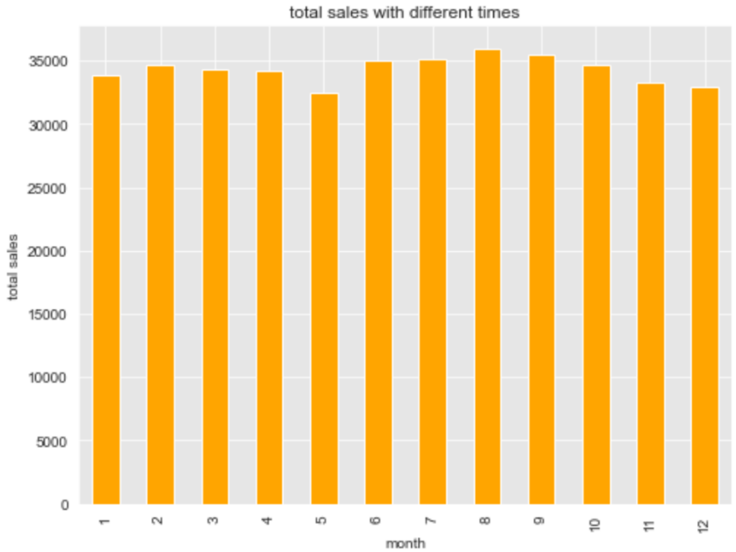
- id
- item_id
- dept_id
- cat_id
- store_id
- state_id
- d_1 to d_1941

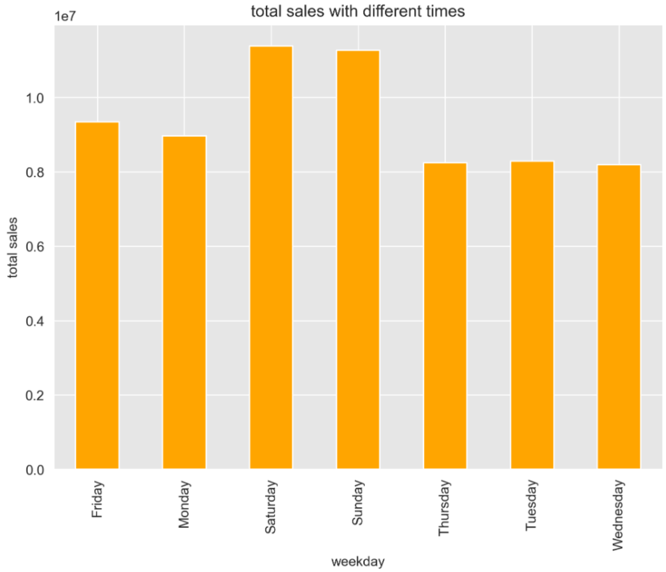
sell_prices.csv

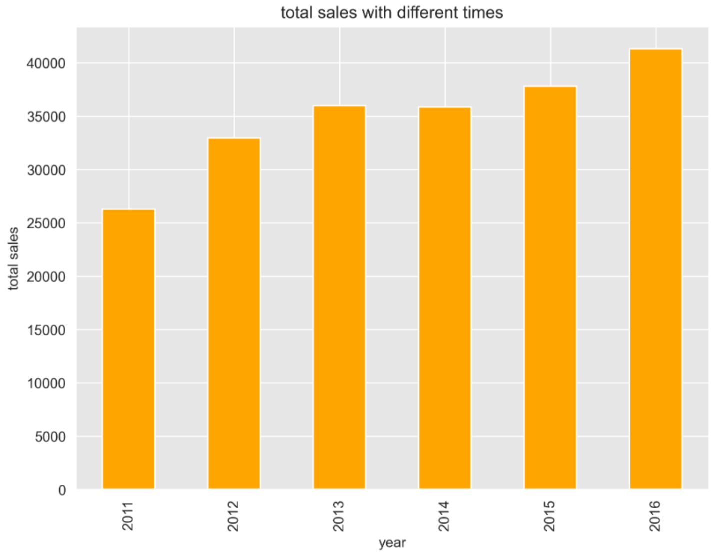
- store_id
- item_id
- wm_yr_wk
- sell_price

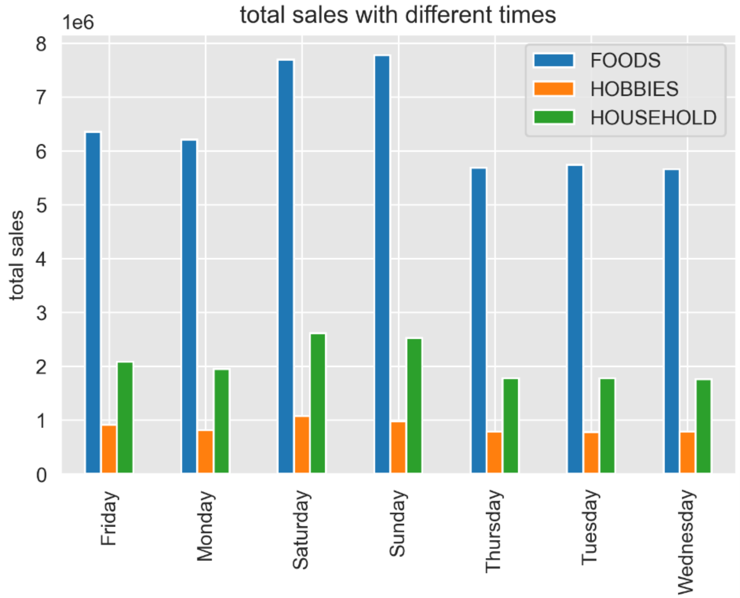
calendar.csv

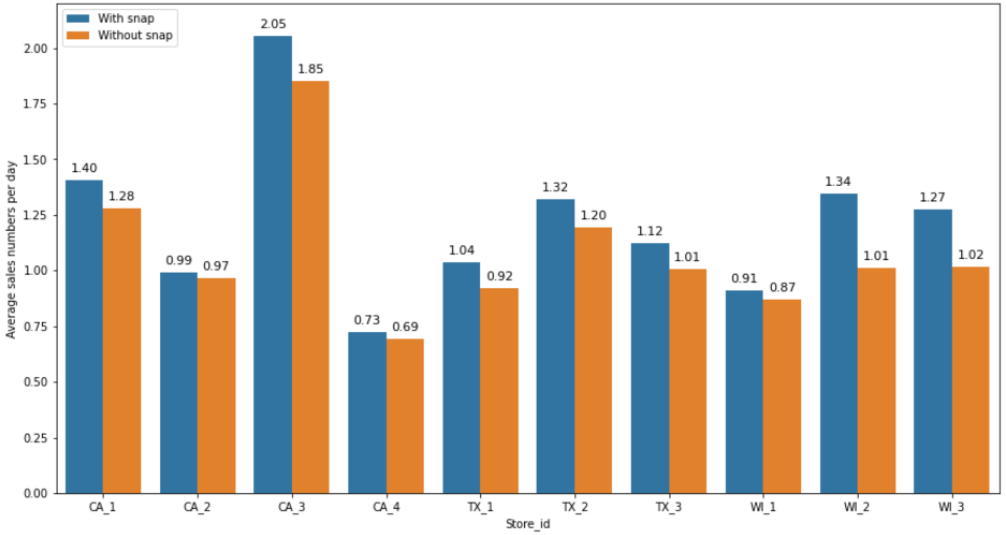
- date
- wm_yr_wk
- month
- year
- d
- event_name_1 and event_name_2
- event_type_1 and event_type_2
- snap_CA, snap_TX, snap_WI

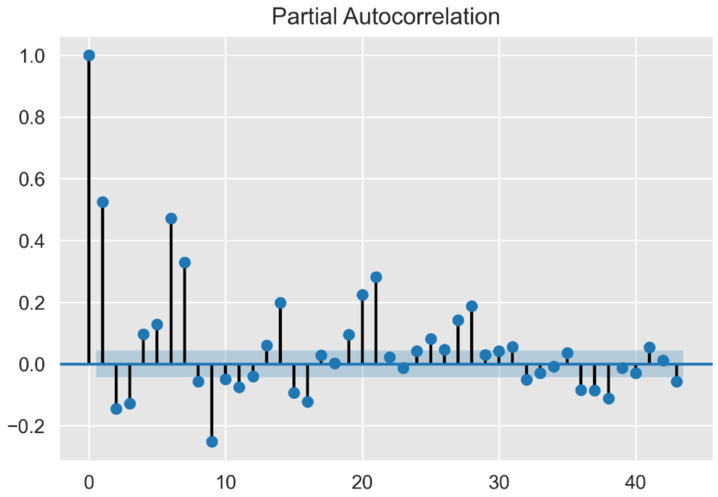

total sales with different times
total sales
month


total sales with different times
1e7
total sales
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
weekday


total sales with different times
total sales
year


total sales with different times
1e6
FOODS
HOBBIES
HOUSEHOLD
total sales
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday


With snap
Without snap
Average sales numbers per day
1.40
1.28
0.99
0.97
2.05
1.85
0.73
0.69
1.04
0.92
1.32
1.20
1.12
1.01
0.91
0.87
1.34
1.01
1.27
1.02
CA_1
CA_2
CA_3
CA_4
TX_1
TX_2
TX_3
WI_1
WI_2
WI_3
Store_id


Partial Autocorrelation

calendar.csv	sales_train_evaluation.csv	sales_train_validation.csv	sell_prices.csv

Total_data_with_price.csv

Shape: (58327370, 19)

	id	item_id	dept_id	cat_id	store_id	state_id	d	num_sold	date	wm_yr_wk	weekday	month	year	event_name_1	event_type_1	event_name_2	event_type_2	snap	sell_price
0	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0	2011-01-29	11101	Saturday	1	2011	NaN	NaN	NaN	NaN	0	9.58
1	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_2	0	2011-01-30	11101	Sunday	1	2011	NaN	NaN	NaN	NaN	0	9.58
2	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_3	0	2011-01-31	11101	Monday	1	2011	NaN	NaN	NaN	NaN	0	9.58
3	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_4	0	2011-02-01	11101	Tuesday	2	2011	NaN	NaN	NaN	NaN	1	9.58
4	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_5	0	2011-02-02	11101	Wednesday	2	2011	NaN	NaN	NaN	NaN	1	9.58
5	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_6	0	2011-02-03	11101	Thursday	2	2011	NaN	NaN	NaN	NaN	1	9.58
6	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_7	0	2011-02-04	11101	Friday	2	2011	NaN	NaN	NaN	NaN	1	9.58
7	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_8	0	2011-02-05	11102	Saturday	2	2011	NaN	NaN	NaN	NaN	1	9.58
8	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_9	0	2011-02-06	11102	Sunday	2	2011	SuperBowl	Sporting	NaN	NaN	1	9.58
9	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_10	0	2011-02-07	11102	Monday	2	2011	NaN	NaN	NaN	NaN	1	9.58

Total_data_with_price.csv

Shape: (58327370, 19)

	id	item_id	dept_id	cat_id	store_id	state_id	d	num_sold	date	wm_yr_wk	weekday	month	year	event_name_1	event_type_1	event_name_2	event_type_2	snap	sell_price
0	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0	2011-01-29	11101	Saturday	1	2011	NaN	NaN	NaN	NaN	0	9.58
1	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_2	0	2011-01-30	11101	Sunday	1	2011	NaN	NaN	NaN	NaN	0	9.58
2	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_3	0	2011-01-31	11101	Monday	1	2011	NaN	NaN	NaN	NaN	0	9.58
3	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_4	0	2011-02-01	11101	Tuesday	2	2011	NaN	NaN	NaN	NaN	1	9.58
4	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_5	0	2011-02-02	11101	Wednesday	2	2011	NaN	NaN	NaN	NaN	1	9.58
5	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_6	0	2011-02-03	11101	Thursday	2	2011	NaN	NaN	NaN	NaN	1	9.58
6	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_7	0	2011-02-04	11101	Friday	2	2011	NaN	NaN	NaN	NaN	1	9.58
7	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_8	0	2011-02-05	11102	Saturday	2	2011	NaN	NaN	NaN	NaN	1	9.58
8	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_9	0	2011-02-06	11102	Sunday	2	2011	SuperBowl	Sporting	NaN	NaN	1	9.58
9	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_10	0	2011-02-07	11102	Monday	2	2011	NaN	NaN	NaN	NaN	1	9.58

FOODS
training + validation + testing

HOBBIES
training + validation + testing

HOUSEHOLD
training + validation + testing

Method: Regression Model

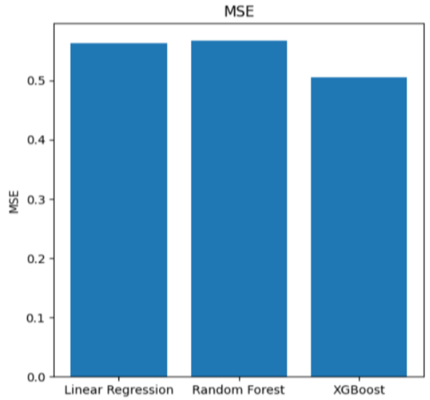
```
top_200_ids = data.groupby('id')['sell_price'].sum().sort_values(ascending=False).head(200).index
filtered_data = data[data['id'].isin(top_200_ids)]
```

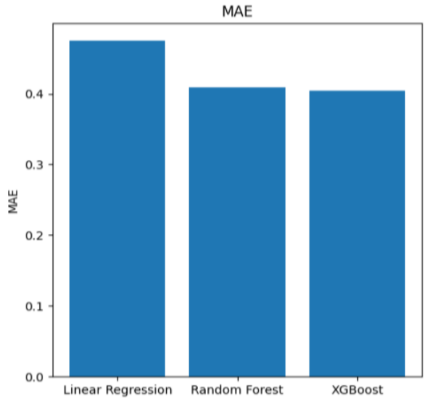
```
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_features),
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)
    ])
```

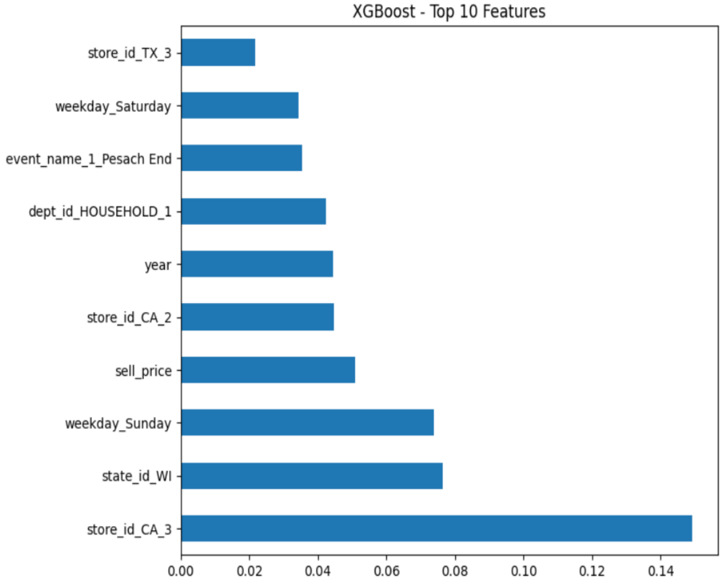
Model performance comparison:

	MSE	MAE	R2
Linear Regression	0.562677	0.475336	0.115883
Random Forest	0.567984	0.408385	0.107543
XGBoost	0.505545	0.404256	0.205653

```
models = {
    'Linear Regression': LinearRegression(),
    'Random Forest': RandomForestRegressor(random_state=42),
    'XGBoost': XGBRegressor(random_state=42)
}
```







Feature importance ranking:

XGBoost:

store_id_CA_3	0.149199
state_id_WI	0.076547
weekday_Sunday	0.073919
sell_price	0.050889
store_id_CA_2	0.044574
year	0.044465
dept_id_HOUSEHOLD_1	0.042428
event_name_1_Pesach End	0.035372
weekday_Saturday	0.034380
store_id_TX_3	0.021790

Method: LSTM Model

```python
LOOKBACK_MAX    = 28    #28? 14?
LOOKBACK_ARR    = np.array([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14])
```

```python
history = model.fit(x = x_time,
                    y = y_time,
                    epochs=10,
                    shuffle=True,
                    batch_size=128,
                    validation_split = 0.1,
                    verbose=1)
```

```python
model = Sequential()
model.add(Input(shape=(LOOKBACK_ARR.shape[0], x_data.shape[1])))
model.add(LSTM(64, activation='relu', return_sequences=True))
model.add(LSTM(64, activation='relu'))
model.add(Dense(1, activation='relu'))
model.compile(optimizer='adam', loss='mse')
model.summary()
```

```python
def split_data(category):
    df                  = pd.read_csv('train_with_price.csv')
    category_data       = df[df['cat_id'] == category]
    train_df, temp_df = train_test_split(category_data, test_size=0.2, random_state=42)
    test_df, val_df   = train_test_split(temp_df, test_size=0.5, random_state=42)
    train_df.to_csv(f'{category}_train_dataset.csv', index=False)
    test_df.to_csv (f'{category}_test_dataset.csv', index=False)
    val_df.to_csv  (f'{category}_validation_dataset.csv', index=False)
```

```python
def create_xy_data(df, pre_type = ""): # for one item at once
  x_train_id = (df['id'] + "_" + pre_type).values
  idx = np.unique(x_train_id, return_index=True)[1]
  idx.sort()
  x_train_id = x_train_id[idx]
  y_train = df['num_sold'].values
  df = df.drop(['id','num_sold','item_id', 'dept_id', 'year'],axis=1)
  return df, y_train, x_train_id
```

```python
train_dummy     = pd.get_dummies(item_train_data, columns=['store_id','state_id', 'weekday', 'snap'], drop_first=True)
train_columns   = train_dummy.columns
val_dummy       = pd.get_dummies(item_val_data,   columns=['store_id','state_id', 'weekday', 'snap'], drop_first=True)
eval_dummy      = pd.get_dummies(item_eval_data,  columns=['store_id','state_id', 'weekday', 'snap'], drop_first=True)
```

Business Problem ... ings   Conclusion

```
LOOKBACK_MAX    = 28    #28? 14?
LOOKBACK_ARR    = np.array([0,1,2,3,
```

```
history = model.fit(x = x_ti
                    y = 
                    epoc
                    shuf
                    batc
                    vali
                    verb
```

```
model = Sequential()
model.add(Input(shape=(LOOKBACK_ARR
model.add(LSTM(64, activation='relu
model.add(LSTM(64, activation='relu
model.add(Dense(1, activation='relu
model.compile(optimizer='adam', los
model.summary()
```

```
train_dummy     = pd.get_dummi
train_columns   = train_dummy.
val_dummy       = pd.get_dummi
eval_dummy      = pd.get_dummi
```

Layer (type)	Output Shape	Param #
lstm_458 (LSTM)	(None, 15, 64)	20,736
lstm_459 (LSTM)	(None, 64)	33,024
dense_229 (Dense)	(None, 1)	65

Total params: 53,825 (210.25 KB)

Trainable params: 53,825 (210.25 KB)

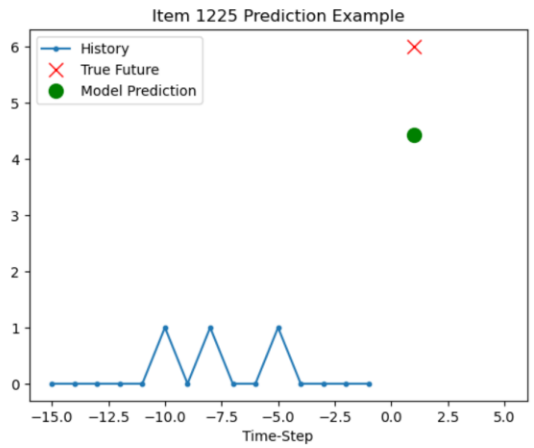
Non-trainable params: 0 (0.00 B)

```
Epoch 1/10
106/106 ━━━━━━━━━━━━━━━━━━━━ 7s 45ms/step - loss: 6403.7334 - val_loss: 30.7428
Epoch 2/10
106/106 ━━━━━━━━━━━━━━━━━━━━ 4s 42ms/step - loss: 30.1137 - val_loss: 6.5796
Epoch 3/10
106/106 ━━━━━━━━━━━━━━━━━━━━ 5s 43ms/step - loss: 5.2300 - val_loss: 2.8025
Epoch 4/10
106/106 ━━━━━━━━━━━━━━━━━━━━ 4s 42ms/step - loss: 3.7690 - val_loss: 5.0772
Epoch 5/10
106/106 ━━━━━━━━━━━━━━━━━━━━ 5s 42ms/step - loss: 4.8840 - val_loss: 2.1489
Epoch 6/10
106/106 ━━━━━━━━━━━━━━━━━━━━ 5s 43ms/step - loss: 2.4137 - val_loss: 1.1928
Epoch 7/10
106/106 ━━━━━━━━━━━━━━━━━━━━ 4s 41ms/step - loss: 1.7161 - val_loss: 2.0740
Epoch 8/10
106/106 ━━━━━━━━━━━━━━━━━━━━ 4s 42ms/step - loss: 2.8033 - val_loss: 0.8048
Epoch 9/10
106/106 ━━━━━━━━━━━━━━━━━━━━ 5s 43ms/step - loss: 1.2330 - val_loss: 0.9527
Epoch 10/10
106/106 ━━━━━━━━━━━━━━━━━━━━ 5s 44ms/step - loss: 1.6963 - val_loss: 0.8504
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step
Sample raw predictions (after inverse transform and clipping): [1.7399429 1.7189262 0.        0.6741688 0.6174445]
RMSE =  2.818309
52/52 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
Predicted y_val range (after inverse transform and clipping): min=0.0, max=20.72597885131836
True y_val range (after inverse transform): min=1.6727970120200553e-08, max=14.999999046325684
```

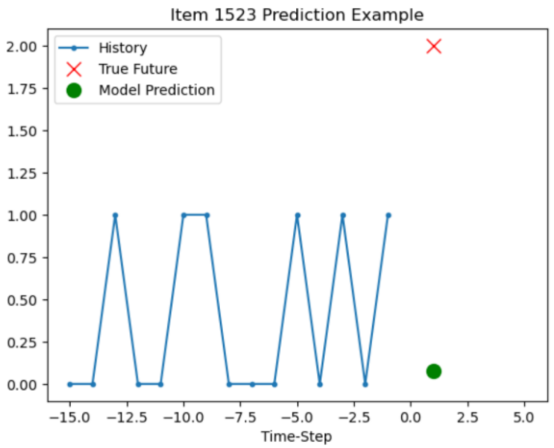
```
sv')
 test_size=0.2, random_state=42)
size=0.5, random_state=42)
dex=False)
ex=False)
', index=False)
```

```
e item at once
es
)[1]

t_id', 'year'],axis=1)
```

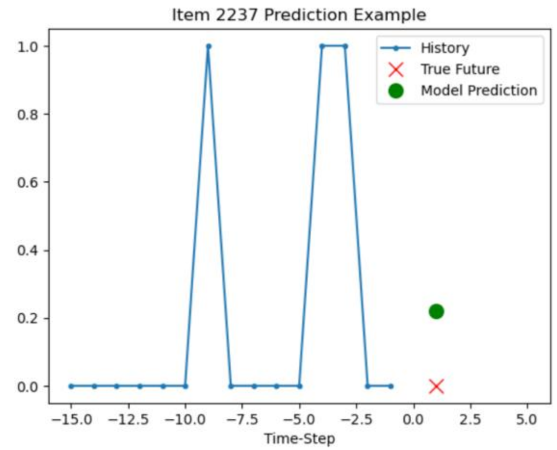
```
rue)

rue)
rue)
```



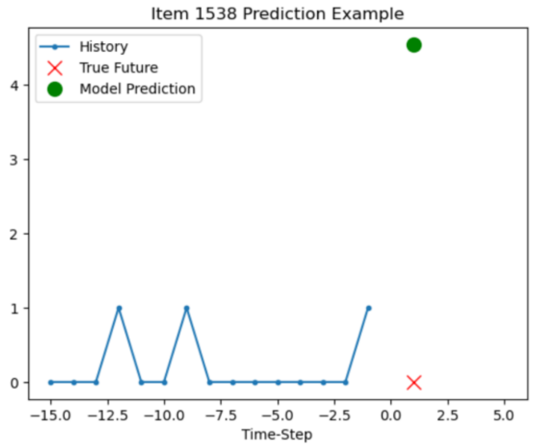
RMSE = 3.220477



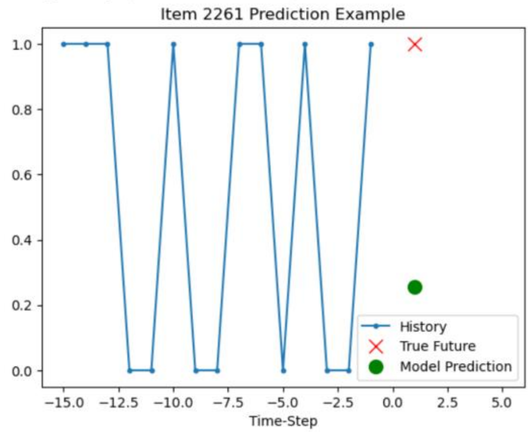
RMSE = 0.3962133



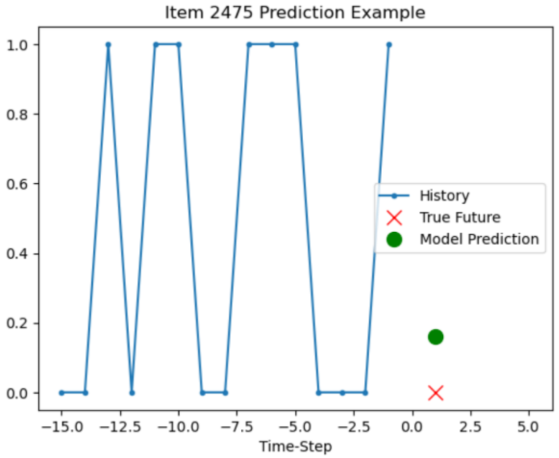
RMSE = 0.6189017



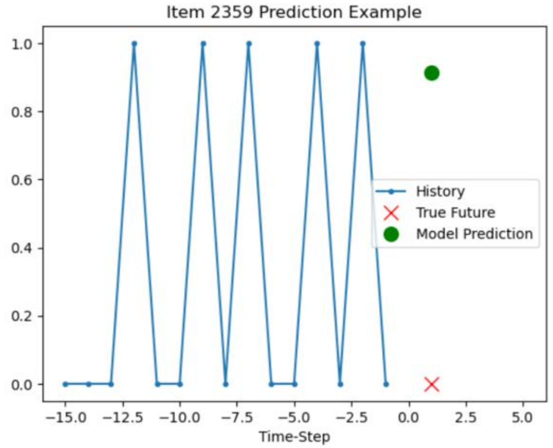
RMSE = 3.69305



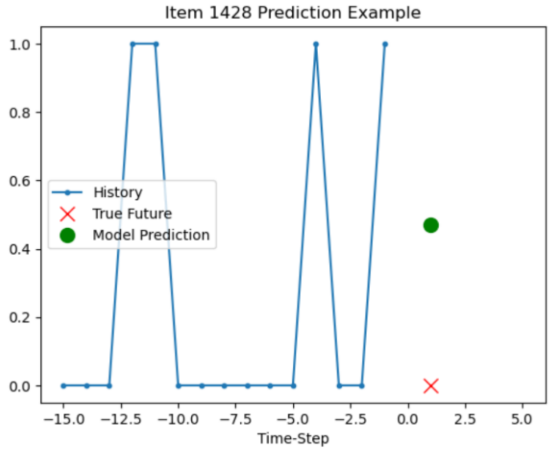
RMSE = 0.6024017



RMSE = 0.5926977



RMSE = 0.83067185



RMSE = 0.83738977

id	F1	F2	F3	F4	F5	F6	F7	F8	F9
HOUSEHOLD_1_225_TX_3_validation_validation	0	3.72288	4.534193	4.33041	4.519439	2.604469	2.497672	2.368856	1.387462
HOUSEHOLD_1_225_CA_3_validation_validation	0	3.535536	3.322803	1.300174	2.751408	2.178629	2.16878	3.503257	3.282847
HOUSEHOLD_1_225_CA_1_validation_validation	2.466105	0	0	0	0.765515	0	0.213705	5.03897	4.606165
HOUSEHOLD_1_225_WI_3_validation_validation	1.855766	2.683368	2.310402	4.024885	2.446894	2.747065	1.522343	3.720071	6.292375
HOUSEHOLD_1_225_TX_1_validation_validation	2.323509	1.735151	0.417039	2.290968	0.455946	1.840039	5.295066	0	2.805219
HOUSEHOLD_1_225_CA_2_validation_validation	3.481088	3.279253	5.707238	4.124717	0.669165	0.11039	0	0	0
HOUSEHOLD_1_225_CA_4_validation_validation	2.007461	0.484292	0	1.308338	4.12614	5.661965	6.220965	7.118368	5.453558
HOUSEHOLD_1_225_TX_2_validation_validation	2.123508	0.528142	0	0	2.963317	2.916285	4.533968	3.459332	1.35537
HOUSEHOLD_1_225_WI_1_validation_validation	5.875822	3.22357	2.05704	4.083245	3.909307	0	0.150121	5.48372	0
HOUSEHOLD_1_225_WI_2_validation_validation	4.582798	4.208896	3.86248	2.276813	4.016234	2.651427	3.431772	4.245406	2.546614
HOUSEHOLD_1_523_TX_2_validation_validation	1.81948	0	0	1.752077	2.862362	1.934778	1.356793	1.839065	1.779787
HOUSEHOLD_1_523_WI_3_validation_validation	3.338043	4.79628	3.746939	3.690301	2.230604	0	6.561465	1.911261	1.780798
HOUSEHOLD_1_523_CA_2_validation_validation	1.495045	4.189199	6.664817	5.608717	5.129328	4.170701	4.721668	5.257751	1.365893
HOUSEHOLD_1_523_CA_3_validation_validation	3.786464	3.04364	3.837278	1.232246	1.378999	1.226985	2.878689	1.715567	4.240127
HOUSEHOLD_1_523_CA_4_validation_validation	7.713728	6.778467	7.217938	5.837289	4.645896	4.134191	0.838498	0	0.287175
HOUSEHOLD_1_523_WI_2_validation_validation	4.436346	4.70757	4.557447	2.692693	2.482768	2.044702	2.014089	3.197657	2.050824
HOUSEHOLD_1_523_TX_3_validation_validation	1.516539	3.260791	2.988107	2.749986	4.340445	3.378036	4.764133	4.539492	5.326109
HOUSEHOLD_1_523_CA_1_validation_validation	7.061749	6.165058	4.898921	2.698347	2.773127	3.730743	2.140677	2.704451	1.444567
HOUSEHOLD_1_523_WI_1_validation_validation	5.899937	6.2181	2.882358	0.356787	1.092459	2.266328	3.21638	2.954967	1.040783
HOUSEHOLD_1_523_TX_1_validation_validation	4.92723	5.423319	2.416975	1.008654	0.656621	0	1.381058	1.267484	0
HOUSEHOLD_1_248_TX_1_validation_validation	3.79107	3.503519	2.860939	3.247947	1.966682	1.508264	2.110701	2.03768	2.048465
HOUSEHOLD_1_248_CA_1_validation_validation	2.655284	2.4955	0.552819	0	2.198925	3.913032	5.33371	3.917039	3.732728
HOUSEHOLD_1_248_TX_2_validation_validation	0	3.21155	1.700251	0.993301	1.18278	3.807396	5.113151	5.870467	6.238527
HOUSEHOLD_1_248_TX_3_validation_validation	2.353503	1.329495	2.902355	2.312949	2.255469	2.232514	2.041126	4.823241	5.74225
HOUSEHOLD_1_248_WI_3_validation_validation	4.083938	4.745503	3.549803	1.558816	0.464858	0.117243	1.00783	0	0
HOUSEHOLD_1_248_WI_1_validation_validation	0.734884	1.614836	1.263439	0.677703	2.485352	0	0	1.601468	3.167662
HOUSEHOLD_1_248_WI_2_validation_validation	0	1.627643	1.922533	1.224383	1.826408	2.322872	5.013169	5.571832	1.505792
HOUSEHOLD_1_248_CA_4_validation_validation	1.946424	3.008403	0.806818	3.194249	3.219413	1.534663	3.363657	0.601724	1.0206

Training Method

1. Regression Model

Linear Regression
Random Forest
XGBoost

2. LSTM Model

Lookback:	28 days
Epochs:	10
Batch Size:	128
Neurons:	64
No. of Layers:	2
Activation:	ReLU
Data Convert:	One Hot Encoding

Future Improvements

1. Now we only judge if there is an event or holiday of the day. Different events as different values will be better.
2. Adjust model's configs for higher accuracy.

Conclusion

1. Model Implementation:

Regression models and Long Short-Term Memory (LSTM) networks aims to extract meaningful patterns from historical sales data, facilitating accurate sales.

2. Predictions Performance Metrics:

The evaluation of the LSTM model, demonstrates promising results with reduced forecasting errors, indicating its ability to learn from historical data and optimize inventory and supply chain management.

3. Strategic Importance of Machine Learning:

The project enabling retailers to better anticipate consumer behavior, align inventory with demand, and enhance overall operational efficiency.