from os. path import exists

import pandas as pd

df = pd.read\_csv("/content/train.csv")

df = df.sample(n = 19\_000,random\_state = 0)

df['store'] = df['store'].astype('str')

df['item'] = df['item'].astype('str')

df['date'] = pd.to\_datetime(df['date'])

df.sort\_values("date",inplace=True)

df.reset\_index(inplace=True,drop = True)

df.head()

# split dataset as training - 2013 to 2016 and test - 2017

train = df[df["date"]<"2017-01-01"]

test = df[df["date"]>="2017-01-01"]

train\_features = train.drop(columns = ["sales"])

train\_target = train["sales"]

test\_features = test.drop(columns = ["sales"])

test\_target = test["sales"]

from upgini import FeaturesEnricher, SearchKey

from upgini.metadata import CVType

enricher = FeaturesEnricher(

search\_keys = {

"date": SearchKey.DATE

},

cv = CVType.time\_series

)

enricher.fit(train\_features,train\_target,eval\_set=[(test\_features,test\_target)])

from catboost import CatBoostRegressor

from catboost.utils import eval\_metric

model = CatBoostRegressor(verbose = False,allow\_writing\_files=False,random\_state=0)

enricher.calculate\_metrics(

train\_features,train\_target,

eval\_set=[(test\_features,test\_target)],

estimator = model,

scoring = "mean\_absolute\_percentage\_error"

)

enriched\_train\_features = enricher.transform(train\_features,keep\_input=True)

enriched\_test\_features = enricher.transform(test\_features,keep\_input=True)

enriched\_train\_features.head()

model.fit(enriched\_train\_features,train\_target)

enriched\_preds = model.predict(enriched\_test\_features)

eval\_metric(test\_target.values,enriched\_preds,"SMAPE")

enriched\_preds

from sklearn.metrics import r2\_score, mean\_absolute\_error

# Calculate R-squared

r2 = r2\_score(test\_target, enriched\_preds)

# Calculate Mean Absolute Error (MAE)

mae = mean\_absolute\_error(test\_target, enriched\_preds)

print("R-squared:", r2)

print("MAE:", mae)

import pandas as pd

test\_data = pd.read\_csv('/content/test.csv')

test\_data['store'] = test\_data['store'].astype("str")

test\_data['item'] = test\_data['item'].astype("str")

test\_data['date'] = pd.to\_datetime(test\_data['date'])

enriched\_test\_features = enricher.transform(test\_data, keep\_input=True)

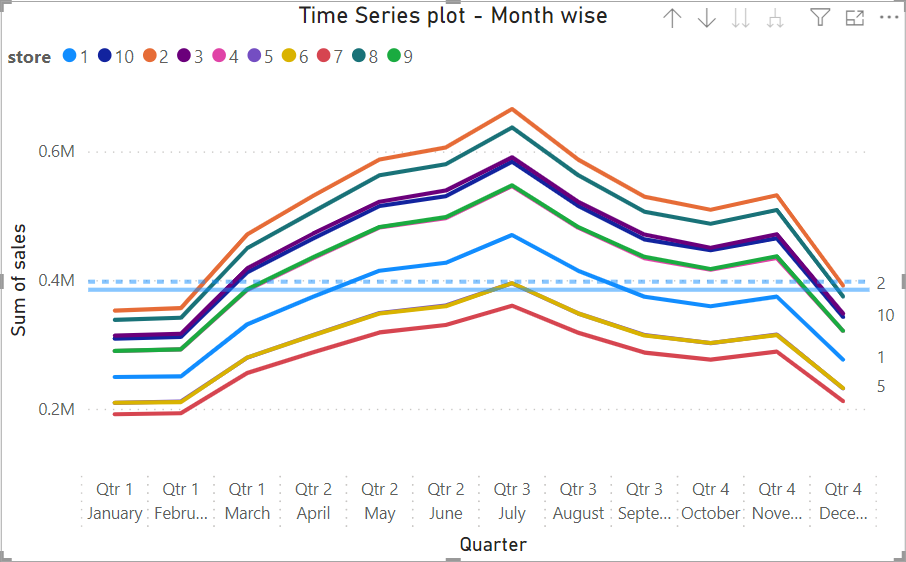
enriched\_preds = model.predict(enriched\_test\_features)

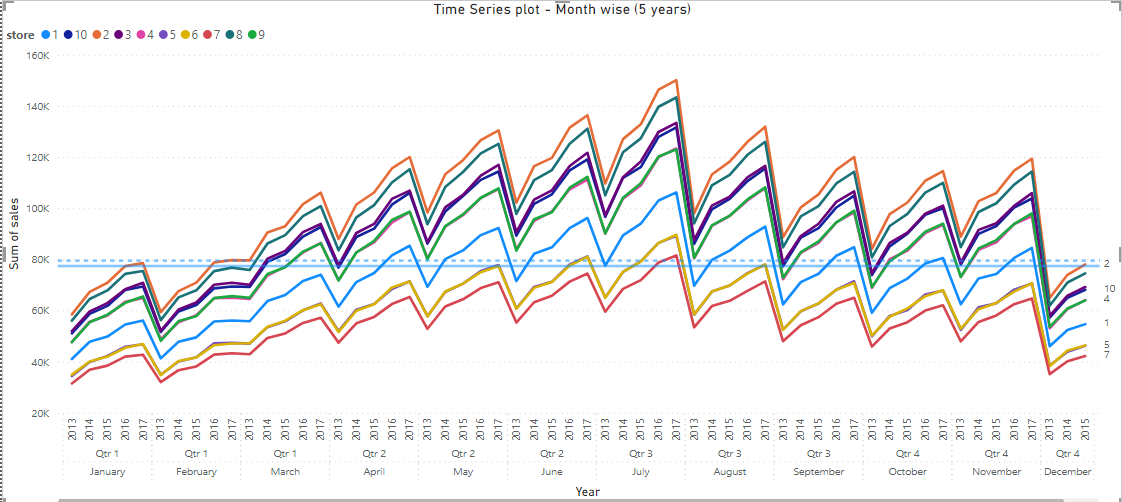
# Add the predicted sales values to the test data

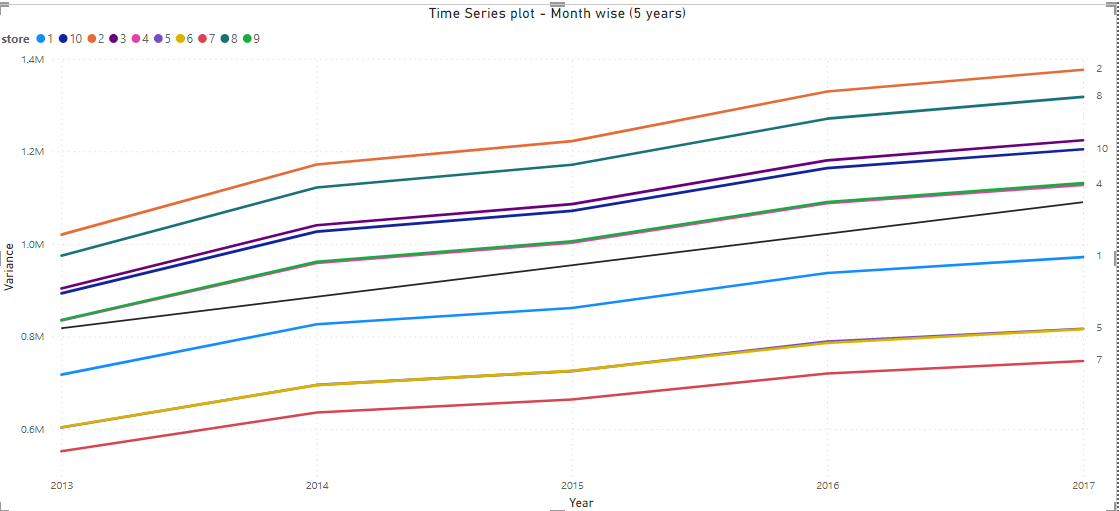
test\_data['predicted\_sales'] = enriched\_preds.round()

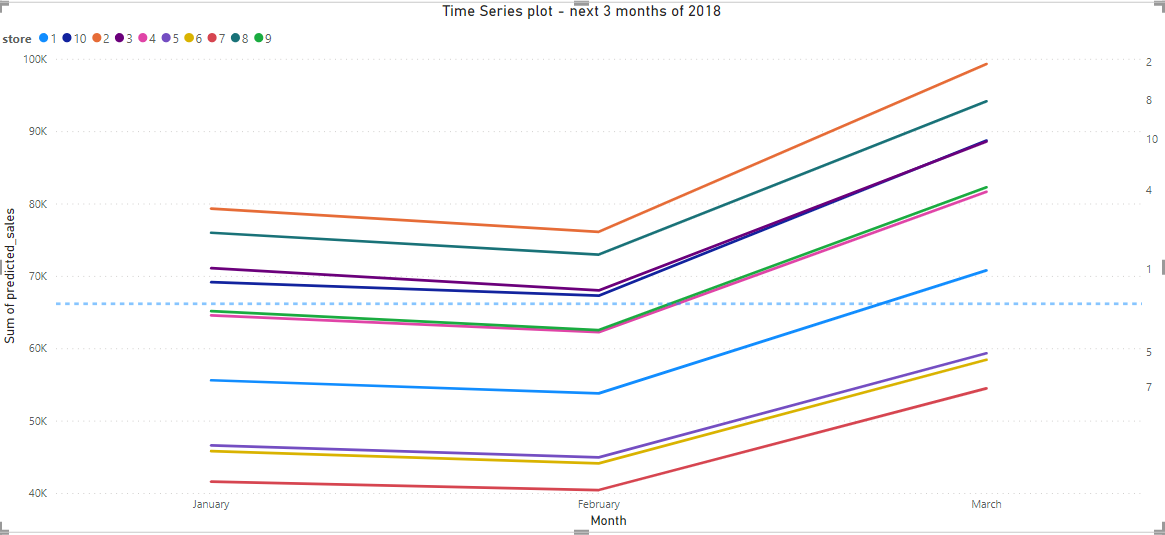
# Save the test data with predicted sales to a CSV file

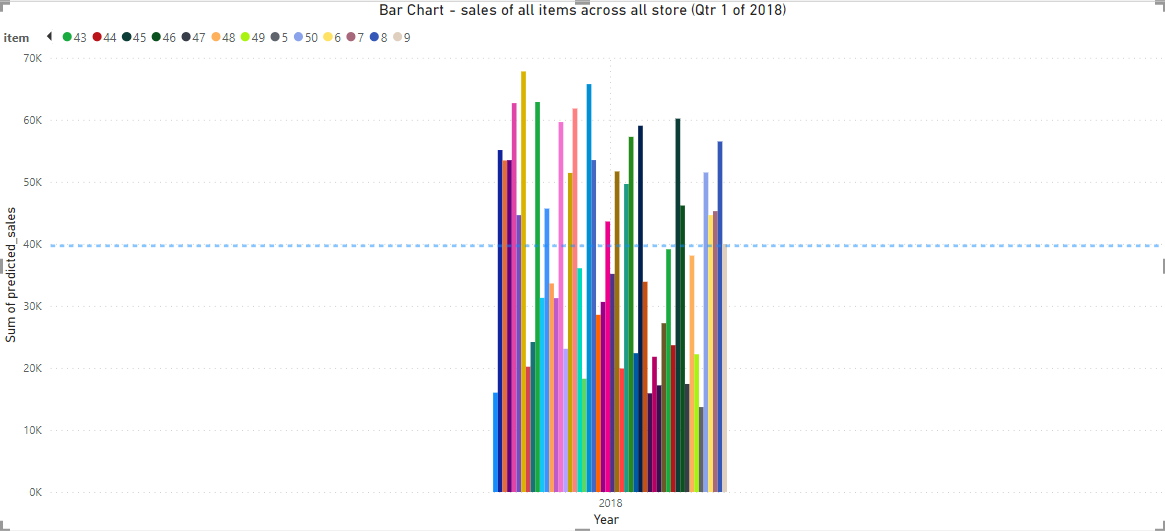
test\_data.to\_csv('test\_with\_predictions.csv', index=False)



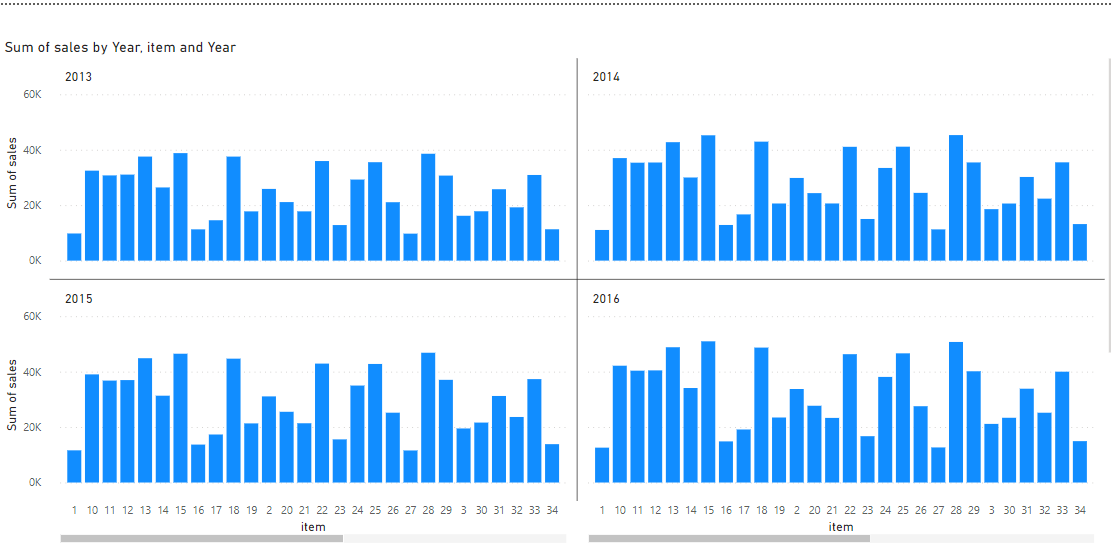




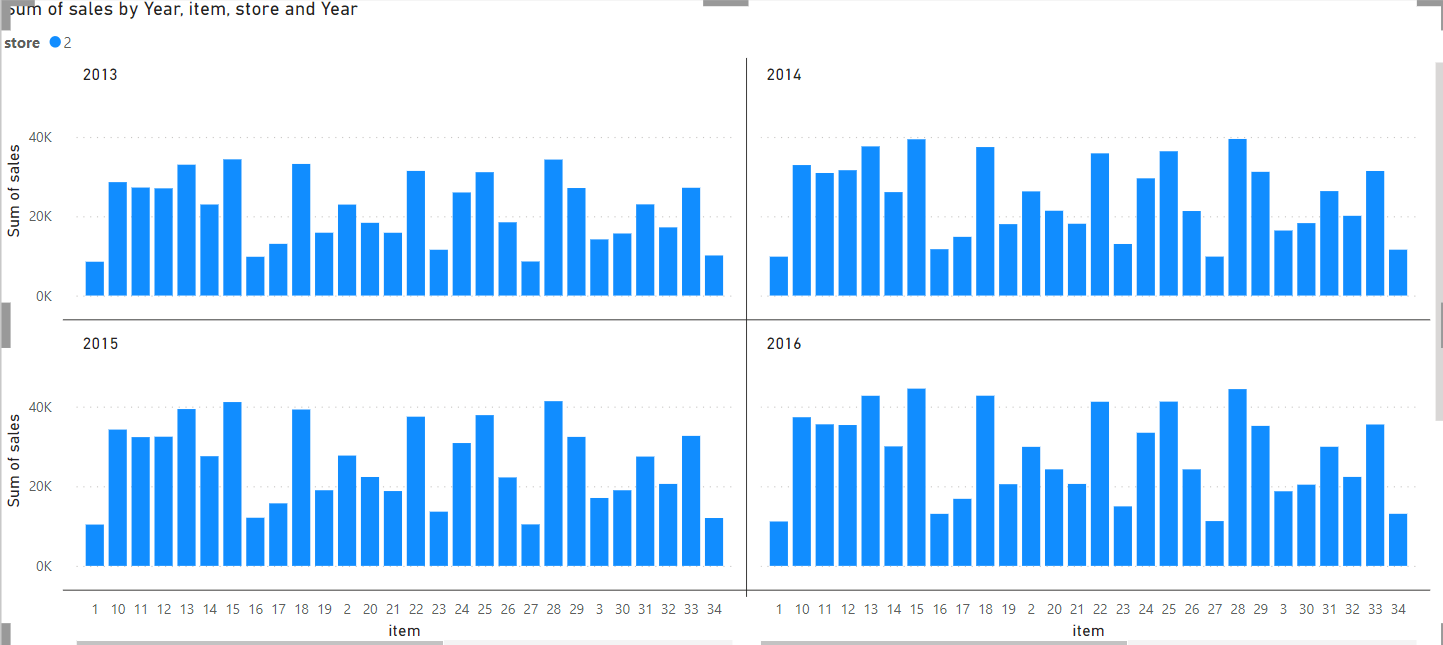




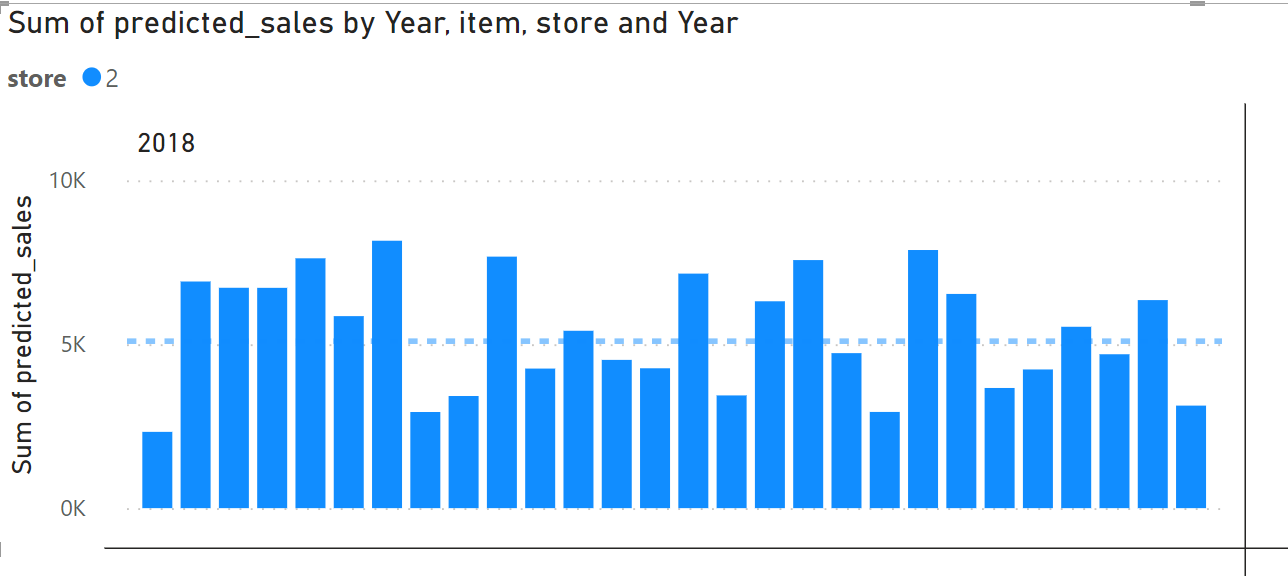
Item-wise predicted sales in 2018 Qtr 1



Item-wise sales from 2013 to 2017



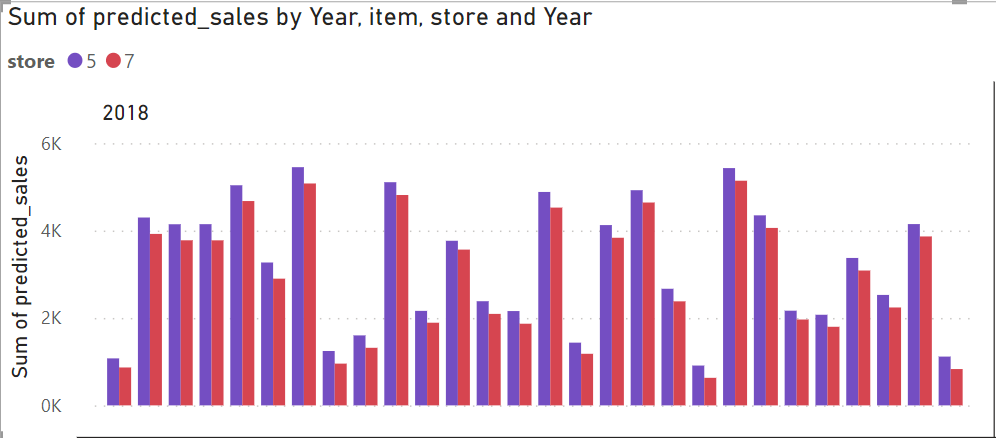
Item-wise sales for store 2 (2013 to 2017)



Item-wise sales of store 2 (2018 Qtr 1)



Item-wise sales stores 5 and 7 (2013 to 2017)



**ANALYSIS:**

For Store 2, it has consistently been the highest seller over the 5 years. In the next 3 months of 2018, it is expected to maintain its position as the highest seller.

Store 3 and Store 10 have been close competitors and have performed well in the past 5 years. They are likely to continue their close competition in the future.

Stores 5 and 7 have shown poor performance over the 5 years, with significantly lower sales compared to other stores. There is not much growth expected for them in the future.

Surprisingly, Store 6 is expected to experience a downtrend in sales in the next 3 months of 2018.

In terms of specific items:

For Store 2, the highest-selling item in the years 2013 to 2017 was item 15, and it remains the highest-selling item in the first quarter of 2018 according to predictions. The lowest-selling items in 2018 Qtr 1 are item 1 and item 27, which was consistent with their performance from 2013 to 2017.

For Stores 5 and 7, in 2018 Qtr 1, the highest selling item is expected to be item 28, while the lowest selling item is item 7. In the years 2013 to 2017, the highest-selling item for both stores was item 15, and the lowest-selling item was item 1.

Based on these findings, it is recommended to focus on promoting and optimizing the sales of the highest-performing items, especially item 15. For stores with lower sales, further analysis and strategies may be needed to improve their performance.