MARASIGAN, VEM AIENSI A.
3BSCS-I

Documentation and Leaderboard
January 2, 2024

Solution only using train. - Try

0 Edit

Notebook Input Output Logs Comments (0) Settings

Add Tags

Importing Libraries

In [1]:
```
# library import
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from datetime import datetime
```

In [2]:
```
train_path = "/kaggle/input/store-sales-time-series-forecasting/train.csv"
test_path = "/kaggle/input/store-sales-time-series-forecasting/test.csv"
```

Next, import the data using pandas. After importing the data, I'll display the first few rows with the head() method to check the dataset's structure and contents.

Exploring Datasets

In [3]:
```
# data import
train_df = pd.read_csv(train_path)
# data_check
train_df.head()
```

Out[3]:

	id	date	store_nbr	family	sales	onpromotion
0	0	2013-01-01	1	AUTOMOTIVE	0.0	0
1	1	2013-01-01	1	BABY CARE	0.0	0
2	2	2013-01-01	1	BEAUTY	0.0	0
3	3	2013-01-01	1	BEVERAGES	0.0	0
4	4	2013-01-01	1	BOOKS	0.0	0

Then, convert the 'family' column (representing product categories) to one-hot encoded format using pd.get_dummies. By using **one-hot encoding**, each unique category in the 'family' column will be converted into a separate column with binary values (0 or 1).

CCSEL1-18 Professional ELective

[Store sales]

MARASIGAN, VEM AIENSI A.
3BSCS-I

Documentation and Leaderboard
January 2, 2024

In [4]:
```
train_df = pd.get_dummies(train_df, columns=['family'])
train_df.head()
```

Out[4]:

	id	date	store_nbr	sales	onpromotion	family_AUTOMOTIVE	family_BABY CARE	family_BEAUTY	family_BEVERAGES	family_BOOK
0	0	2013-01-01	1	0.0	0	True	False	False	False	False
1	1	2013-01-01	1	0.0	0	False	True	False	False	False
2	2	2013-01-01	1	0.0	0	False	False	True	False	False
3	3	2013-01-01	1	0.0	0	False	False	False	True	False
4	4	2013-01-01	1	0.0	0	False	False	False	False	True

5 rows × 38 columns

Table of Contents
- Importing Libraries
- Exploring Datasets
- Date and Time Conversion
- Solution with...
- Submission

Date and Time Conversion

Convert date to date time format. Then, add day of the week and month as new columns.

In [5]:
```
train_df['date'] = pd.to_datetime(train_df['date'])

train_df['day_of_week'] = train_df['date'].dt.dayofweek
train_df['month'] = train_df['date'].dt.month

train_df[['id', 'date', 'day_of_week', 'month']]
```

Out[5]:

	id	date	day_of_week	month
0	0	2013-01-01	1	1
1	1	2013-01-01	1	1
2	2	2013-01-01	1	1
3	3	2013-01-01	1	1
4	4	2013-01-01	1	1
...	...	...	...	...
3000883	3000883	2017-08-15	1	8
3000884	3000884	2017-08-15	1	8
3000885	3000885	2017-08-15	1	8
3000886	3000886	2017-08-15	1	8
3000887	3000887	2017-08-15	1	8

3000888 rows × 4 columns

Table of Contents
- Importing Libraries
- Exploring Datasets
- Date and Time Conversion
- Solution with...
- Submission

MARASIGAN, VEM AIENSI A.
3BSCS-I

Documentation and Leaderboard
January 2, 2024

Solution with RandomForestRegressor

Let's create and train our model. In this step, I will use a RandomForestRegressor.

In [6]:
```
# Model training with RandomForest
X = train_df.drop(['sales', 'date', 'onpromotion', 'store_nbr'], axis=1)
y = train_df['sales']

# Splitting data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Creating and training the RandomForest model
model = RandomForestRegressor(n_estimators=50, max_depth=10, n_jobs=-1, random_state=42)
model.fit(X_train, y_train)
```

Out[6]:
```
RandomForestRegressor
RandomForestRegressor(max_depth=10, n_estimators=50, n_jobs=-1, random_state=42)
```

In [7]:
```
# prediction
y_pred = model.predict(X_test)

# RMSLE
log_actual = np.log1p(y_test)
log_pred = np.log1p(y_pred)

rmsle = np.sqrt(np.mean((log_pred - log_actual) ** 2))

print("RMSLE:", rmsle)
```

Submission

Finally, create the file for the submission using test data.

In [8]:
```
# apply the model to test data for submission
test_df = pd.read_csv(test_path)
test_df = pd.get_dummies(test_df, columns=['family'])

test_df['date'] = pd.to_datetime(test_df['date'])
test_df['day_of_week'] = test_df['date'].dt.dayofweek
test_df['month'] = test_df['date'].dt.month

X_submit = test_df.drop(['date', 'onpromotion', 'store_nbr'], axis=1)
y_pred_submit = model.predict(X_submit)

submission_df = pd.DataFrame({
    'id': test_df['id'],
    'sales': y_pred_submit
})

submission_df.to_csv('submission.csv', index=False)
```

I only forked this solution from the public notebook, simple solution by Junko k

Table of Contents
- Importing Libraries
- Exploring Datasets
- Date and Time Conversion
- Solution with...
- Submission

Table of Contents
- Importing Libraries
- Exploring Datasets
- Date and Time Conversion
- Solution with...
- Submission

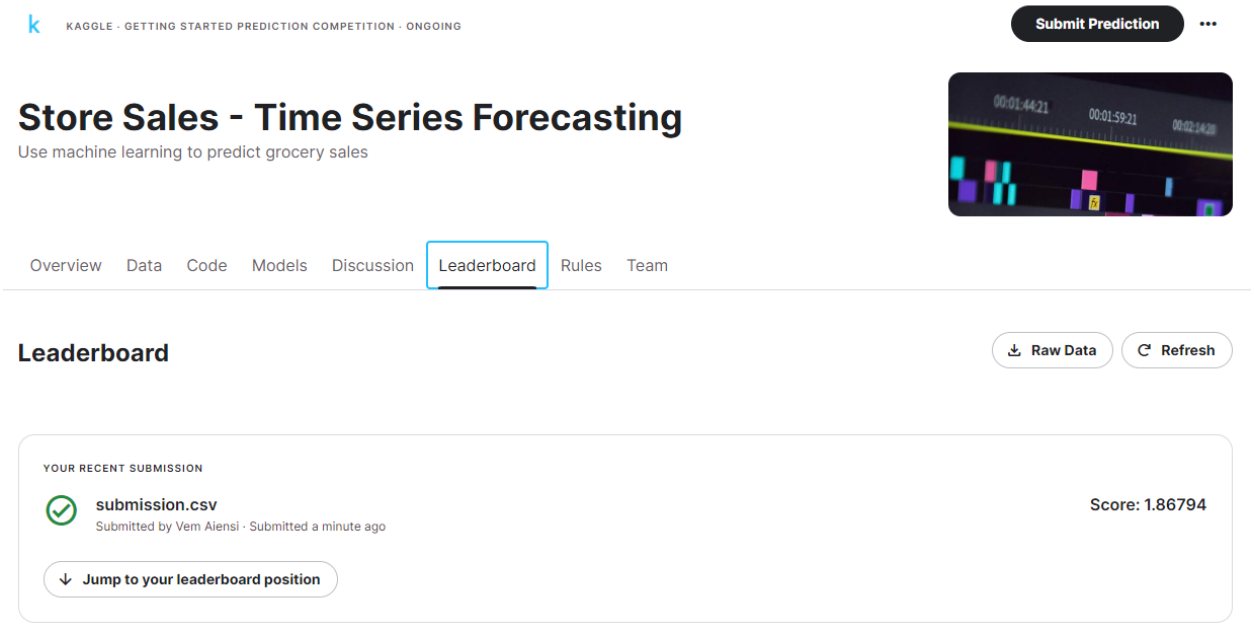
- Exploring Datasets
- Date and Time Conversion
- Solution with...
- Submission

CCSEL1-18 Professional ELective

[Store sales]

MARASIGAN, VEM AIENSI A.
3BSCS-I

Documentation and Leaderboard
January 2, 2024



Files for this challenge

https://github.com/VemAiensi/Professional-Elective-Course/tree/main/Kaggle-Competiton/Store-Sales

Other Competition

https://github.com/VemAiensi/Professional-Elective-Course/tree/main/Kaggle-Competiton