Data Mining Fianl Project

Utilize machine learning to predict grocery sales

Group 3 許捷翔、古森允、朱柏諺、黃 榆婷、陳品程

Agenda

1. Introduction
2. EDA
3. Data Preprocessing
4. Methodology
5. Result
6. Conclusion

1 INTRODUCTION

Objectives: utilize machine learning to predict sales of retail products

Experimental Background	Participate in a competition hosted by Kaggle, and utilize the dataset provided by Kaggle to employ different machine learning models for predicting the sales volume of retail products.
Experimental Propose	By forecasting sales volume, the product waste can be minimized, and customer satisfaction may also be enhanced by ensuring sufficient inventory levels.
Experimental Objectives	Use datasets with time series from Corporación Favorita, a large retailer in Ecuador. The aim is to utilize 4 and ⅔ years of data to predict the total sales volume of various product categories within each store over the next 16 days.
Evaluation Metric	Root Mean Squared Logarithmic Error (RMSLE)

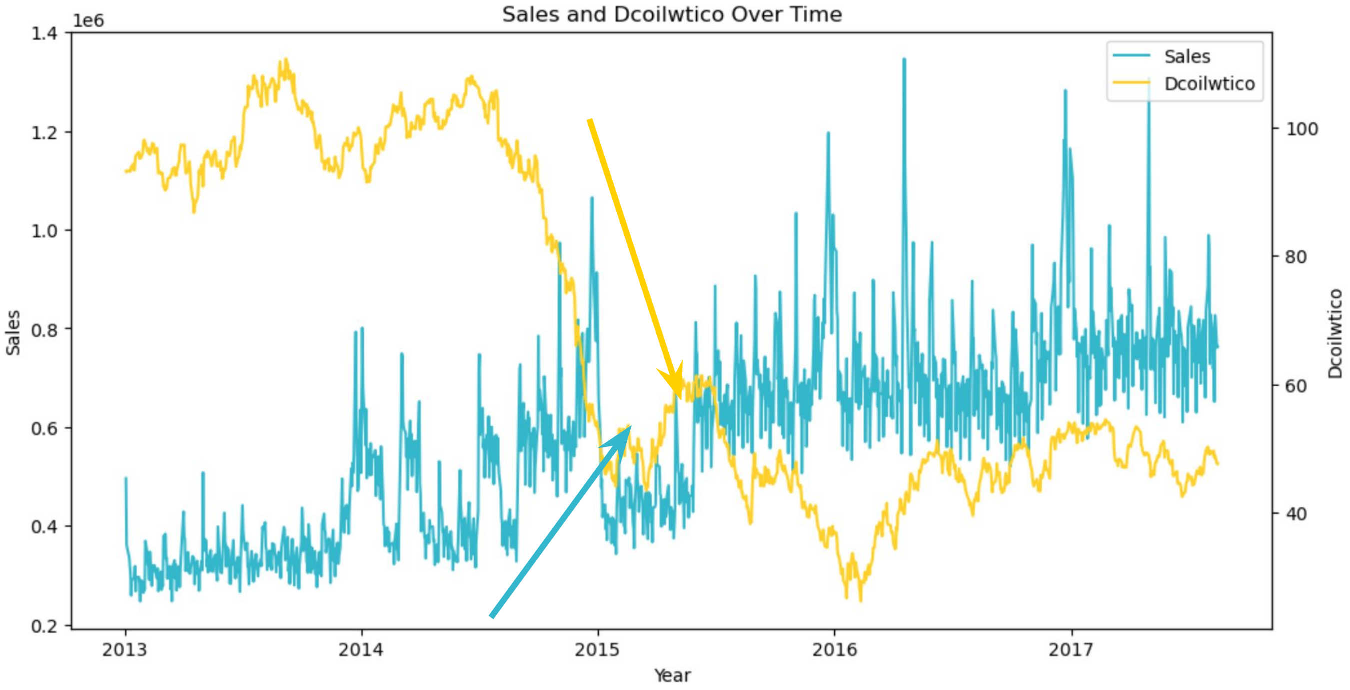
$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log(1+\hat{y}_i)-\log(1+y_i)\right)^2}$$

Datasets provided for predicting future sales

Train/Test	id / date (train: **2013/1/1 - 2017/8/15, test: 2017/8/16 - 2017/8/31**) / store number / product category / promotion quantity / sales (There's no sales column in testing data)
Stores	store number / city / state / type / store cluster
Transactions	date / store number / transactions
Oil	date / oil price
Holiday Events	date / type / locale / locale name / description / vacation transferred

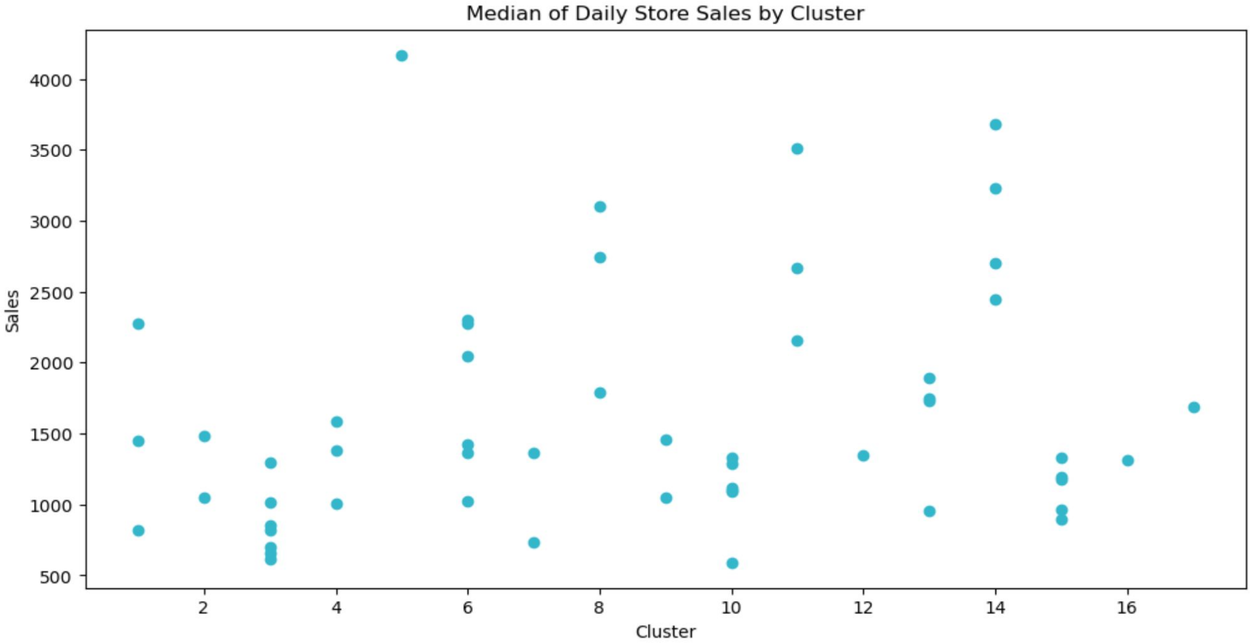
2 Explorational Data Analysis

The correlation between daily store sales and oil prices is highly negative



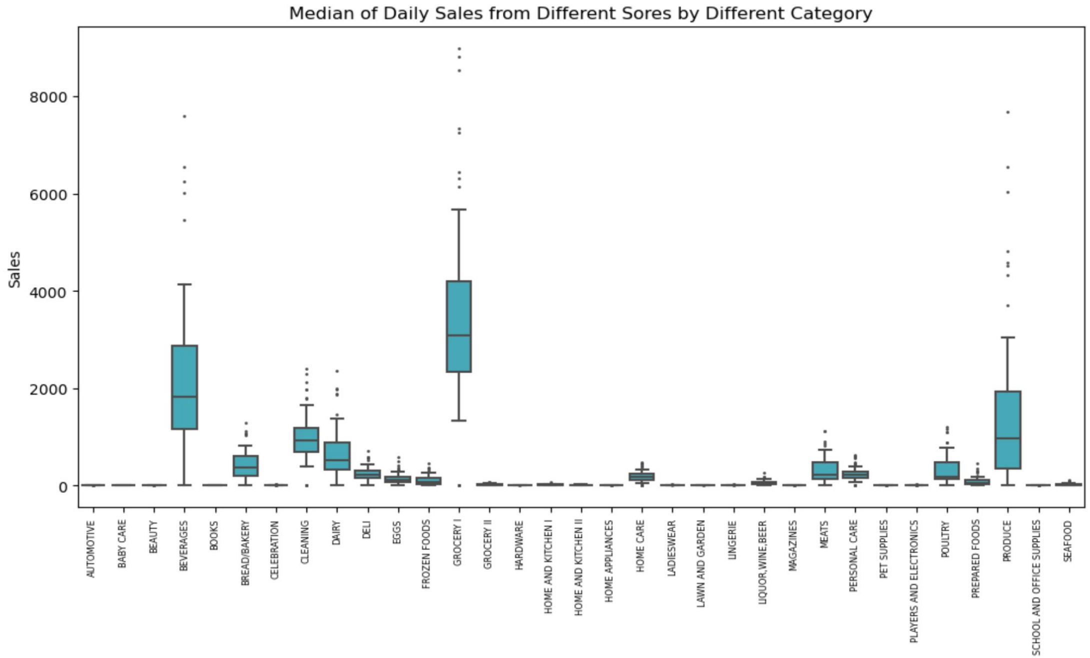
While oil prices have been falling since 2014, store sales have been increasing over time. The overall correlation coefficient during the data period is -0.71.

The store clusters fail to adequately represent the similarities in daily store sales



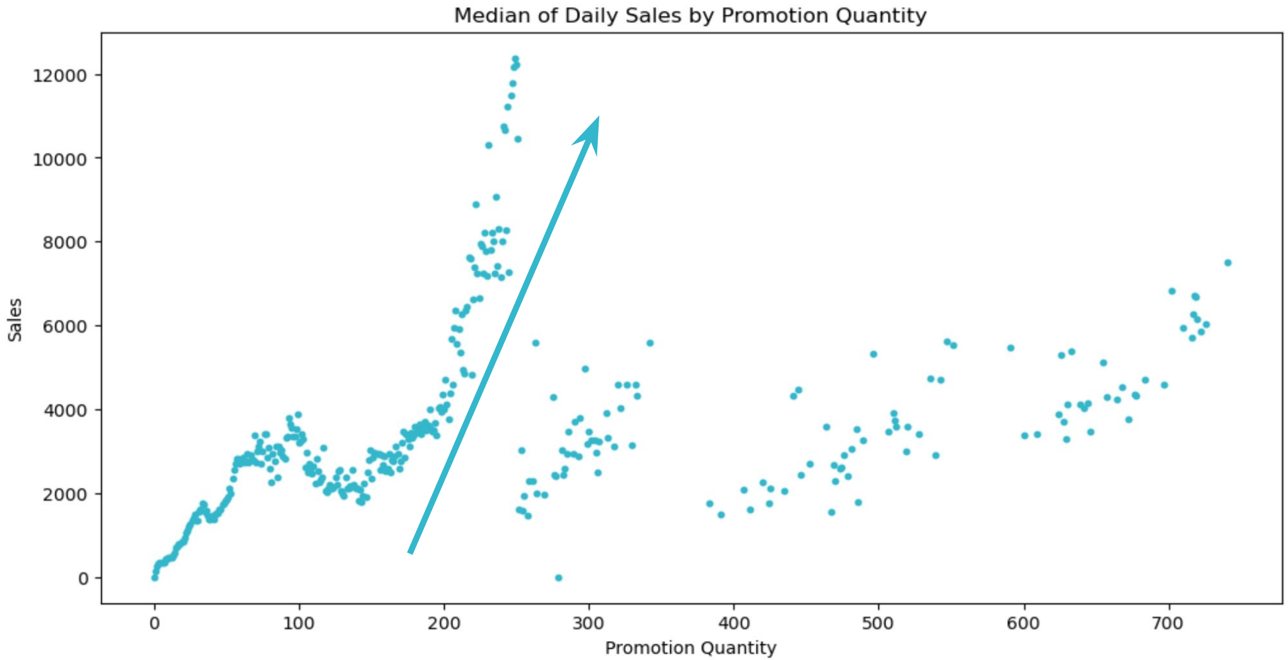
The dataset provides 17 clusters for 54 stores; however, it fails to capture similar daily store sales situations due to the wide distribution within some clusters.

Each category has its own daily sales distribution across different stores



There are 33 categories for the products, with some categories showing similar daily sales across different stores, while others exhibit wide variations in daily sales.

Promotions have a stimulating effect on daily store sales



The dataset provides the quantity of promotion items. Overall, there is a positive correlation between the quantity of promotion items and the median of daily store sales.

3 Data Preprocessing

Utilized a Data Scheme to Summarize the Relationship Between Datasets

Stores
store_nbr (foreign key)
store_type
cluster
……

Transactions
date (foreign key)
store_nbr (foreign key)
transactions

Training / Testing
id
date (foreign key)
store_nbr (foreign key)
city
……
sales (objective variable)

Holiday Events
date (foreign key)
city (foreign key)
event_type
……

Oil Prices
date (foreign key)
dcoilwtico

Conducted Data Reduction and Feature Engineering to Enhance Data Quality

training data

Data Reduction	Only included 2015/08/15~2017/08/15 data for the following study.
min-max normalization	Applied min-max normalization to “onpromotion”, “dcoilwtico” and “transactions” columns.
Date Information Extraction	Extracted the “year”, “month”, “day” and “day_of_week” columns from the “date” column.
Word Embedding	Utilized Bert to convert the “family” and “description” columns into word vectors.
One-Hot Encoding	Converted the “categorical” columns to One-Hot Encoding format.
Spatial Information Extraction	Mapped the “city” column to a geographic dictionary, getting the “longitude” and “latitude” columns.
Oil Price Imputation	On weekends or holidays, oil price information wasn't provided. To avoid inconsistency, we used interpolation to fill the missing value.

Conducted Data Reduction and Feature Engineering to Enhance Data Quality

testing data

min-max normalization	Applied min-max normalization to “onpromotion”, “dcoilwtico” and “transactions” columns.
Date Information Extraction	Extracted the “year”, “month”, “day”, “day_of_week”, “isHoliday” and “isEvent” columns from the “date” column.
Word Embedding	Utilized Bert to convert the “family” and “description” columns into word vectors.
Spatial Information Extraction	Mapped the “city” column to a geographic dictionary, getting the “longtitude” and “latitude” columns.
One-Hot Encoding	Converted the “categorical” columns to One-Hot Encoding format.
Transaction Data Imputation	The “transaction” column is only available in the training data; thus, averaging the data in the previous two years for supplementing.
Oil Price Imputation	On weekends or holidays, oil price information wasn't provided. To avoid inconsistency, we used interpolation to fill the missing value.

4 Methodology

Linear Regression Models

Multi-index Linear	• Treats each data point at a specific time as an independent observation. • Simple implementation and direct interpretation of coefficients. • May not capture non-linear relationships and complex interactions.
Moving Average	• Smooths out short-term fluctuations by averaging sales over the window. • Effective in reducing noise and identifying trends. • Choice of window size is crucial and may lag behind actual trends.
Exponential Smoothing	• Assigns exponentially decreasing weights to past observations. • More responsive to recent changes compared to Moving Average. • Adapts quickly to data trends and can capture seasonality.
ARIMA	• Combines autoregression, differencing, and moving average components. • Requires careful parameter selection and tuning. • Computationally intensive compared to simpler models.

Ensemble Models

Random Forest	Train multiple decision trees to make predictions, then average the predictions from these trees to obtain the final result.
XGBoost	Gradually reduce prediction errors through training, with each new model attempting to correct the errors of the previous model.
LightGBM	It is a highly efficient gradient boosting framework that uses tree-based learning algorithms, optimized for speed and memory usage.
CatBoost	It is designed to handle categorical features automatically without extensive preprocessing.

Multi-Index LSTM Models

Convert to Multi-Index Data

date	family	store_nbr	sales
12/27	a	1	2.10
12/28	a	1	0.00
12/27	a	2	36.6
12/27	b	3	200.4

Date	x1_a_1	x2_a_2	sales_a_1	sales_a_2
12/27	1.05	0.9	2.10	36.6
12/28	1.2	0.8	0.00	…
12/29	…	…	…	…
12/30	…	…	…	…

Use *N_Unique(family) * N_Unique(store_nbr) * N(original_variables)* of variables
To predict *N_Unique(family) * N_Unique(store_nbr)* of sales

Multi-Index LSTM Models

Train_X.shape = (100, 100)

Train_y.shape = (100, 10)

Create Sequences
Timesteps = 32

Train_X.shape = (68, 32, 100)

Train_y.shape = (68, 32, 10)

Train_X.shape = (100, 100)

extract last 31 rows

Test_X.shape = (16, 100)

Train_X.shape = (31, 100)

Create Sequences
Timesteps = 32

Test_X.shape = (16, 32, 100)

Multi-Index LSTM Models

```
def create_model(input_shape, lstm_units=1782, dropout_rate=0.2):

    input_layer = Input(shape=input_shape)

    # add multiple LSTM layers and dropout layers
    lstm_layer = LSTM(lstm_units, return_sequences=True)(input_layer)
    lstm_layer = Dropout(dropout_rate)(lstm_layer)
    lstm_layer = LSTM(lstm_units)(lstm_layer)
    lstm_layer = Dropout(dropout_rate)(lstm_layer)

    # add dense layer
    output_layer = Dense(1782, activation='relu')(lstm_layer)
    model = Model(inputs=input_layer, outputs=output_layer)

    # compile with MSLE
    model.compile(optimizer='adam', loss='mean_squared_logarithmic_error')
    return model
```

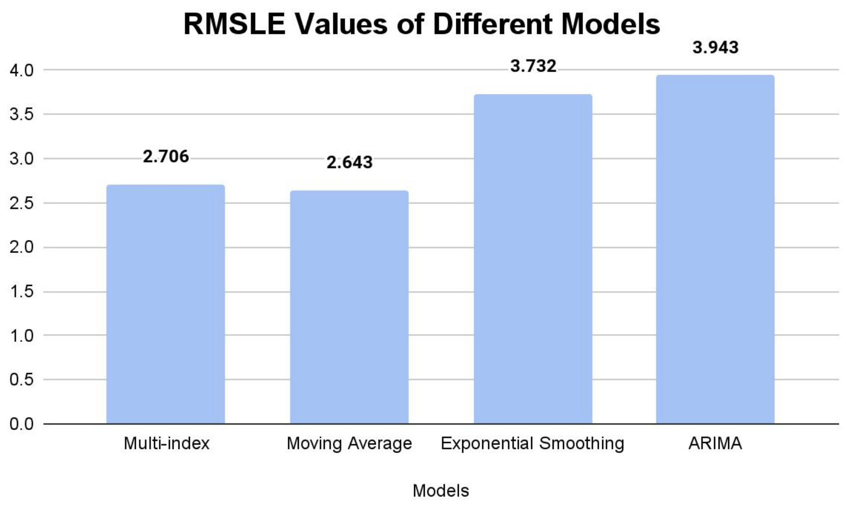
Multi-Index LSTM Models

Single-Index	Train *Unique(family)* * *Unique(store_nbr)* LSTM models to make predictions independently.
Multi-Index	Use the data preparation method mentioned previously, train one LSTM model.

5 Result

Linear Regression Models

We trained linear regression models using all preprocessed columns, and the results are as follows:



Among them, the performance of Moving Average is the best, but the results of the four models are not significantly different. Overall, the performance is not very good.

Linear Regression Models Result Interpretation

Non-linear Relationships

Linear models may struggle to capture complex non-linear relationships and interactions between features.

Computational Complexity

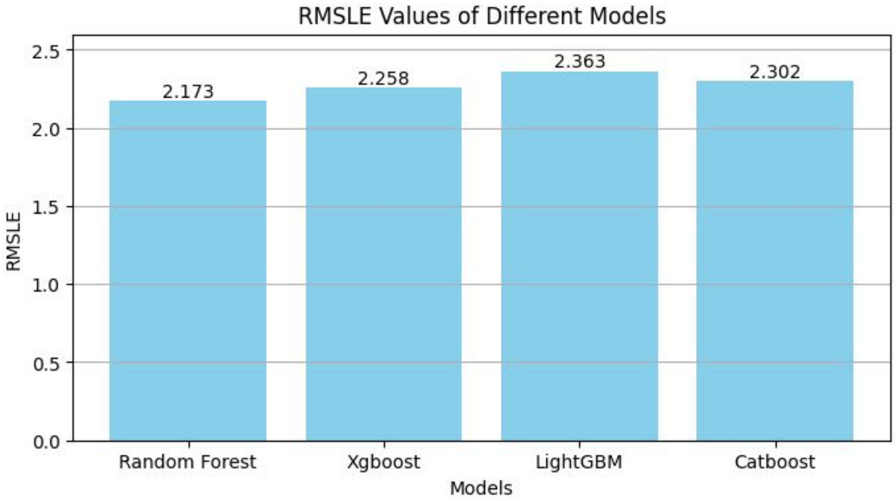
More complex models, such as ARIMA, can be computationally intensive and require significant resources for optimal performance.

Parameter Sensitivity

Models like Exponential Smoothing and ARIMA require careful parameter selection, which can be challenging and impact performance.

Ensemble Models

We trained ensemble models using all preprocessed columns, and the results are as follows:



Among them, the performance of Random Forest is the best, but the results of the four models are not significantly different. Overall, the performance is not very good.

Ensemble Models Result Interpretation

Lack of Adaptability to Temporal Patterns

Ensemble models may struggle to capture and adapt to complex temporal patterns, such as seasonality, trends, and cyclical variations.

Limited Incorporation of Time Dependencies

Ensemble models typically combine multiple base models independently without explicitly considering the time dependencies present in the data.

Difficulty in Handling Dynamic Changes

Time series data often involve dynamic changes over time. Ensemble models may have difficulty quickly adapting to these dynamic changes and updating their predictions accordingly.

LSTM Models

We trained LSTM models using all preprocessed columns. The results (public test score) are as follows:

Model	Single-Index LSTM	Multi-Index LSTM
RMSLE	2.51721	1.25308

The performance of Multi-Index LSTM is significantly better than Single-Index. Furthermore, it performs better than other models in this project, but it did not beat competitors on Kaggle.

Multi-Index LSTM Result Interpretation

Multi-Index is trained in a global scope

Comparing to Single-Index, the model get a peek of other families/stores'variables, which can better reflect the overall market condition.

Considered Time Dependencies

Comparing to other proposed models, creating time-sequence data to train in LSTM heavily focus on the time dependencies nature of the task.

Feature Extraction may be insufficient

Most method posted on Kaggle use time-series-based models, which is similar to this model. Relatively unideal performance may result from insufficient interpretable features.

6 Conclusion

Prediction Method Improvements

Extract Time-based Features	Since time-series model outperforms other models, it is highly possible that time-based features are more interpretable. More time- based features like inferring seasons by holidays should be extracted. Those features are relatively insufficient in our methods comparing to others.
Increase Computing Power & Memory	In the Multi-Index LSTM method, many features are dropped (including month, day, BERT related...etc) due to lack of sufficient memory while some of these features may be helpful. Alternatively, we can also investigate importance of each feature and optimize the result given limited memory or computing power.

Application

Inventory Optimization	By accurately predicting sales, supermarkets can better manage their inventory, avoiding both overstock and stockouts, which in turn reduces operational costs.
Food Waste Reduction	Accurate sales forecasting can help supermarkets avoid over-purchasing, thereby reducing food waste. This not only lowers costs but also contributes to environmental sustainability goals, enhancing the company's image of social responsibility.
Increased Customer Satisfaction	Ensuring popular items are not out of stock enhances the shopping experience and loyalty of customers. This results in higher customer retention rates and better reputation, ultimately translating into higher sales.

Future Work

Expansion to Other Stores and Product Categories

Based on the successful experience of the model, expand the prediction scope to more stores and different product categories.

Dynamic Adjustment and Optimization

Continuously monitor the predictive performance of the model and make adjustments and optimizations based on actual situations to maintain the model's efficiency and accuracy.

Technology Transfer and Training

Apply the successful experience and techniques from this competition to other data analysis and prediction projects within the company, enhancing the overall data analysis capabilities of the enterprise.

Thank You For Listening