Transaction Prediction System

Northwestern

Team 16: Junyi Chen, Fuqian Zou, Ananya Rattan Khurana

Problem Statement

Dillard's faces critical inventory and procurement challenges in today's competitive retail market, requiring a predictive model to optimize demand forecasting and drive data-driven decision making.

Predict Transaction Amount to Optimize Inventory and Enhance Operational Efficiency

- Determine the factors driving transaction amounts and their impact on sales performance.
- Analyze demand fluctuations across months, stores, and SKUs to identify trends and patterns.
- Assess the influence of department, brand, sale date, and pricing strategies on transaction behavior.
- Explore how SKU characteristics, store location, and historical demand contribute to monthly sales variability.

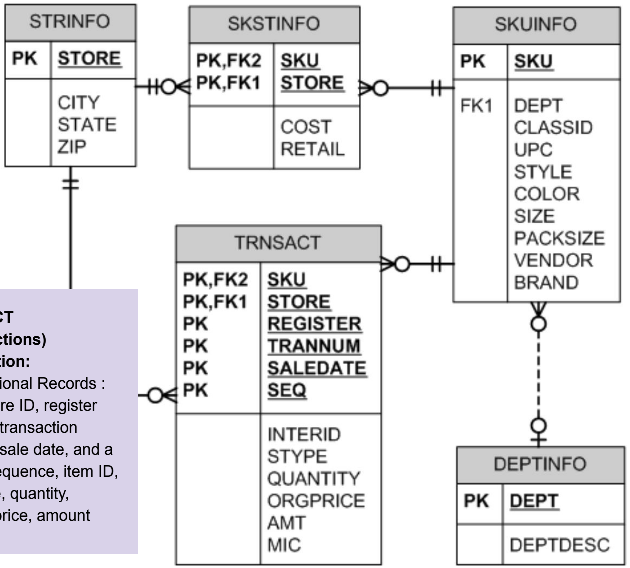
Entity Relationship

STRINFO (Store Information)
Description:
city, state, and zip code help identify the location details of each store.

SKSTINFO (Stock Store Information)
Description: Links products (SKU) to stores and contains cost and retail price of products in specific stores.

SKUINFO (Stock Keeping Unit Information)
Description: Product/SKU info like department, class ID, UPC, color, size, brand etc

PK – Primary key
FK – Foreign key



TRNSACT (Transactions)
Description:
Transactional Records : SKU, store ID, register number, transaction number, sale date, and a Order Sequence, item ID, sale type, quantity, original price, amount etc.

DEPTINFO (Department Information)
Description: Dept-Store wise info. This table is crucial for organizing products under various department categories.

Columns description (key columns)

Attribute	Description	Value Types
AMT	Total amount of the transaction charge to the customer	26.25, 44.00, ...
CITY	City where the store is located	ST. LOUIS, TAMPA, ...
COST	The cost of the stock item	9.00, 15.00, ...
DEPT	Department where the stock item belong	800, 801, 1100, ...
DEPTDESC	Description of the department	CLINIQUE, LESLIE, ...
ORGPRICE	Original price of the item stock	75.00, 44.00, ...
QUANTITY	Item quantity of the transaction	1, 2, 3, ...
RETAIL	The retail price of the stock item	19.75, 34.00, ...
SALEDATE	Sale date of the item stock	2005-01-20, 2005-06-02, ...
SKU	Stock Keeping Unit number of the stock item	4757355, 2128748, ...
SPRICE	Sale price of the item stock	26.25, 65.00, ...
STATE	State where the store is located	FL, MO, AR, ...
STORE	Store Number	2, 3, 4, 100, ...
STYPE	Type of the transaction (Return or Purchase)	P, R

Technical Architecture

pgadmin

Servers

linux+sql

python

VS Code

posit™

deptinfo.csv
skstinfo.csv
skuinfo.csv
strinfo.csv
trnsact.csv

ETL(extract, transform, and load)

deptinfo
skstinfo
skuinfo
strinfo
tmp
trnsact

Directly connect to database

Data quality check
Data cleaning

Data quality check
Data cleaning
EDA

Original data

Store cleaned data

Build Model for sales prediction

database

Create table

Include constraints and permissions based on the ER diagram.

Load data

1. deptinfo.csv→team16.deptinfo
2. strinfo.csv→team16.strinfo
3. skstinfo.csv→team16.skstinfo
4. skuinfo.csv→team16.tmp→team16.skuinfo
5. trnsact.csv→team16.trnsact

Load data sequentially according to their hierarchy.

Check counts

server

60 ./deptinfo.csv
39230146 ./skstinfo.csv
120916896 ./trnsact.csv
453 ./strinfo.csv
1564178 ./skuinfo.csv

Same counts

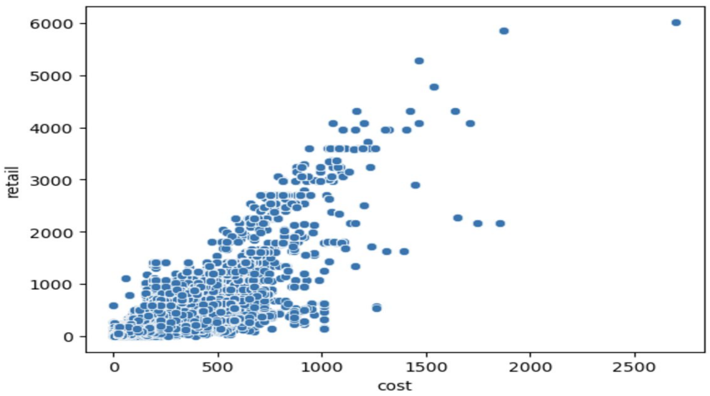
database

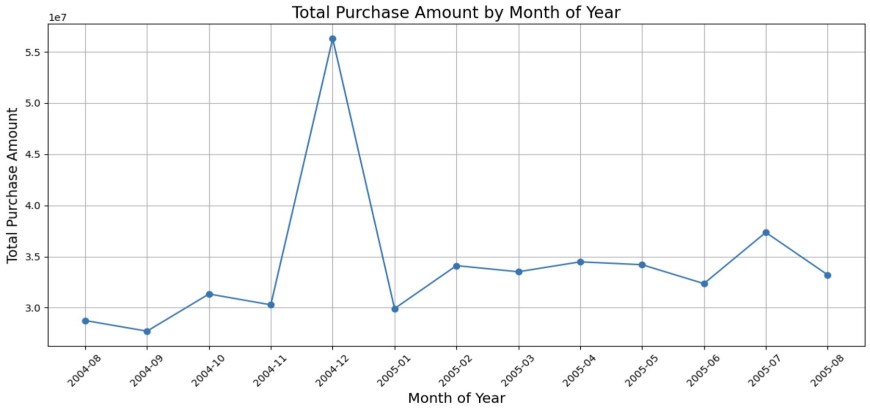
```
NOTICE:  Table: deptinfo, Rows: 60
NOTICE:  Table: skstinfo, Rows: 39230146
NOTICE:  Table: trnsact, Rows: 120916896
NOTICE:  Table: strinfo, Rows: 453
NOTICE:  Table: skuinfo, Rows: 1564178
NOTICE:  Table: tmp, Rows: 1564178
```

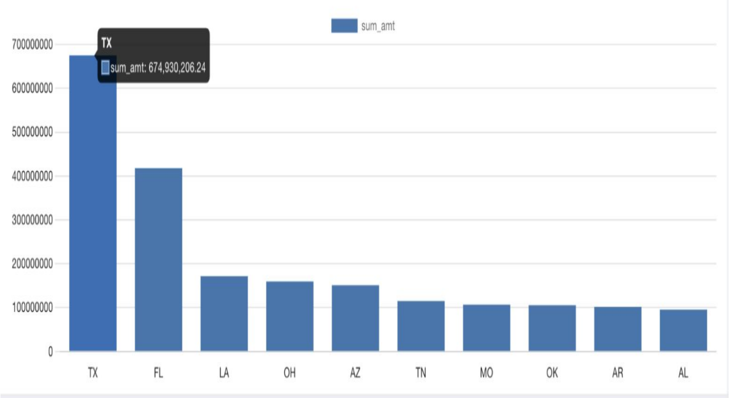
Exploratory Data Analysis

Insights:

1. Cost and retail price are positively correlated.
2. There are seasonal patterns in total sales. Total sales are highest in December.
3. Texas, Florida, Arkansas have the most number of stores.
4. Total sales in Texas and Florida is significantly higher than other states.







Data Cleaning

1. **Check data types:** Converted date to datetime type
2. **Correct typos and errors:** Removed blank space after string
3. **Check duplicate rows:** No duplicate rows in the dataset
4. **Check missing values:** Imputed missing values with mean/median
5. **Detect outliers:** Found outliers that are more than 3 standard deviations away from the mean and dropped/imputed outliers

	dept	deptdesc	bool
0	800	CLINIQUE ...	0
1	801	LESLIE ...	0
2	1100	GARY F ...	0
3	1107	JACQUES ...	0
4	1202	CABERN ...	0

→

	dept	deptdesc	bool
0	800	CLINIQUE	0
1	801	LESLIE	0
2	1100	GARY F	0
3	1107	JACQUES	0
4	1202	CABERN	0

```
sku         int64
store       int64
register    int64
trannum     int64
saledate    object
seq         int64
interid     int64
stype       object
```

→

```
sku         int64
store       int64
register    int64
trannum     int64
saledate    datetime64[ns]
seq         int64
interid     int64
stype       object
```

Feature Selection

Response	Numeric Predictors (Standardized)		Categorical Predictors (One-hot encoded)	
Monthly sales of sku in the store	Cost	The cost of the stock item	Month	Extracted month from saledate and dropped data in August 2025 to avoid double counting monthly sales in August
	Retail	The retail price of the stock item	Color_category	The color of the stock item. Most frequent colors were categorized to major types (black, white, red, blue, etc.)
	Packsize	The quantity of item per pack	State	State where the store is located

Model Training

Split the dataset into 80% for training and 20% for testing.

Model	Reason for Choice
Multiple Linear Regression	Simple and interpretable model for understanding the linear relationships between features and sales amount.
Ridge Regression	Add L2 regularization to prevent overfitting by penalizing large coefficients, especially when multicollinearity exists among predictors.
Lasso Regression	Add L1 regularization to encourage sparsity, making it useful for feature selection and simplifying the model.
LightGBM	Gradient-boosting model optimized for speed and efficiency, suitable for handling large datasets and capturing non-linear relationships.
XGBoost	Another gradient-boosting model known for its high performance and advanced regularization techniques, effective in handling non-linear and complex datasets.

Model Evaluation

Model	R-squared	Mean Squared Error
Multiple Linear Regression	0.631	292.905
Ridge Regression	0.631	292.905
Lasso Regression	0.631	293.055
LightGBM	0.758	192.572
XGBoost	0.753	195.986

Best model

ROI- Driving 27,400.20% Returns at Dillard's

Cost Reduction

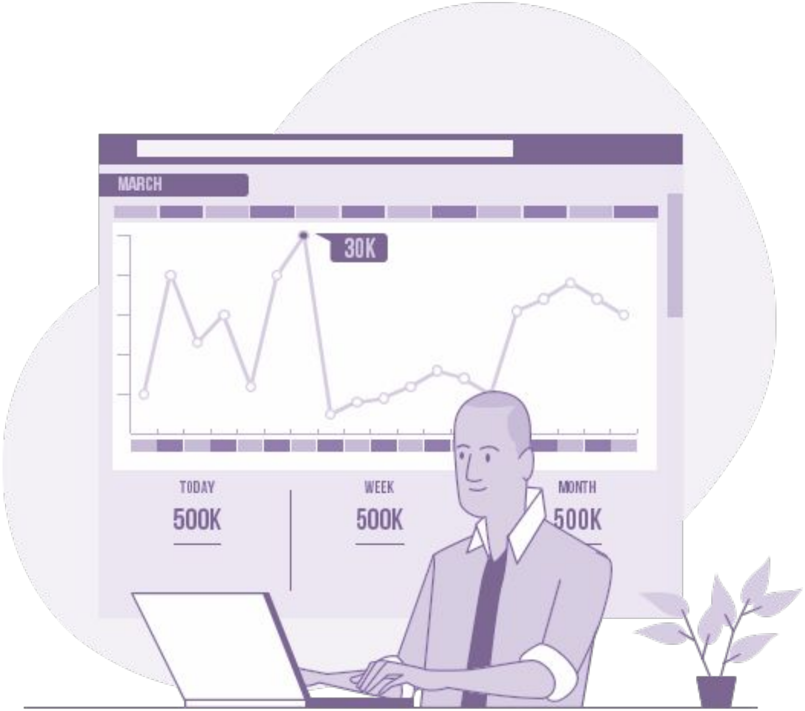
- Achieved a **50% reduction** in overstock and stockout rates, resulting in annual savings of **$102.36M** and **$51.18M**, respectively.
- Lowered storage costs and minimized markdown risks.
- Optimized working capital by aligning inventory with demand.

Revenue Growth

- Recovered **$153.54M annually** by preventing revenue losses due to overstock and stockout scenarios.
- Enabled strategic promotions and peak period optimization. Enhanced cross-selling with accurate SKU-level alignment.

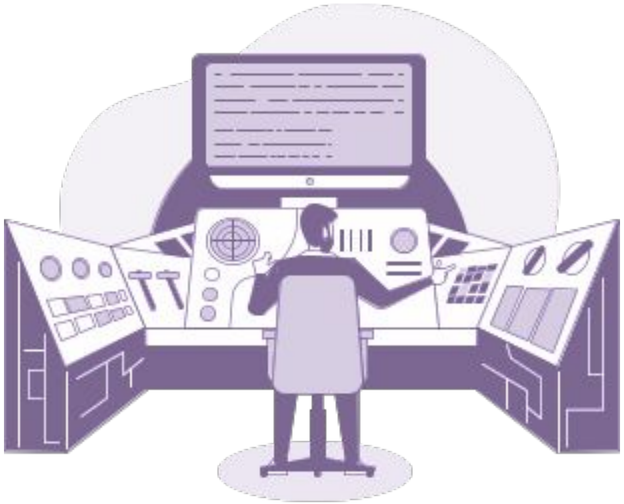
Operational Efficiency

- Automated forecasting with **LightGBM (R^2=0.7575)**, streamlining procurement.
- Improved supply chain and vendor coordination for cost-effective operations.



Model Application and Recommendation

- **Adjust Inventory**: Allocate inventory dynamically, ensuring high-demand SKUs are stocked in sufficient quantities while limiting low-demand SKU purchases.
- **Target Promotions**:
  - **For low-demand SKUs:** Launch store-specific promotions, bundle offers, or discounts to stimulate sales. Encourage personalized campaigns
  - **For high-demand SKUs:** Optimize pricing strategies to maximize revenue without discouraging purchases.
- **Real-Time Updates**: Integrate the model with real-time sales data to make quick adjustments in restocking and promotional efforts.



Strategy Examples–Target Promotions

Low-Demand SKUs

Scenario: A store identifies through the model that SKU Z (a high-end kitchen appliance) has consistently low sales.

Action Plan:

- **Bundle Offer:** Combine SKU Z with a popular SKU (e.g., a cooking pan) at a discounted price to encourage sales.
- **Localized Discounts:** Offer 20% off SKU Z in stores with historically low sales for kitchen appliances.
- **Marketing Campaign:** Create targeted advertisements for SKU Z, emphasizing its unique features and limited-time pricing.

Outcome: Increased sales for SKU Z while creating perceived value for customers through bundling and discounts.

High-Demand SKUs

Scenario: SKU A (a best selling smartphone) is predicted to sell out during the holiday season.

Action Plan:

- **Dynamic Pricing:** Slightly increase SKU A's price during peak demand periods while staying competitive to maximize margins.
- **Stock Prioritization:** Ensure adequate stock in high-performing stores based on historical sales data.
- **Upselling:** Offer complementary accessories (e.g., phone cases, chargers) as add-ons with SKU A purchases.

Outcome: Optimized profit margins and improved customer experience through the availability of both the main product and complementary items.

Strategy Examples–Real Time Updates

Scenario:
During a promotional weekend, real-time sales data indicates that SKU B (a popular snack) is selling out faster than anticipated in urban stores.

Action Plan:

- **Dynamic Restocking:** Use the sales model to prioritize SKU B's restocking in urban stores from nearby distribution centers.
- **Adjust Promotions:** Temporarily pause promotional discounts on SKU B in urban stores to slow demand and maintain stock.
- **Redistribution:** Transfer excess SKU B stock from suburban stores, where demand is lower, to urban stores.

Outcome: Prevented stockouts in high-demand areas and reduced inventory waste in low-demand areas.

Conclusion

The sales prediction model delivers a data-driven solution for optimizing inventory and enhancing retail operations. It enables precise demand forecasting at the SKU level, helping businesses:

- **Minimize overstock** and stockouts to reduce costs.
- Tailor promotions and pricing strategies to **boost revenue**.
- Streamline operations with **automated workflows** and scalable processes.
- Make informed, **real-time decisions** based on actionable insights.

Appendix

Github: https://github.com/NUMLDS/MLDS-400-2024-Team16