CS 4641 Project Final Report

1. Background

Predicting sales is extremely important for any business. Predicting food sales is especially important considering the shelf-life of most grocery store items. In the United States, it is estimated that 31% of food resources are lost at the retail and consumer levels [1].

Dataset

The dataset we chose was collected using grocery store scanner data from Circanna from 2019 - 2023. The dataset has around 90,000 rows, with each row showing sales data corresponding to a week of sales of a product in each state [2].

This link

Literature Review

To get started we reviewed some existing literature. In a research paper from the Balkan Journal of ECE, sales forecasting techniques were compared. The authors summarize, "Our experiments show that the regression techniques provide higher performance and accuracy compared to the time series analysis techniques." In their research, they evaluated three separate metrics: Mean Absolute Error, Root Mean Squared Error, and the Coefficient of Determination. Introduction to these results and metrics has been helpful for our project [3]. We also researched a second paper that discusses how Artificial Neural Networks could be useful [4].

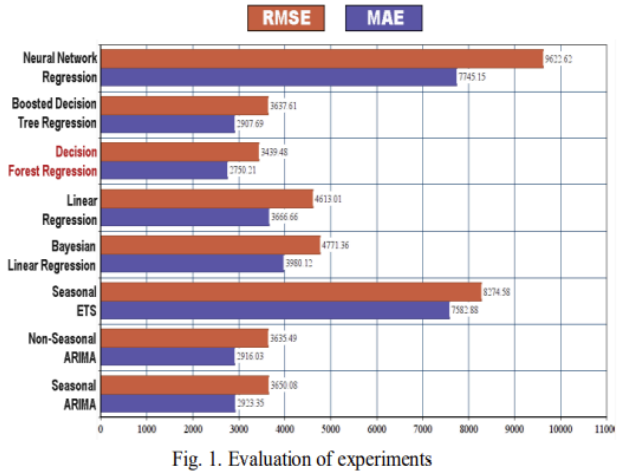


Fig. 1. Evaluation of experiments

Figure taken from [3].

2. Problem Definition

Predicting product sales not only leads to financial growth, but it can also greatly improve overall supply chain efficiency. Suppose businesses want to know how much of a product will be sold in future weeks. To solve this question we hope to estimate forecasted sales utilizing various machine learning techniques and algorithms.

3. Methods

Preprocessing Methods

Our three main preprocessing methods will include data cleaning, working with imbalanced data, and feature engineering. We will need to properly remove null or missing values or fix incorrect inputs (e.g. if product ids are missing/wrong). There is also the potential to be working with imbalanced data and finding out how to work around this when using our algorithms. Finally, we will most likely be employing feature engineering in order to get our data in a suitable format for our times series model by grouping product sales together for certain time periods.

Algorithms/Models

The Machine Learning Algorithms we will employ in this project are K-means clustering, Hierarchical clustering, time series regression, and neural network. The two unsupervised methods will help organize the data into specific product categories for us to try and learn what similar metrics these categories share. The time series regression model will help us try and predict future sales based off of the historical sales data.

4. (Potential) Results and Discussion

Metrics

We plan to use the following metrics for our regression model:

- **Max Error**: To measure how far off our prediction is in the worst case.
- **Root Mean Squared Error (RMSE)**: To evaluate the overall accuracy of the model.
- **Explained Variance**: To assess how much variance in sales can be explained by our model.

We also plan to use the **completeness score** for our potential clustering algorithm to group products together.

Project Goals

1. **Regression Model**: To predict future sales using past sales data.
2. **Clustering Model**: To group products in a category together.

Sustainability and Ethical Considerations

- **Sustainability**: Predicting sales can help minimize wastage and overproduction.
- **Ethical Considerations**: Our dataset is from the US Department of Agriculture and is publicly available to everyone.

Expected Results

For the regression model, we expect:

1. **Minimizing Max Error**: Keep the max error within an acceptable range to show good performance even in the worst-case scenario.
2. **Low RMSE**: Achieve a low root mean squared error to demonstrate strong prediction accuracy.
3. **High Explained Variance**: Attain a high explained variance score, indicating that the variance in predicted sales can be explained by the model.

For the clustering model, we expect:

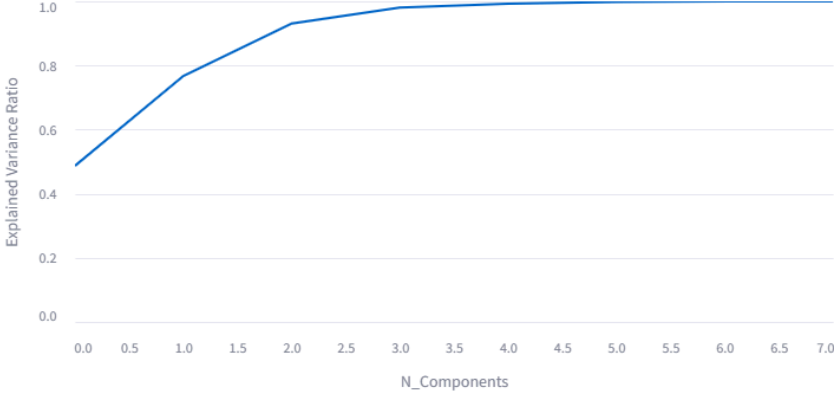
- A **high completeness score** showing that data points belonging to the same class are assigned to the same cluster.

5. (Midterm) Results and Discussion

Preprocessing:

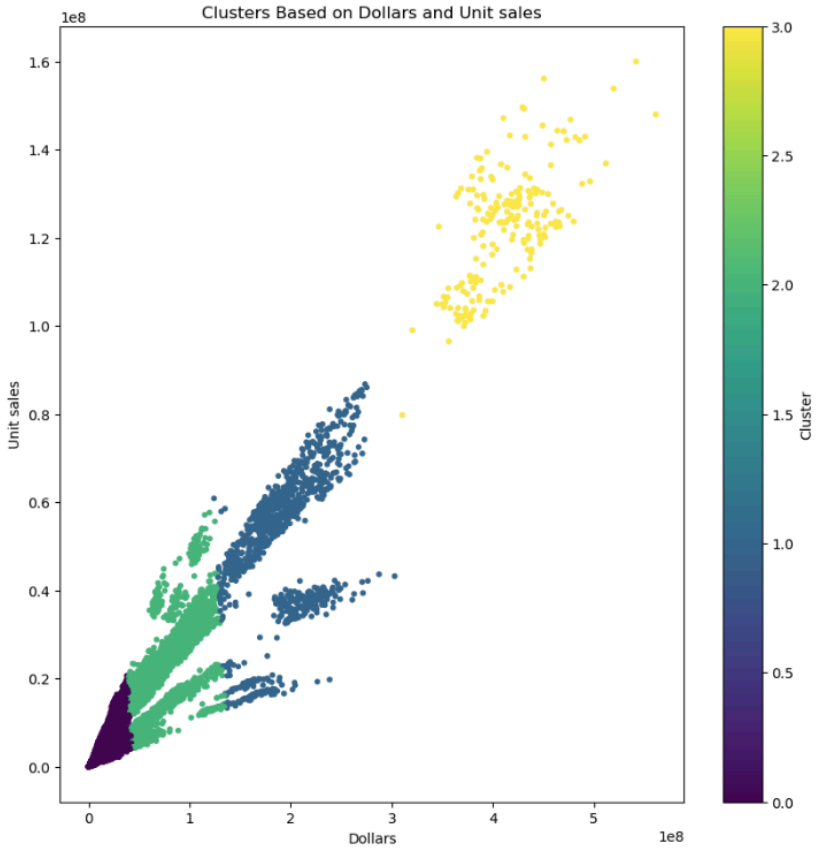
The data was initially cleaned by removing all of the NaN values. Then, using the sklearn.preprocessing package, the data was standardized with a StandardScaler object to fit and transform the desired features. With the goal of keeping the features that retained almost all the variance, we plotted an elbow chart with the number of features vs. the retained variance. We then kept the features that retained the variance above our specified variance threshold.

Elbow Chart of Retained Variance



The data was also split into training and testing data with a split ratio of .8. We also split the data by state and category as well as formatted the time columns in order to prepare the data for use in our linear regression model.

KMeans



Silhouette Score:

: `.6924774999962319`

Analysis:

The graph values are on a 10^8 scale. Each cluster represents a group of data points (items) with similar "Dollars" and "Unit sales" characteristics. The representation of the clusters can be seen as those with a relatively similar value for unit and dollar sales are in the same group.

The legend on the right shows that:

- **Purple** represents low dollar and unit sales.
- **Yellow** represents high dollar and unit sales.
- **Green and Blue** represent values between purple and yellow.

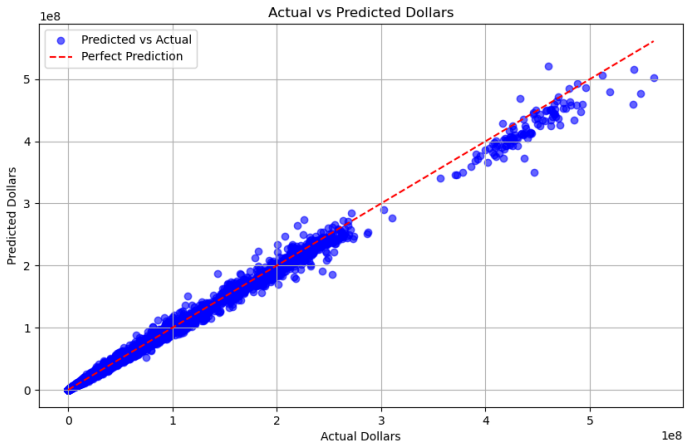
The silhouette score of `.6924774999962319` suggests that the clusters are relatively well defined.

Insights:

This clustering could help companies categorize products into different sales performance levels, potentially allowing for targeted inventory or marketing strategies. For example:

- **High-revenue items** could be prioritized for stock.
- **Low-performing items** might be candidates for promotions or discontinuation.

Linear Regression



MSE:

: `1.5488×10^13`

RMSE:

: `3.9355×10^6`

Analysis:

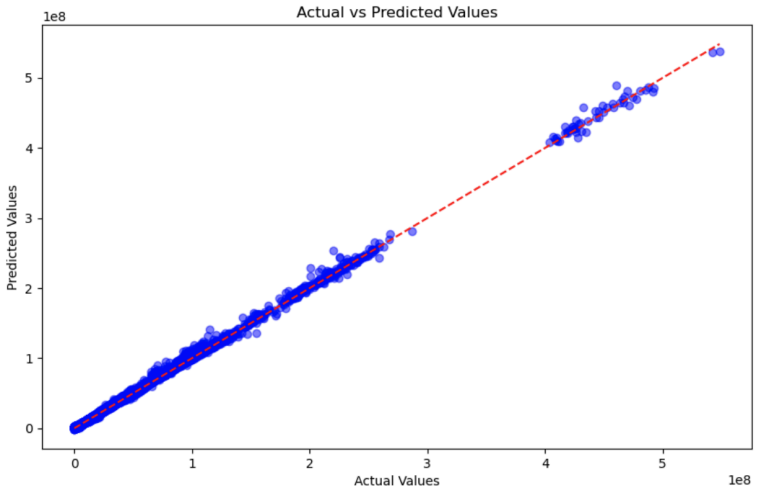
A pipeline was used for this method which includes both preprocessing and regression. To preprocess the data for linear regression, first the rows with an invalid date are dropped from the dataset. The data is also additionally sorted by date from earliest-latest, however, this is unnecessary since the goal is to predict the Dollars column in the current year based on Dollars last year,Unit sales, State and Category. Using a ColumnTransformer, a preprocessor is created for the pipeline. One Hot Encoding is used for State and Category to convert them into numerical values suitable for computation. The numerical columns, which are Dollars Last year and Unit Sales are passed through as is since these are already in numerical format.

An 80-20 test split is done on the data using array slicing.

The algorithm then indexes into the dataset to create X_train, y_train, x_test and y_test arrays.

At this point, the pipeline is ready to be implemented. Sklearn's LinearRegession() library is used as the regressor. The MSE and RMSE are calculated using sklearn.metrics' mean_squared_error library.

###Comparision to the initial attempt The first attempt at Linear Regression included all the remaining columns as features except Dollars, which proved ineffective, since it is basically cheating the algorithm since two columns gave percent changes for 1 year and 3 years ago. The updated model had a similar RMSE to the inital flawed model, which proves that these four features are enough to bring the optimal results. Below is the visualization for the initial attempt.



Initial MSE:

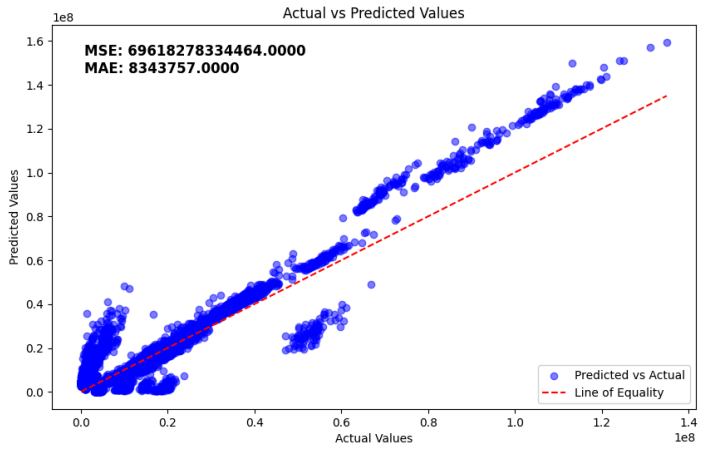
: `4.57×10^12`

Initial RMSE:

: `6.76×10^7`

Overall, the model captures the trend well but there are some errors of precision as noted by the RMSE. Comparing this to the K-Means algorithm, this was much more effective to predict future sales.

Neural Network



MSE:

: `6.9618×10^13`

RMSE:

: `8.3437×10^6`

Analysis:

Preprocessing done is similar to regression.

An 80-20 test split is done on the data using array slicing and it is not randomly split as we want to use the data from the past to predict the future.

The algorithm then indexes into the dataset to create X_train, y_train, x_test and y_test arrays.

The neural network has 4 layers in total: 1 input layer (128 neurons), 2 hidden layers (64 and 32 neurons) and 1 output layer. The activation functions used was relu for all layers except the output layer. The MSE and RMSE values are calculated by running the model after training on the test dataset.

Conclusion:

Overall, the model captures the trend well but there are some errors of precision as noted by the RMSE. Comparing this to the K-Means algorithm, this was much more effective to predict future sales. While comparing this to the Linear Regression algorithm, the linear regression seems to be a better fit given the error values.

Next Steps

The next steps of this project are to utilize more algorithms to analyze and compare the results with each other. Other algorithms such as hierarchical clustering and random forest can be used to further investigate the data. In addition, we will also utilize more metrics to understand the data better such as max error and explained variance. We are also considering using feature selection or transformation to reduce dimensionality and noise in the dataset. For the clustered data, we are considering using it to individually find regressions for more catered results or findings by cluster. Another way we are thinking about making the regressions more relevant is by doing regressions for each state and category within the data.

5. References

[1] Why should we care about food waste? (n.d.). USDA. https://www.usda.gov/foodlossandwaste/why

[2] USDA ERS - Weekly Retail Food sales. (n.d.). https://www.ers.usda.gov/data-products/weekly-retail-food-sales/

[3] Catal, C., Ece, K., Arslan, B., & Akbulut, A. (n.d.). Benchmarking of Regression Algorithms and Time Series Analysis Techniques for Sales Forecasting. Balkan Journal of Electrical and Computer Engineering, 7(1), 20–26. https://doi.org/10.17694/bajece.494920

[4] Lu, C.-J., Lee, T.-S., & Lian, C.-M. (2012). Sales forecasting for computer wholesalers: A comparison of multivariate adaptive regression splines and artificial neural networks. Decision Support Systems, 54(1), 584–596. doi:10.1016/j.dss.2012.08.006

Contribution Table and Gantt Chart

Aaron	Create Github page, start intro, reformat contribution table. Midterm: Created Linear Regression model, visualized actual vs predicted sales and calculated MSE
Connor	Worked on Streamlit formatting, background (literature review primarily), and the problem definition. Midterm: Created the KMeans model with Dollars and Unit Sales as features. Also created scatter plot and calculated silhouette score to test clustering method.
Soumyadeep	Worked on metrics, goals and expected results as part of the results and discussion. Midterm: Worked on preprocessing data: split data by state and category, split it into train and test data (80-20), converted dates to integer and states to categorical values with one hot encoding. Organized code into different files. Final: implemented neural network to predict sales, cretaed visualizations, and evaluated it against other methods.
Devin Wade	Worked on preprocessing methods and ML algorithms/models. Midterm: Worked on preprocessing methods, specifically cleaning the data and performing Principal Component Analysis.
Tom	Worked on Video Presentation and the Results and Discussion for the Midterm Report

Link to Gantt Chart

GANTT CHART

PROJECT TITLE	Sales Prediction																
					PHASE ONE												
					Sep 27							Oct 4					
TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	M	T	W	R	F	S	U	M	T	W	R	F	
Project Proposal																	
Introduction & Background	Aaron, Connor	9/27/2024	10/4/2024	7													
Problem Definition	Aaron, Connor	9/27/2024	10/4/2024	7													
Methods	Devin	9/27/2024	10/4/2024	7													
Potential Results & Discussion	Soumyadeep	9/27/2024	10/4/2024	7													
Video Recording	Tom	10/4/2024	10/7/2024	3													
GitHub Page	Aaron	10/4/2024	10/7/2024	3													

Gantt Chart screenshot.