Product Performance Analysis

Jonathan Garcia, Orestes Lorda, Alessandro Squadritto

6338496, 6406166, 6429219

Florida International University

Introduction:

Our project is a flask web application that sets out to analyze products sales data with machine learning algorithms. These machine learning algorithms are implemented through a python backend in flask. The web app analyzes and preprocesses the data given to it and uses algorithms like k-means clustering to group data and regression models to predict product performance. This is all achieved with a user-friendly interface that allows anyone to use and understand the application

Data Preprocessing:

The project helps clean and prepare the dataset using a structured preprocessing line. Essentially, missing values are handled by dropping any rows with missing entries, while gaps that remain are filled potentially using imputations for numerics. Outliers in numeric features are addressed using IQR (Interquartile Range) in which extreme values are capped at the lower and upper bounds, this keeps rows while reducing the impact of values that can be larger or smaller than usual. Numeric features are standardized using z-score normalization which makes sure they are consistent, this is important for potentially distance based algorithms.

K-means Clustering Analysis:

The product grouping in this project was done with an implementation of k-means clustering developed in the python backend. This implementation of K-means Clustering consists of starting with random centroids (the number of them is specified by the user) then from there iteratively finding the closest points to those centroids with Euclidean distance, then after that recalculating the centroids based on the mean of all the points inside the clusters and then repeating this process until convergence happens and the centroids don't change position

anymore. The program also calculates an elbow method graph and allows the user to perform themselves the elbow method, with the program suggesting the optimal k value. The webapp also shows a cluster scatter graph that portrays the price vs units sold. Lastly, after finishing the clustering algorithm the program shows the user each cluster's number of products, average price, average units sold and average profit. Each cluster is named, given characteristics that describe it and business insight based on its calculated values and attributes.

Regression Analysis:

The regression analysis has two models to predict the business outcomes; Linear Regression and Polynomial Regression. We chose linear regression as the baseline because of its efficiency and how easy it is to interpret, this makes it super ideal to understand the direct relationship between features such as price, cost, and units sold. We used polynomial regression to capture the non-linear relationships as well as feature interactions that could exist in product sales data. For the training process we used a 70-30 train-test split with a fixed random state. In addition, we also have a performance comparison table that displays training and test MSE values alongside MAE for model evaluation, with the best model being automatically selected based on the lowest test MSE. To assess the reliability, we included an overfitting analysis by calculating test-to-train MSE rations, from which a ratio below 1.5 shows healthy generalization while higher values suggest overfitting concerns. We acknowledge key tradeoffs such as: linear regression offering simplicity but may miss complex patterns, while polynomial regression captures non-linearity at the cost of more complexity and higher overfitting risk. The model comparison is presented through tables and tradeoff analysis cards that show the pros and cons of each approach, computational cost, and flexibility to name a few.

Visualizations:

Our analysis has five key visualizations throughout the results to provide insights into both clustering and regression performance. The elbow curve plots k against WCSS to identify the optimal number of clusters, where the "elbow" shows the ideal k value. The scatter plot visualizes products in 2-D using price versus units sold and has points color coded by cluster assignment. For regression evaluation, the actual vs predicted plot shows actual test set values against model predictions for the better performing model with a red diagonal line that represents the perfect prediction; having the points cluster tightly around the line means high accuracy. The residual plot compliments all of this by showing the predicted values against residuals, in which models should display random scatter around the zero line with no patterns, since any curves or clustering could mean underfitting. These visualizations give a full picture of the model's performance.

Conclusion:

Throughout the project, we have explored and analyzed many parts going form K-means Clustering which dealt with implementing the algorithm of K-means along with Regression Analysis. Throughout this part of the course there was definitely a lot to learn regarding the clustering, learning about elbow curves along with the visualization of the clusters. With this homework we were using AI tools to help us with coding as well as debugging errors faced in our path while coding and helping with the implementation of regression as well as K-means clustering.

All in all, the homework helped with learning more about K-Mean, Visualization and Regression along with MSE/MAE, the concepts of artificial intelligence, using Github to share progress of the project with one another as well as a good amount more diving into the world of artificial intelligence/machine learning.