Predicting Product Sales from Product Features

For project 2 the goal is to find out whether product information can be used to predict sales. For that purpose, the research question was:

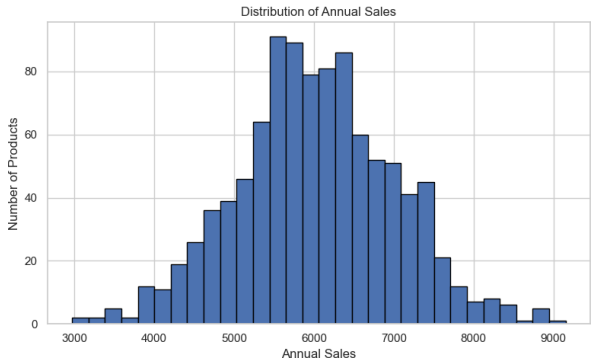
Can product characteristics such as price, category, review score, and review count predict the number of sales, and what factors are associated with increased sales?

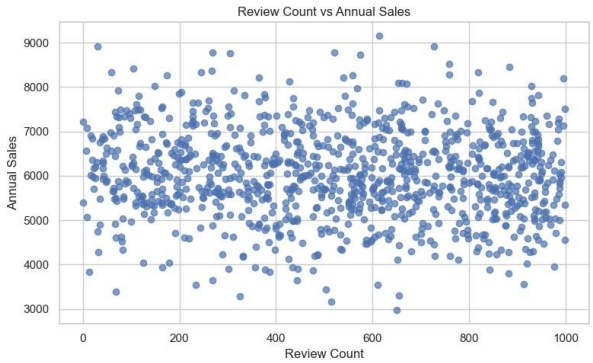
I wanted to make the question more business oriented than last time and came to the conclusion that businesses of course want to understand what makes products sell better. If certain factors are strongly related to sales, a company could use that information to make better decisions about pricing, advertising, etc. Beyond just creating machine learning models, would the available variables in the dataset actually contain useful information for predicting sales.

The dataset contains information on around **1,000 products**. Each row represents one product, and the columns include product details along with monthly sales data. This is actually, pretty small compared to the last dataset. The main variables used for this project were:

1. category
2. price
3. review_score
4. review_count
5. sales_month_1 through sales_month_12
6. annual_score
7. avg_monthly_score

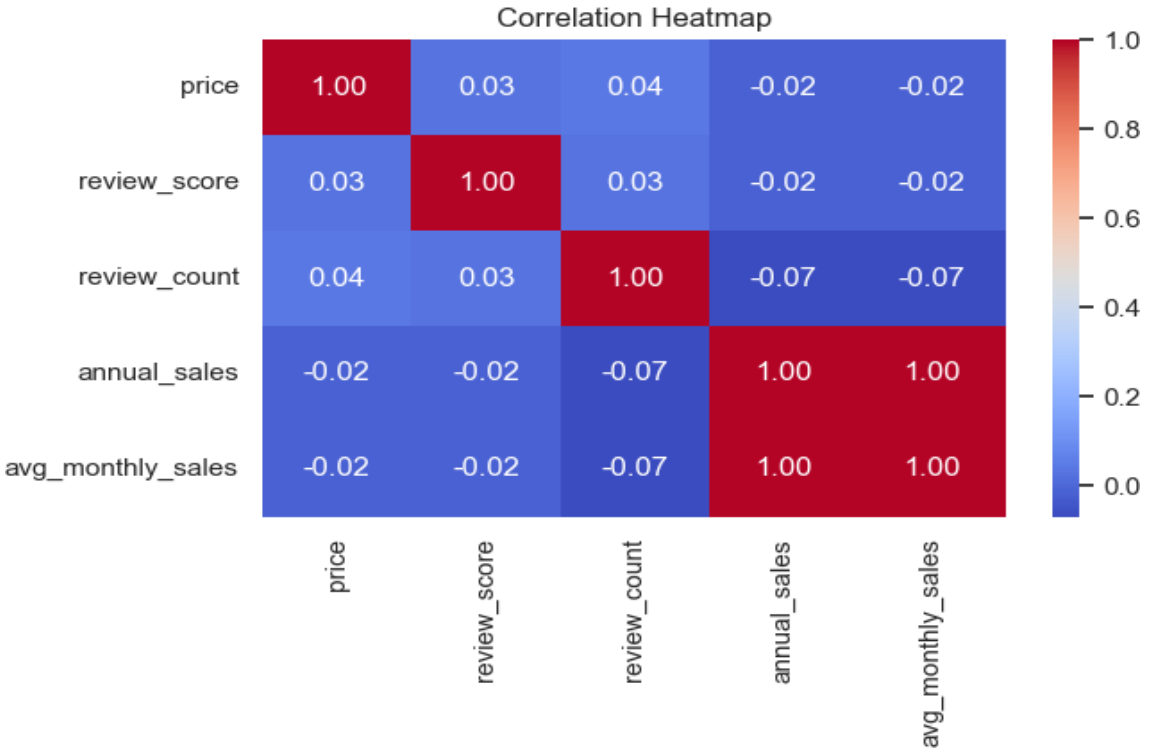
First, I loaded the dataset from the zip file and inspected the structure of the data. I checked the column names, data types, shape of the dataset, and whether any values were missing. Just like last time (I suppose this is true for a lot of Kaggle datasets), there were no duplicates and the data was already really clean. I used summary statistics to get another rough outline of the data.





After that, I selected my variables and began to do exploratory data analysis to better understand the dataset looking at the distribution of annual_sales. After checking for correlation I created a distribution and a dot plot. This helped show whether sales were spread evenly or whether some products sold much more than others. The distribution suggested a wide range of yearly sales value.

Next, I calculated correlations between the numeric variables: price, review_score, review_count, annual_sales, and avg_monthly_sales.



Heatmaps like the one above are a staple because they show strong linear relationships, however the correlations between these predictors and annual sales were weak. No strong relationships means there might not be a correlation for the machine learning model to learn from correctly. Predictors might not explain sales very well. With this in mind I continued towards preprocessing, preparing my data for machine learning.

Machine Learning:

This project used **three regression models** because the target variable, annual_sales, is numeric.

1. Linear Regression

Linear regression was used as a baseline model. It is simple, interpretable, and useful for checking whether there is a basic linear relationship between the predictors and annual sales.

2. Random Forest Regressor

Random forest is a tree-based ensemble model. I chose it because it can capture nonlinear relationships and interactions between variables that linear models might miss.

3. Gradient Boosting Regressor

Gradient boosting is another ensemble model that often performs well on structured datasets. I included it to test whether a more flexible model could detect patterns that the simpler models could not.

When evaluating I split the dataset into training and testing sets. The models were trained on the training set and evaluated on the test set. This is important because it tests how well the model works on unseen data rather than only on the data it learned from. To evaluate performance I used **MAE (Mean Absolute Error)** which is the measure of the average absolute difference between the true sales and the predicted sales, **RMSE (Root Mean Squared Error):** which is basically prediction error, but it penalizes larger errors more heavily than MAE, and **R^2 (R-squared),** which measured how much of the variation in the target variable is explained by the model. Higher values are better. A value near 1 means strong predictive power, while a value near 0 means weak predictive power. A negative value means the model performs worse than simply predicting the mean.

7. Model Results

The model comparison results were:

- **Linear Regression:**
  MAE = 777.93
  RMSE = 956.90
  R^2 = -0.0068
- **Random Forest Regressor:**
  MAE = 830.72
  RMSE = 1038.51
  R^2 = -0.1858
- **Gradient Boosting Regressor:**
  MAE = 845.32
  RMSE = 1048.37
  R^2 = -0.2084

The best model was **Linear Regression**, but its R^2 value was still slightly below 0. Unfortunately even the best model did not explain annual sales well and ultimately this means that **the available features were poor predictors of annual sales**. That is the answer to our question based on our results. It's not what I expected. I believed that it made sense that products with better reviews and more review counts would have higher sales but the model's failure suggests that these variables alone do not capture enough information. This could be because Important variables are missing, relationships were weak from the beginning (likely), the dataset was too small (models had too limited information to learn from), etc. The negative R^2 values are especially important because they show that the models are not just imperfect; they are failing to capture a useful predictive signal from the available inputs. With **ethical considerations** in mind If a model performs poorly as in this case the honest conclusion is that the current variables are not enough for accurate sales prediction. **Despite that**, a negative result can still be meaningful in data science, explaining that in order to have better predictions we may require better data rather than just more complex models.