**PRODUCT SALES ANALYSIS**

DAC\_PHASE 4 Submission Document Part 2

P.Rajavel[812121106026]

Sales analysis is a comprehensive task that involves various activities. Here's a breakdown of the key steps involved.

**Data Collection:**

Gather relevant sales data, which may include transaction records, customer information, product details, and market trends.

**Data Preprocessing:**

Clean the data by handling missing values, outliers, and duplicates. This is also the stage where you perform feature engineering to create new features that might be useful for analysis.

**Exploratory Data Analysis (EDA):**

Visualize and explore the data to gain insights into sales trends, patterns, and relationships. This step can involve generating charts, histograms, and summary statistics.

**Feature Engineering:**

Create new features or transform existing ones to improve the model's performance. This can involve techniques like one-hot encoding, scaling, and creating interaction terms.

**Model Selection:**

Choose the appropriate machine learning or statistical model for sales prediction. Common models include linear regression, decision trees, random forests, or more advanced models like neural networks.

**Model Training:**

Train the selected model on your dataset using techniques such as cross-validation to ensure it generalizes well to new data.

**Model Evaluation:**

Assess the model's performance using relevant evaluation metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE). You might also use techniques like k-fold cross-validation for a more robust assessment.

**Hyperparameter Tuning:**

Optimize the model's hyperparameters to improve its predictive accuracy. This may involve grid search or random search.

**Model Deployment:**

If the model meets the desired performance criteria, deploy it to make predictions on new sales data. This could be integrated into a business intelligence tool, a web application, or another suitable platform.

**Monitoring and Maintenance:**

Continuously monitor the model's performance in a production environment and update it as needed.

**Reporting and Visualization:**

Create reports and visualizations to communicate the results and insights from the analysis to stakeholders. Tools like Tableau or Power BI can be helpful for this purpose.

Each of these steps is essential in conducting a thorough sales analysis that can lead to better decision-making and improved business outcomes.

**Discretization**

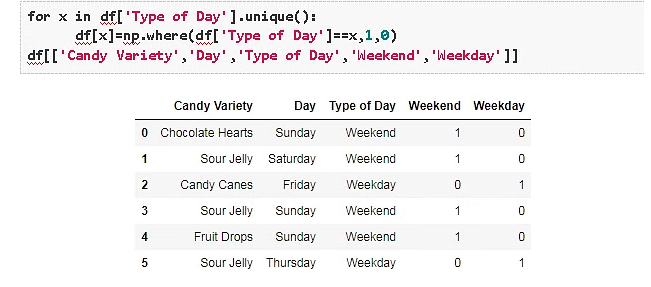
Discretization involves taking a set of data values and grouping sets of them together logically into bins (or buckets). Binning can apply to numerical values as well as to categorical data values. This could help prevent data from overfitting but comes at the cost of loss of granularity of data. The grouping of data can be done as follows:

* Grouping of equal intervals
* Grouping based on equal frequencies (of observations in the bin)
* Grouping based on decision tree sorting (to establish a relationship with target)



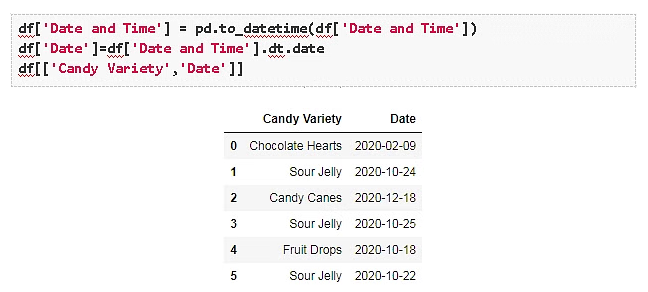
### ****Categorical Encoding****

Categorical encoding is the technique used to encode categorical features into numerical values, which are usually simpler for an algorithm to understand. One hot encoding(OHE)  is a popularly used technique of categorical encoding. Here, categorical values are converted into simple numerical 1’s and 0’s without losing information. As with other techniques, OHE has disadvantages and must be used sparingly. It could dramatically increase the number of features and result in highly correlated features.



### ****Feature Splitting****

Splitting features into parts can sometimes improve the value of the features toward the target to be learned. For instance, in this case, Date better contributes to the target function than Date and Time.



### ****Handling Outliers****

Outliers are unusually high or low values in the dataset, which are unlikely to occur in normal scenarios. Since these outliers could adversely affect your prediction, they must be handled appropriately. The various methods of handling outliers include:

* Removal: The records containing outliers are removed from the distribution. However, the presence of outliers over multiple variables could result in losing out on a large portion of the datasheet with this method.
* Replacing values: The outliers could alternatively bed treated as missing values and replaced by using appropriate imputation.
* Capping: Capping the maximum and minimum values and replacing them with an arbitrary value or a value from a variable distribution.

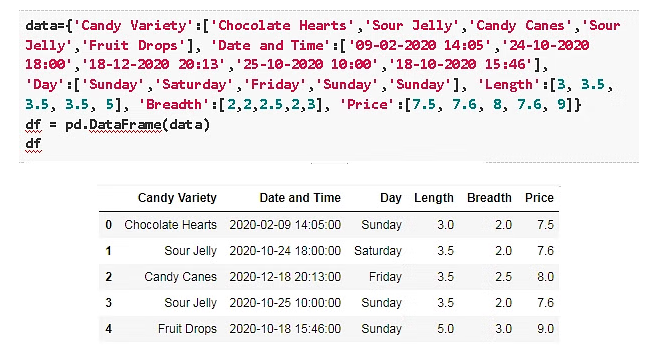
### ****Scaling****

Feature scaling is done owing to the sensitivity of some[machine learning algorithms](https://www.projectpro.io/article/7-types-of-classification-algorithms-in-machine-learning/435) to the scale of the input values. This technique of feature scaling is sometimes referred to as feature normalization. The commonly used scaling processes include:

* **Min-Max Scaling-** This process involves rescaling all values in a feature from 0 to 1. In other words, the minimum value in the original range will take 0, the maximum value will take 1, and the rest of the values between the two extremes will be appropriately scaled.
* **Standardization/Variance Scaling-** All the data points are subtracted by their mean, and the result is divided by the distribution's variance to arrive at a distribution with a 0 mean and variance of 1.

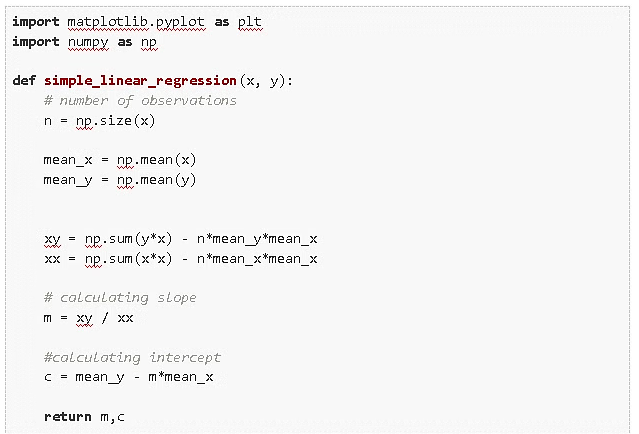
It is necessary to be cautious when scaling sparse data using the above two techniques as it could result in additional computational load.

Let’s consider a simple price prediction problem for our candy sales data –



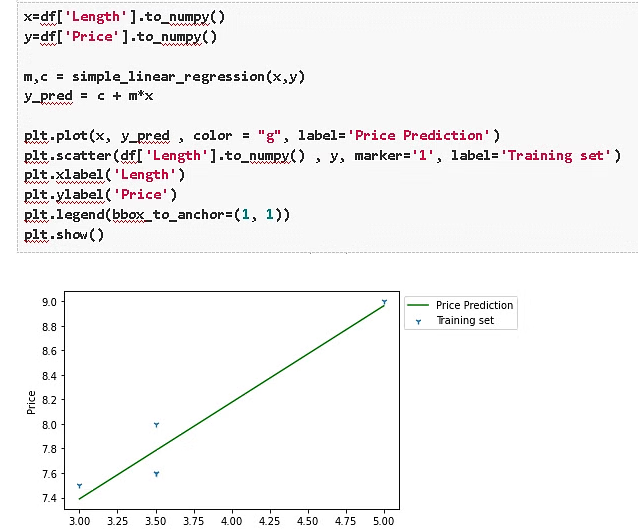
We will employ a basic linear[regression](https://www.projectpro.io/article/types-of-regression-analysis-in-machine-learning/410/) model to predict the price of various candies and learn how to implement Python ML feature engineering.

Let us start by building a function to calculate the coefficients using the standard formula for calculating the slope and intercept for our simple[linear regression model](https://www.projectpro.io/article/machine-learning-regression-projects-ideas/501).



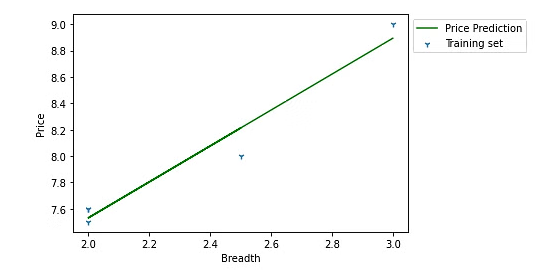
Now we build our initial model without any Feature Engineering, by trying to relate one of the given features to our target. From observing all the variables in the given data we know that it is most likely that the Length or the Breadth of the candy is most likely related to the price.

Let us start by trying to relate the length of the candy with the price.



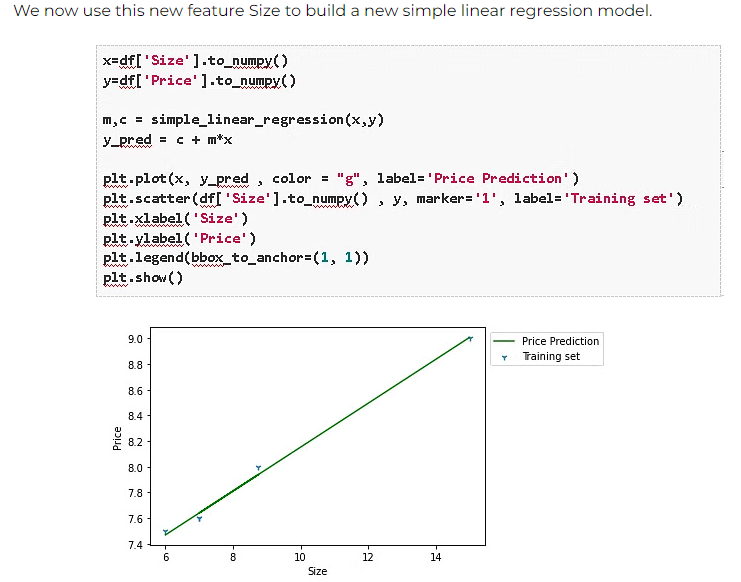
We observe from the figure that Length does not have a linear relation with the price.

We attempt a similar prediction with the Breadth to get a similar outcome. (You can execute this by replacing ‘Length by ‘Breadth in the above code block.)



Finally, it’s time to apply our newly gained knowledge of Feature Engineering in Python! Instead of using just the given features, we use the Length and Breadth feature to derive a new feature called Size which (you might have already guessed) should have a much more monotonic relation with the Price of candy than the two features it was derived from.





If you thought that the previous predictions with the Length(or Breadth) feature were not too disappointing, you would agree that the results with the Size feature are quite spectacular!

We have demonstrated with this example that by simply multiplying the Length and Breadth features of a pack of candy, you can achieve Price predictions well beyond what you would with the much less efficient relationship of Prices to Length (or Breadth). However, when working with real-life data, Feature Engineering could be the difference between a simple model that works perfectly well and a complex model that doesn’t.