## **Feature Engineering Techniques For Machine Learning -How to do Feature Engineering?**

While understanding the training data and the targeted problem is an indispensable part of Feature Engineering in machine learning, and there are indeed no hard and fast rules as to how it is to be achieved, the following feature engineering techniques for machine learning are a must know for all data scientists-

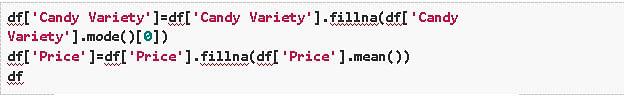
### **1. Imputation**

Imputation deals with handling missing values in data. While deleting records missing specific values is one way of dealing with this issue, it could also mean losing out on valuable data. This is where imputation can help. It can be classified into two types-

* Categorical Imputation: Missing categorical variables are generally replaced by the most commonly occurring value in other records
* Numerical Imputation: Missing numerical values are generally replaced by the mean of the corresponding value in other records









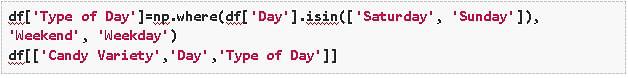
Notice how the technique of imputation given above corresponds with the principle of normal distribution (where the values in the distribution are more likely to occur closer to the mean rather than the edges), which results in a relatively good estimate of missing data. Other ways to do this include replacing missing values by picking the value from a normal distribution with the mean/standard deviation of the corresponding existing values or even replacing the missing value with an arbitrary value.

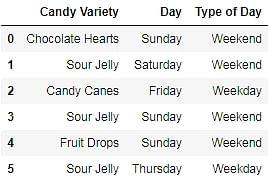
However, one must be reasonably cautious when using this technique because retention of data size with this technique could come at the cost of deterioration of data quality. For example, in the above candy problem, you were given 5 records instead of one with the ‘Candy Variety’ missing. Using the above technique, you would predict the missing values as ‘Sour Jelly,’ possibly predicting the high sales of Sour Jellies all through the year! Therefore, it is wise to filter out records with greater than a certain number of missing data or critical values missing and apply your discretion depending on the size and quality of the data you are working with.

### **2. 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)

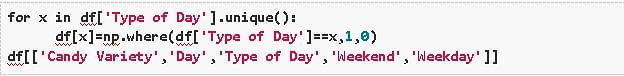


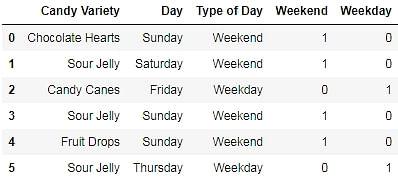


**Recommended Projects to Learn Feature Engineering for Machine Learning**

### **3. 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.



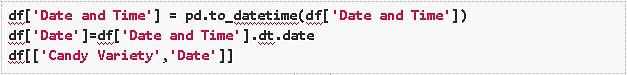


Besides OHE there are other methods of categorical encodings, such as-

* Count and Frequency encoding- captures each label's representation,
* Mean encoding -establishes the relationship with the target, and
* Ordinal encoding- the number assigned to each unique label.

### **4. 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.





### **5. 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.

### **6. Variable Transformations**

Variable transformation techniques help with normalizing skewed data. One such popularly used transformation is the logarithmic transformation. Logarithmic transformations operate to compress the larger numbers and relatively expand the smaller numbers. This, in turn, results in less skewed values, especially in the case of heavy-tailed distributions. Other variable transformations used include Square root and Box-Cox transformations, which generalize the former two.

### **7. 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.

### **8. Feature Creation in Machine Learning**

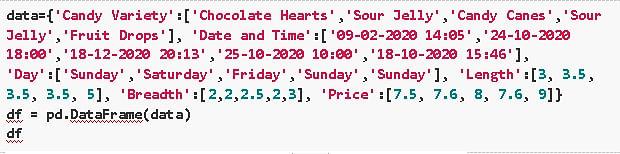
Feature creation involves deriving new features from existing ones. This can be done by simple mathematical operations such as aggregations to obtain the mean, median, mode, sum, or difference and even product of two values. Although derived directly from the given input data, these features can impact the performance when carefully chosen to relate to the target (as demonstrated later!)

While the techniques for feature creation in machine learning listed above are by no means a comprehensive list of techniques, they are popularly used and should help you get started with feature engineering in machine learning.

## **Feature Engineering Python-A Sweet Takeaway!**

We have gone over ML Feature Engineering, some commonly used feature engineering techniques in machine learning projects, and their impact on our machine learning model’s performance. But why just take someone’s word for it?

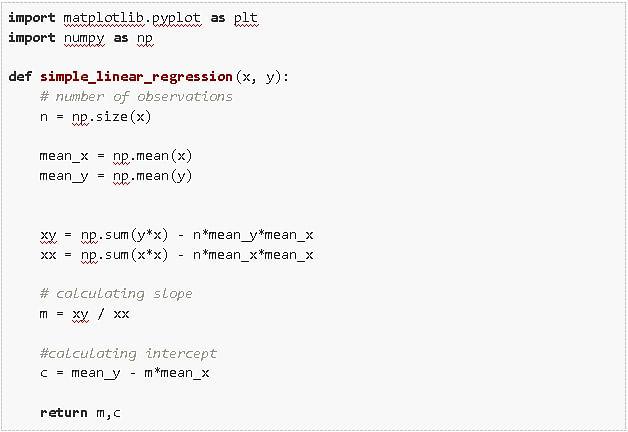
Let’s consider a simple price prediction problem for our candy sales data –





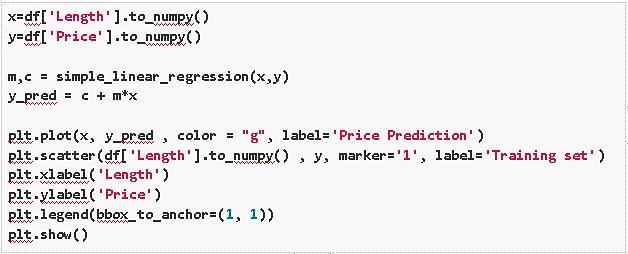
We will employ a basic linear [regression](https://www.projectpro.io/article/types-of-regression-analysis-in-machine-learning/410/) model to predict the price of various candies and learn how to implement Python ML feature engineering.

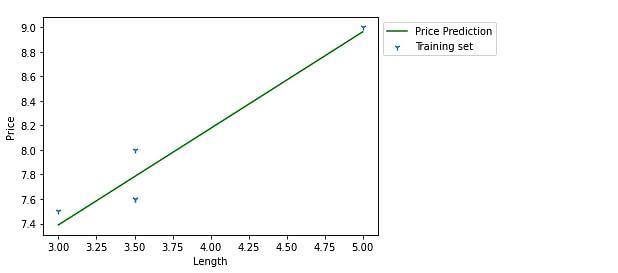
Let us start by building a function to calculate the coefficients using the standard formula for calculating the slope and intercept for our simple [linear regression model](https://www.projectpro.io/article/machine-learning-regression-projects-ideas/501).



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.

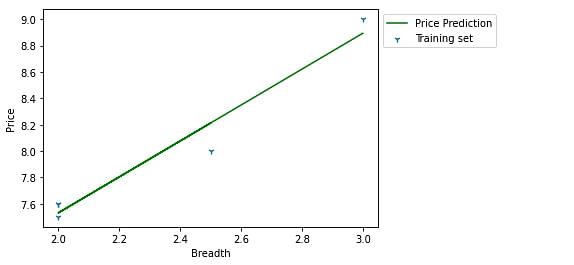
Let us start by trying to relate the length of the candy with the price.





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.)

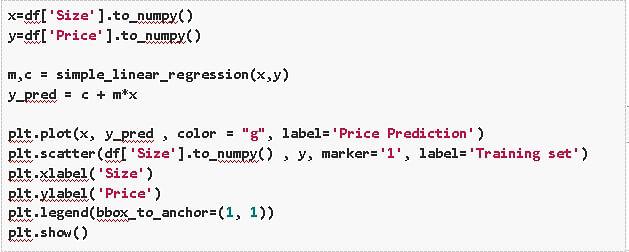


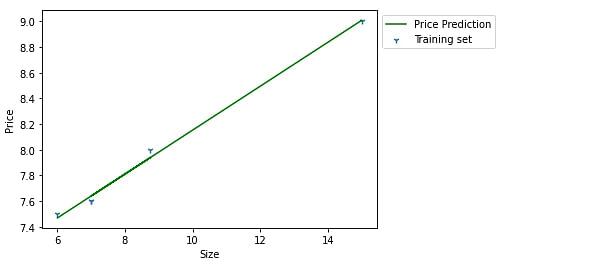
Finally, it’s time to apply our newly gained knowledge of Feature Engineering in Python! Instead of using just the given features, we use the Length and Breadth feature to derive a new feature called Size which (you might have already guessed) should have a much more monotonic relation with the Price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

Image for Feature Engineering Code



We now use this new feature Size to build a new simple linear regression model.





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.

## **Master Feature Engineering Techniques in Machine Learning With ProjectPro**

Don't settle for average-performing machine learning models when you have the power of feature engineering at your fingertips. The famous physicist Richard Feynman once said, "What I cannot create, I do not understand." So, take the leap from theory to practice by engaging in real-world data science and ML projects offered by [ProjectPro](https://www.projectpro.io/?utm_source=Blog423&utm_medium=HomePage). Gain hands-on experience in implementing feature engineering techniques and witness firsthand the magic they bring when building effective ML models. By delving into these projects, you will discover how to wrangle and transform your data, from creating new features to selecting the most relevant ones. *Remember*- it's not just about the algorithms; it's about the artistry of the feature engineering process that sets individuals apart as data scientists or ML professionals. Master the art of feature engineering by exploring the ProjectPro repository, and let your data science journey shine brighter than ever before.