Tejasvini Mavuleti

09/01/2022

Final report - Python project

**Manufacturing cost regression using Python for analyzing the economics of scale**

**Project background**

In regression analysis, fixed and variable costs produce the most accurate estimate. If there are no unexpected data points in the data set, it is crucial to check it first, possibly in the form of a scatter graph, and to ensure there are no outliers. The high-low method and regression analysis are the primary cost estimation techniques used to determine the amounts of fixed and variable expenses. To forecast and plan for the future, managers must separate mixed expenses into their fixed and variable components.

The amounts, numbers, budgets, and other account information related to the organization can be managed and organized using cost structures. These models formally establish the connection between the relevant accounts, projects, and goods. Additionally, we may see and analyze our sales data based on the inherent meaning. Using a regression model, I would like to analyze and define the factors relevant to predicting the cost of raw materials and manufacturing. Also, I can make a hypothesis structure on what factors might influence and cut down the project budget. The dataset is based on the economics of scale of X company and gives us 1000 different units and their measures.

So I looked at the bigger picture, analyzed what was happening industry-wide, and conducted external research. I saw if they are using a regression model in cost analysis and how it affect product quality and performance. I wanted to see the Morse way of costing and testing the prototypes, which would be very interesting. After exploring the data, there is a possibility that economies might have factors of scale. Therefore, each instance can be assigned with various categories; this type of problem needs to use multi-label classification. I used non-linear regression to test the model. This would be my first task to cleanse and start preparing the dataset.

**Data Preparation**

I pulled the manufacturing cost information from the Cognos database. The cost function also plays a crucial role in understanding how well your model estimates the relationship between the input and output parameters. It calculates the difference between the expected and predicted values and represents it as a single real number. I used linear regression analysis to estimate one variable's value relative to another's. The variable you want to predict is called the dependent variable. The variable you use to predict the value of the other variable is called the independent variable. Linear regression models are relatively simple and provide an easy-to-interpret mathematical formula that can produce cost predictions. Here I took the material cost and the total unit costs in the circular saw blade kine. This model will work best for comparing the types of cost that goes into making a single unit of circular saw blades.

**Creating a profile report**

This profile shows the number of variables, observations, missing cells, duplicate rows, and the actual size of the data set. It enables the models to take the gradient or direction to reduce the errors by reaching to least possible error. Here, direction refers to how model parameters should be corrected to reduce cost function. The error in your model can be different at different points, and you have to find the quickest way to minimize it, to prevent resource wastage. One of the consequences of increasing the efficiency of workers is the company's cost reduction. In addition, with machine learning, costs can be reduced even more since it predicts which services or products may have more demand. You can find business and sales opportunities without spending so many resources. Here, I only take the material cost for a unit because Cognos does not give the total manufacturing cost in the database. So, I moved forward using the closest values to create a scatter plot and look at the regression line.

**Splitting the data into train and test**

The quality of the products coming out of the machines is predictable. Statistical process control techniques are the most common tools on the manufacturing floor that tell us if the process is in control or out of control. Using a multi-label classification is the best way to train and test this model. It is used in classification problems for which instances are allocated to one of more than two classes. Now I want to use the logistic regression model available in the scikit-learn library. It is beneficial to have a library like that. I started by importing the logistic regression model from the scikit-learn library. Then fit our training data into the model. Now, we can use this trained model to predict the admitted or not admitted label and finally get the accuracy score. I also used the k fold classification method where, my approach was to training/testing split the total cost values for every unit. For k-fold cross validation, the dataset is divided into k parts. Each part serves as the test set in each iteration and the rest serve as training set. The out-of-sample performance measures from the k iterations are averaged. We have the following targets - to find the best fit for all our data points so that our predictions are much more accurate; to get the best fit, we must reduce the error; cost function comes into play here and cost function finds an error.

I used the following equation for costing for CIRCULAR SAWS only -

y = total cost, a = total fixed cost, b = variable cost, x = number of units

**Unit cost (Material cost)**

**y = a/x +b**

**Total cost**

**y = a+bx**

**Using sklearn**

I believe having access to the cost is useful. Because the cost function is a surrogate to your actual metric, it is helpful to see whether or not your actual metric is improving as your cost is minimized. This can give intuition into whether or not you should pick one cost function (model) over another or whether you should change your optimization algorithm. For example, after using sklearn.linear\_model to fit a training data set, I would like to obtain the value of the cost function for the training data set and a cross-validation data set. It might be possible to have sklearn simply give me the value (at the fit minimum) of the function it minimized. I found sklearn.metrics.log\_loss, but of course, this is not the actual function being minimized. This is the best way to look at the accuracy of this validation set.

**Polynomial regression**

The cost function measures the performance of a machine learning model for a data set. The cost function quantifies the error between predicted and expected values and presents that error as a single real number. The cost function can be formed in many different ways depending on the problem. The purpose of the cost function is to be either minimized or maximized. Every function has to be differentiable for algorithms relying on gradient descent to optimize model parameters.

In our total cost function *y = a+bx:*

The two entities can have changeable values (variable) a, which is the point at which the line intercepts the x-axis, and b, which is how steep the line will be, or slope. At first, if the variables are not correctly optimized, you get a line that might not properly fit the model. As you optimize the model's values, you will get the perfect fit for some variables. The ideal fit will be a straight line running through most data points while ignoring the noise and outliers. And these outliers are the factors that we need to eliminate during the production of every single unit. The more you minimize these overheads, the better you can fit into the scale. This function will be the minimum of the Root Mean Squared Error of the model, obtained by subtracting the predicted values from actual values. The cost function will be the minimum of these error values.

**Decision tree regression**

I used the mean squared error, one of the most commonly used and earliest explained regression metrics. MSE represents the average squared difference between the predictions and expected results. In other words, MSE is an alteration of MAE where we square those differences instead of taking the absolute value of difference of the manufacturing and over head costs. In MAE, the partial error values were equal to the distances between points in the coordinate system. Regarding MSE, each partial error is equivalent to the area of the square created out of the geometrical distance between the measured points. All the clustered areas are summed up and averaged. I used NumPy for every value to correspond to their coordinates and came up with a 2-D array to got the output.

**Random forest**

A random forest is an ensemble of decision trees. This is to say that many trees, constructed in a particular “random” way, form a random forest. Each tree is created from a different sample of rows. And at each node, a different sample of features is selected for splitting. Each of the trees makes its prediction. These predictions are then averaged to produce a single unit of saw. The averaging makes a random forest better than a single decision tree hence improving its accuracy and reducing overfitting. A prediction from the random forest regressor is an average of the predictions produced by the trees in the forest.

**Best classifier**

A multi-class classification cost function is used in classification problems for which instances are allocated to one of more than two classes. It is designed to be used with multi-class classification with the target values ranging from 0 to 1, 3 upto n classes. In a multi-class classification problem, cross-entropy will generate a score that summarizes the mean difference between actual and anticipated probability distribution. The value should be zero for a perfect cross-entropy when the score is minimized.

**Conclusion**

The critical parameter we draw from this model is the learning parameter which influences the the cost function. I explored the logistic regression to optimize the cost function. Building this model from scratch using raw reports made me understand that there is a broad scope of actually adding in any specific product and their costs of manufacturing and sales to see where we stand in the industry and how we as a manufacturing company can make a difference in our budgeting plans. Our strategic ideas can have a role of this model too. I will continue to develop this model from here and paly around with data around all our products. Time could be another factor that I want to add as well to see if there is a scope of optimization.

**References**

Baptiste, S. (2021, April 29). Optimization of a weekly production plan with python and Gurobi - part 1. Medium. Retrieved September 1, 2022, from <https://towardsdatascience.com/optimization-of-a-weekly-production-plan-with-python-and-gurobi-part-1-d1257ad29a9>

Cost function: Types of cost function machine learning. Analytics Vidhya. (2021, March 1). Retrieved September 1, 2022, from <https://www.analyticsvidhya.com/blog/2021/02/cost-function-is-no-rocket-science/>

Mwiti, D. (2022, July 21). Random Forest regression: When does it fail and why? neptune.ai. Retrieved September 1, 2022, from <https://neptune.ai/blog/random-forest-regression-when-does-it-fail-and-why>

Nagdev. (2020, February 27). Data Science in Manufacturing: An overview: Python-bloggers. Python. Retrieved September 1 , 2022, from <https://python-bloggers.com/2020/02/data-science-in-manufacturing-an-overview/>