FOUNDATION OF DIGITAL BUSINESS

Surojit Mookherjee

Vinod Gupta School of Management

Indian Institute of Technology Kharagpur

Week 07

Lecture 31

Lecture 31 : Data to Regression Model - Lifecycle

Good morning. My next module is about tools and techniques for AI, and the first topic I will cover is the data-to-regression model life cycle. So, you are talking about data, data, data, and we are talking about models, etcetera. So, I thought I would touch upon this topic of how you start with the data and the steps we take to finally end up with an AI model—a linear regression. I will take the simple example of a linear regression model. However, please note that this overlaps a lot with econometrics, a topic in statistics.

So, there will be a lot of statistical terminologies which I will not discuss here because this is not a statistics or econometrics session. The reason I am giving this is so you get an idea of what entails, what goes behind this. So, if you are not aware of these terminologies and the statistical or econometric methods, then you can study them if you are interested. So, I have given you the topics or some terminologies, etcetera. which you can, of course, find easily in many open-source programs on the internet.

If you want to learn more about that, feel free—it is a very interesting area. You can deep dive into any of these topics and learn more, and you can become an analytics person, really. So, if you want to be an analytics person or an econometric person, you can deep dive into these topics. I have listed them out and also suggested some tools, techniques, and courses briefly so that you can use these as inputs for your further studies. So, I will talk about the steps like data cleaning, data preprocessing, finding the regression model, testing the model, and deploying the model. So, what is data cleaning and data preprocessing?

So, you have a scattered data set, a set of data which is scattered, random data. So, how do you find a relation between Y and X, and how to find trends? So, we know an axis

like if you plot the data, we plot on a two-dimensional axis: the Y-axis, the vertical one, and the horizontal one is called the X-axis, a traditional mathematical expression. So, analyzing to uncover the relationship between various variables like Y, and we want to find an equation like Y is equal to something like X, etc. The simplest equation we know is of a straight line, called Y is equal to MX plus C. More about that later.

And so, we want to basically spot a trend, which often requires a combination of statistical and visual techniques. So, this is a structured way to approach this. So, this is what I am going to talk about. First is to understand the data. So, inspect the data, look for outliers, missing values, or inconsistencies.

So, if you have a set of data, something called an outlier means you will find that most of the data falls within a broad range. However, one or two or a few values are way beyond the range, either on the lower side or the higher side. So, these are called outliers. So, they are not lying within that main depth of the field or that region, but they are outside, far outside the region. So, they are called outliers.

So, there is a problem with this data. So, shall we or shall we not consider them as part of my dataset? Some values are missing, and some values are inconsistent, looking very odd and out of place, not matching the rest of the characteristics of the other data. So, these are the things you need to inspect. And then you categorize the variables.

So, determine whether X and Y are continuous variables, categorical, or mixed, etc. Now comes the visual exploration: you plot Y against X to visually assess the pattern. Once you plot that, you will find some pattern—it could be a straight line, a parabolic curve, etc. So, examine the distribution of X and Y. For a heat map, if it is multivariate—meaning there are multiple X variables (independent variables)—then correlation heat maps can highlight relationships between them, like X1, X2, X3, X4. Are they correlated? Is X1 impacting X2? Are they independent of each other? So, that is also a factor to consider.

So, if I do a heatmap, you can find out to what extent they are related—they may not be completely unrelated, which is probably not possible. So, there is some correlation. So, what is the extent of that relationship? Then you do a statistical analysis. So, you find out the correlation coefficient.

So, we are talking about correlation among these variables. So, how do you find out this correlation? These techniques are called Pearson's correlation for linear relationships and

Spearman's rank or Kendall's tau for monotonic, but non-linear relationships. Again, as I said, these are all econometric topics. So, if you are interested, you can always read about this.

I just brought it up here. So, these are the tools that are used. Regression analysis starts—the main thing is regression is to find out this relationship. So, the simplest equation is y is equal to ax plus b is a straight line and where b is the intercept that when x is 0 what is the value of y. So, if when the 0 at the y scale what is the value of b. So, the line starts from there literally if I do not consider the negative values of x and a is the slope of the line tan theta.

Then, if it does not fit a straight line—which we should always try first—you move to polynomial, logistic, or other non-linear models if the relationship is non-linear. But the first one you should try is linear regression. These are the simplest to understand. Then you do something called hypothesis testing, like ANOVA (analysis of variance), which can assess if trends or differences in data are statistically significant. Again, this goes deep into statistics, which we will not discuss here, but I am just giving you the pointers you need to know.

Trend analysis, moving average, smooths the data to visualize short-term and long-term trends. So, you can use the moving average method. So, these are again techniques—statistical techniques. Time series analysis, if data is time-dependent, so the x-axis is time, and with time, the value changes. So, you can use techniques like ARIMA or seasonal decomposition to identify underlying trends, seasonality, and noise.

So, seasonality is data varies with the seasons if you take years data every year you find something going up during month of spring, summer time something coming down the sales are decreasing the winter time etcetera. So, there are various products which behave with seasonal variations. Clustering algorithms find patterns in scattered datasets using methods like k-means clustering. So, you have groups that can classify this data into certain families or groups—that is clustering.

So, when you use this technique—k-means clustering. The advanced techniques now—we are getting into the AI regime, moving out of the statistical domain into AI. So, they are all quite interlinked with each other. It is called feature engineering—creating new variables or transforming the existing ones to uncover features, which are nothing but the independent variables. So, X1, X2, X3 represent particular features of the real-life situation.

It could be parameters like temperature, viscosity, density, or pressure—whatever. Machine learning models are algorithms like random forest or support vector machines, which can model complex relationships and detect trends when traditional methods fall short. So, that is where you are now moving ahead. Once you have to do linearization then you do a random forest or support vector mechanisms machines to model more complex relationships. The data cleaning is the crucial first step to ensure the quality and reliability of your model.

It involves identifying and addressing errors, inconsistencies, inaccuracies, outliers etcetera which I mentioned briefly earlier. So, handling missing values, if your values are missing there could be certain strategies. Again these are again going back to statistics. Deletion, remove rows with missing values if the amount of missing data is small and random. or remove entire columns if the variable is not critical or has too many missing values.

A particular variable or a feature has got many missing values do not consider it. But if the variable is not very critical not very important in signal so you can remove it. So, that is part of your data cleaning do not have unnecessary things. Imputation fill in missing values for this also there are certain common techniques again statistical techniques. Mean median mode.

So, replace the missing value with the mean for normally distributed numerical data or a median for a skewed numerical data or a mode for categorical data again these are very statistical. So, I will not discuss the details. However, point is that you have to replace find out what is the missing value and replace it. So, that is there is no gap and use any of these techniques. Regression imputation predict a missing values using other variables in the data set.

and k nearest neighbors imputation input values based on the values of the nearest neighbors again these are very statistical techniques. Treating the outliers. Detect the data points that are significantly different from other observations, visualization tools like box plots, scatter plots or statistical methods like the z-score or interquartile range can be used. And that strategies could be removal, delete the outliers if they are due to data entry errors and cannot be corrected or apply

mathematical transformations example log or square root to reduce the impact of the outliers. So, use a log value or a square root of that value. So, that the difference

decreases. Correcting errors and inconsistencies. Look for typos, inconsistent units or impossible values.

Example age is equal to 200 of a person. So, that is not right. Standardize the data. For example, USA, U.S.A or United States should all be consistent. So, you cannot have three types representing the same element like the country United States.

removing duplicates to identify remove the duplicate records from the datasets. So, this is all data cleaning activities, then you do the preprocessing once clean data then you do a preprocessing. It needs to be transformed into a suitable format for the regression model. So, something called a feature engineering creating new features derive new informative features from existing ones. For example, creating an age feature from a date of birth feature.

So, you have data but you convert that into a age or some interaction terms example a one feature multiplied by other x 1 into x 2 gives some other features. Encoding categorical variables, regression models require numerical inputs. So, label encoding assign a unique numerical value to each category or dummy coding similar to one hot encoding, but typically drops one category to avoid multicollinearity or double variable type. So, multicollinearity is indications of a particular two variables having some strong correlation with each other. So, in that case you can drop one of the variables.

Feature scaling, it ensures that all features contribute equally to the model training process. It normalizes the range of independent variables features in a data set. So, each of these variables must be within a some particular range, it should not be happen that one of the feature has a different completely different or a long or short range of that data. It ensures that features are on a similar scale. So, preventing any single feature with a large range from dominating the models performance, because if it has a large scale then it will play more dominating effect on the overall model.

So, you want all of the features to play somewhat very similar role or impact or effect on the overall model. So, no particular feature should be very biased. Splitting the data, divide the data set into training and testing sets. So, if you have say 100 data points, you say I will use 70 for training and 30 for testing. Training set is used to train the regression model.

So, that is the most important step of machine learning and the testing set is used to evaluate the performance of the trained model on the unseen data. A common split is 70

to 80 percent for training and 20 to 30 percent for testing. Obviously, we need more data for training it because that is what is defining the model and then how good was your model is can be proven by the testing data. Finding the regression model, this step involves selecting the appropriate regression algorithm and training it using the processed training data. So, choosing a regression algorithm the choice depends on the nature of the relationship between the variables and complexity of the data and the desired interpretability.

Some of the common ones are linear regression, multiple linear regression and polynomial regression. So, linear regression what was talking about initially assumes a linear relationship between independent and dependent variables and that y is equal to mx plus c. Multiple linear regression is two or more independent variables. So, y is equal to c 0 plus m 1 x 1 plus m 2 x 2 like that. So, it will be more number of x variables.

And polynomial is models a non-linear relationship by adding a polynomial terms of the independent variables. So, the independent variables will be x square or x cube or something like that. So, that it would not be a straight line, it will be something like a parabola hyperbola whatever some curve. The other one is called some of the others are like lasso regression, least absolute shrinkage and selection operator, another regularized linear regression that adds an L1 penalty term, it can shrinks some coefficients to exactly 0 effectively performing feature selection.

And there is something called support vector regression, it finds a hyper plane that best fits the data with a specified margin of tolerance, it is effective in high dimensional spaces. Then the common ones which used is a decision tree regression, a tree based model that splits data based on the feature values to make predictions can capture also non-linear relationship. And random forest from the name is similar that it uses multiple trees and ensemble or gathering of trees method that builds a multiple decision trees and averages their predictions. So, what is done by decision tree and then here you take an ensemble method that you have multiple trees and take an average of the predictions. So, obviously, it will be more robust and accurate than a single decision tree.

And finally, gradient boosting regression for the example some of these tools are XGBoost, LightGBM and CatBoost. There are also ensemble methods that build trees sequentially with each new tree corrects the errors of the previous ones. So, often provide state of that performance. So, it is going on a continually improving. So, the model gets more and more improved at each step.

The next part is training the model, feed the processed training data which you have cleaned and preprocessed to the chosen algorithm. The algorithm learns the relationship between the features and the target by optimizing its internal parame the coefficients in the linear regression to minimize a cost function that is mean squared error. So, you have this coefficients of for each of the variables independent variables. So, those are you have to keep changing till you get the best fit and your mean square error is minimized.

The mean square error is the model which you have selected the straight line and then the difference from that line to each of these points is the error value. So, if you sum up all these errors because there will be say whatever 100 points, 1000 points or whatever and you have drawn one line. So, you can calculate the distance the perpendicular distance from a point to the line. So, if you tabulate that and square it and average of that should be you have the lowest when you get the lowest average value that is the line which will be the best fit line. There is something called hyperparameter tuning that parameters that control the learning process of a model.

So, they are not learned during training, but rather set before and to influence how the model will learn from the data. Examples can be the learning rate, the batch size, the number of layers of nodes or polynomial degree etcetera. So, those all of these features you define before you train the model. And the improved model performance by finding the optimal hyper parameter settings you can significantly improve the model accuracy and generalization capabilities. So, this is completely a iterative process.

So, all of these things the optimal parameter settings you can decide and similarly you can select the various coefficients, the weightage you want to give to each of these coefficients to get again finally goal is to minimize The squared error at the lowest value is your best fit line. So, that becomes the model. So, you can use that model now to predict any y for any x. What is machine learning model training?

In the machine learning model, model training refers to the process of allowing a machine learning algorithm to automatically learn patterns based on the data. So, these patterns are statistically learned by observing which signals make an answer correct or incorrect. In supervised learning, or by discovering the inherent patterns in data without being told the correct answer, we call it unsupervised learning. The end result of the training process is a computer program, also known as the model, that can now make decisions and predictions on data it has never seen before. So, this is your final goal.

To develop a computer program, or what you call a model or that equation, or whatever it is, based on which you can predict an output based on data which has not been trained or seen before—new data. So, data preparation from your data warehouse or data store, you prepare the data, then into the model training, then you have a trained model, and then you make predictions. So, what is machine learning model training? Machine learning model training is one of the key steps in the machine learning development life cycle. It is usually an iterative process, as data scientists have to train the model, inspect the model's performance, and fine-tune accordingly before repeating the process.

So, it is an iterative step-by-step process—go on repeating, go on repeating, change it, repeat, see the result, go on changing it, change something again, run, see the result. The fine-tuning step can involve tweaking the settings of the algorithm, adding more data, and changing the signals, known as the features, used for learning. So, the features are the coefficients of those independent variables. you can add more data to improve it or else you can pick the settings the hyper parameter settings and there could be other steps involved and it just depends on the algorithm and the problem that is being solved.

So, we have taken the simple algorithm the linear regression model, but they could be more complex as I have told earlier and so you can have other settings involved till the time you minimize the error. The model training stops with the performance is found to be acceptable on a data set that was not used for development. In real life scenario what happens is when you have all data sets are always limited data set because you cannot have infinite data set. So, you start with a certain value of number of data sets and you develop a model and then you find out that your prediction accuracy with real life situation is may be 60 percent or 70 percent.

So, now the idea is when will you say it is an acceptable or deployable model, it will obviously vary from case to case. So, let us take the case of medical imaging for example. So, you image scan, scan reports and you want to detect and find out whether this scan report is indicating say lung cancer or not. And you have trained the model on say may be 500,000, 10,000 scans and then you find out that the accuracy of prevention, accuracy of prediction is may be 80 percent, 85 percent. So, 80 percent of the time it will correct, 20 percent will not be correct.

So, when it is not correct that is false positive happens, it says it is cancer, but it is not. Then what is the implications? Implications for the patient, for the doctor, for the hospital. It is very high obviously, we are talking about a disease like cancer, if you tell a

patient that he has a cancer, but he does not have it and he undergoes all of the tests and starts medication or chemotherapies and stuff like that, these are very you know not only they are expensive, but they have lot of effects side effects.

So, in such cases, you need to be confident to the level of maybe 95, 98, 99 percent, etcetera. But there are cases where the use cases are very simple models, etcetera, with not so much large impact, but you are just probably automating things and you know stuff like that. So, there you can decide that fine with even with 80 percent or the cost impact is not but my convenience is high, giving me some automation, etcetera, and some relief, then probably I can start with deploying it even with an 80 percent level of acceptability. But the goal should be that as you start using it, you are generating more and more data points.

So, you feedback all those new data points that is the actual case—whether it was right or wrong or whatever—and then feedback those data points to further enrich your model. The more data you give for training, the better the model will be. So, I have a model to start with, I deploy it, and then I go on developing and fine-tuning it with more and more data because I am getting more data with use. So, I am using that to make my model better and better and better. So, that is how this machine learning model life cycle works.

Testing the model. After training the model, you must evaluate it on unseen data. Making predictions: use the trained model to predict the target variable for the instances in the testing set. Now, what are the evaluation metrics? So, how do you test, and what is the result?

So, to select an appropriate matrix to assess the model's accuracy, some of the common ones are the mean absolute error, the average absolute difference between predicted and actual values. The mean squared error. So, one was mean error, then you square it—the average of the squared difference between predicted and actual values. It will penalize larger errors more heavily because you are now squaring it. And then the root mean squared—the square root of the MSE—it is in the same unit as the target variable, making it more interpretable than the MSE. So, these are all, again, very statistical.

So, you will get—if you want to find out what the real differences are, etcetera—you have to read a bit of statistics or econometrics. Interpreting the results, analyze the metric values to understand how well the model is performing, compare the performance on the training set versus the testing set. A significantly better performance on the training set might indicate something called overfitting. This happens when the model has learned the

training data too well, including noise and outliers, and does not generalize. So, it becomes kind of biased.

So, it will always try to give more say that positive answer what you trained for output, but it will not be able to give the other side the negative or the opposite answer, because it is too biased toward giving the better fit. Residual analysis: examine the residuals—the difference between the actual and predicted values, which is the error part. Plot the residuals and look for patterns. Ideally, residuals should be randomly scattered around 0, suggesting that the model has captured the underlying relationship. So, if you analyze the residual—the error part, actual versus the predicted result—and plot it on a graph.

So, you should have something like a normal distribution or it should be very scattered around the 0 part—both sides, maybe positive or negative. So, it is showing you that it has really understood the model. But if it is only giving positive errors or all the errors on the negative side, then it is overfitting or underfitting. check for normality of the residuals for some models the linear regression the residuals are assumed to be normally distributed and this can be checked using a QQ plot (something known as a QQ plot) or histograms. So, all of these will help you analyze or interpret your results, again using statistical and econometric tools and techniques.

Now, deploying the model: once you have a satisfactory model, the next step is to make it available for making predictions or whatever in a real-world data scenario. To save the model, you serialize the trained model into a file. Common libraries for this include pickle or joblib in Python. Choosing a deployment environment—again, I am giving all those terms and names so that, if you want to, you can further deep-dive into any of these areas. Choosing a deployment environment: batch prediction and real-time prediction. Batch is when the model makes predictions on a batch of data at scheduled intervals.

Example: nightly or whatever. Results might be stored in a database if you want to get it in batch mode. So, if you have, say, manufacturing or production going on, you might want to capture certain data by batches, by dates, or whatever. Real-time prediction is online prediction. The model—the most commonly used one—is integrated into an application and provides predictions on demand via an API endpoint. So, it is available to customers or whoever, anytime they log in and use it. You can use it as a web service—deploy the model as a web service using frameworks like Flask, FastAPI, or Django in Python.

Cloud platforms: utilize cloud services like AWS SageMaker, Google AI Platform, or Azure Machine Learning for scalable deployment and management. You can also use it on edge devices for applications requiring low latency or offline capabilities—deploy the model directly onto edge devices. Examples: mobile phones or IoT devices, etc., which are very close to you—not the cloud, which is normally somewhere else. So, the latency factor is much lower in edge devices. As one of the major application is autonomous car where the response time needs to be very fast or even say automated or robotized surgical operations.

So, these are again various AI tools platforms etcetera which you can use to deploy your AI model for actual use. Building an API for real time deployment create an API endpoint that accepts the input features and returns the models prediction. So, you have to give an interface user interface to this basically user interface where user can use the model. So, ensure the input data undergoes the same pre-processing steps during training. So, you have to take care of the input data for the model because the input data also should not be like.

So, pre processing you are cleaning the data part of the data cleaning steps or you are organizing them in particular format. So, number range or number code like that USA example I gave you u dot s dot a or USA or United States. So, all the input similarly when you are doing it for the real life also should be similar to what you have used during your training. Whatever discipline data discipline had been practiced during used during training. Now, for monitoring and maintenance, performance monitoring can track key metrics to detect any degradation over time.

So, the models can degrade over time. How? Data drift and concept drift. The statistical properties of the input data has changed over time. Concept drift, relationship between the input variables and the target variables change over time.

To understand this suppose you take in the COVID situation and post COVID situation. Many things have changed. So, many data also getting impacted. If I take some do a survey on how much online shopping you do, what you did during COVID, now what you are doing now or what was pre COVID, all three will find very drastic change.

So, the data point obviously gets changed pre-COVID hardly any online shopping, during COVID fully online shopping, post-COVID somewhere in between etcetera. So, everybody the behavioral patterns have also drastically changed which all of us have understood now because all of us have experienced a major disruption called COVID and

lockdown. Retraining, periodically retrain the model with new data to maintain its accuracy and adapt to changing patterns. The frequency of retraining depends on how quickly the data or underlying concepts change. So, you have to keep retraining, get fresh data to update the dataset, so that your model remains more effective.

Document the models architecture, data sources, pre-processing steps, performance metrics and deployment procedures all of these must be documented for Because you have developed the model, but somebody else will in future probably will be maintaining it, you may not be there. So, everything this typically standard for any IT product development the entire step the entire life cycle must be duly documented. Now, some tools for regression analysis and trend detection. Statistical tools include Python, SPSS, data visualization tools like Tableau and Power BI.

These are various types of tools you might want to learn. It is up to you, but these are, of course, very important and useful for an AI professional. Some machine learning platforms are TensorFlow and PyTorch, which are useful for building advanced regression models and detecting complex trends. RapidMiner is a no-code platform for predictive analytics, trend detection, time series analysis, and specialized trend analysis tools. I have listed these for your reference. If you want to learn about them, you are highly encouraged to do so. And some other resources which will help you to learn regression analysis, some tutorials and blogs

mentioned some of these, but there are n number of courses available online, but you have to be very slightly judgmental confuse you. With that, I will conclude this session on tools and techniques for the linear regression model development lifecycle. Thank you very much.