Data-centric approach for building robust machine learning model

Abstract

During the transform 2019 event, which talks about how artificial intelligence technology is actually producing returns on investment, it has been said that 87% of the data science projects fail in production. Some of the possible reasons for these failures are poor data quality, lack of model monitoring, and poor project planning. Recently, Andrew NG, one of the AI leaders, emphasized on data-centric approach to improve the data quality. Currently, most data scientists, machine learning engineers focus on improving the accuracy of the machine learning model while keeping the data constant. Most of the time is spent on experimenting with the model and its hyperparameters but less time on doing an error analysis on data post-training. Data-centric approach emphasis keeping the code/model constant and iteratively improving the data. I have developed a use case following a data-centric approach. The use-case is to classify YouTube comments into “Questions”, “Request”, “Suggestion”, “Misc”, “Praise” and then showing the top-k intents to the YouTubers so that they get to know about their subscribers better and improve on their content.

Data-centric Vs Model-centric

Most data scientists and machine learning engineers start with a dataset available either from an external or internal source, perform basic preprocessing, and develop a model to get high accuracy on the test set. They iteratively change hyperparameters and the model until they get a better test accuracy. This way of changing the code or the model while keeping the data constant is a **model-centric** approach.

For the past three years, we have seen machine learning models developed by researchers that are good at understanding languages, images and give high accuracy. For structured data, it has been published that gradient boosting machines do better than deep learning models. Now that we have models that can understand the data better and produce good results, at least for classification and regression tasks, it is time to move from model-centric approach to a data-centric approach.

In data-centric approach, we keep the code or the model constant and iteratively improve the data. Data can be improved by collecting data that covers all the edge cases, making the label definition consistent, training the model, doing an error analysis on why the model predicted the inputs incorrectly, and again increasing the data based on the error analysis.

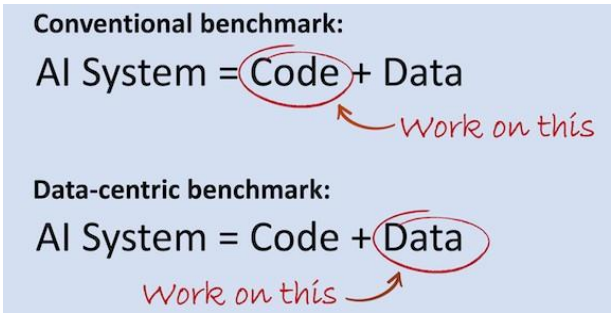


Fig 1.0 Model centric Vs Data centric

Machine Learning Life Cycle

There are four phases in a machine learning project

1. Defining the project
2. Defining and collecting the data
3. Training, error analysis and iterative improvement
4. Deploy, monitor and system maintenance

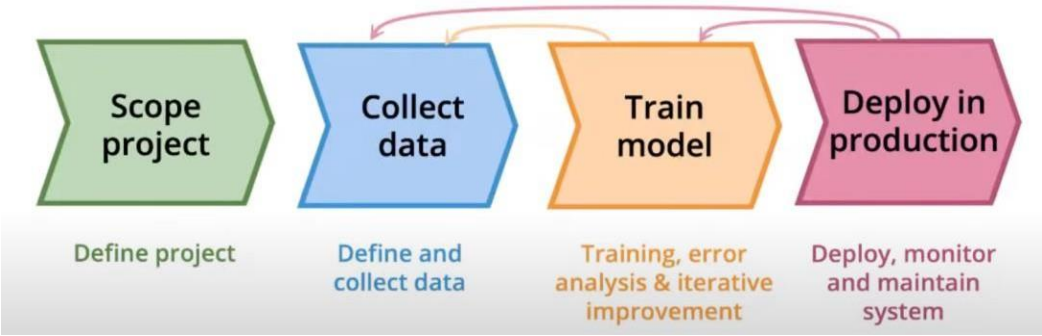


Fig 1.1 Machine Learning Life Cycle

We iteratively improve the data for getting good results in the third phase (train model)

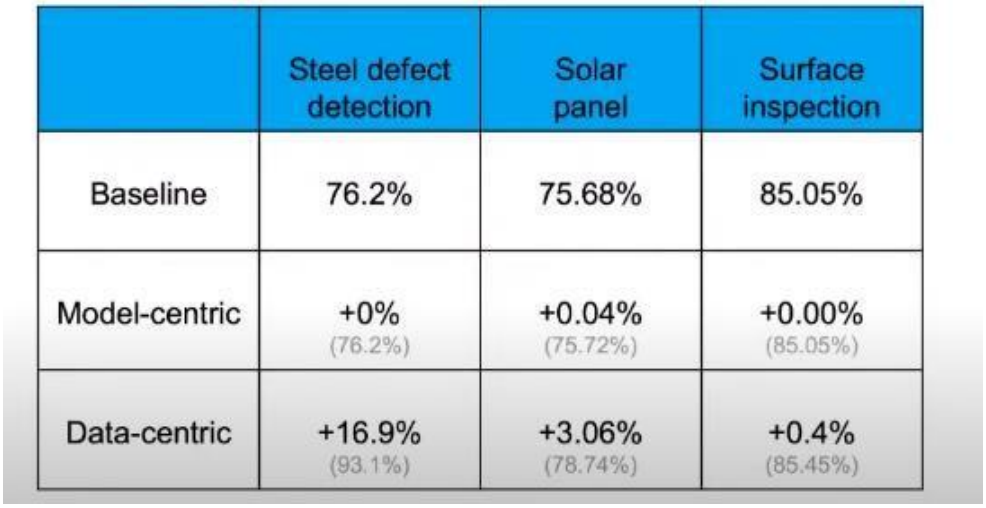
1. Defining the project

The scope of the project is where we define the problem we want to solve. For example, for a telecommunication company, the scope of the project might be to retain their customers. The scope of the YouTube intent classification is to help YouTubers understand their audience better and improve their content.

2. Define and collect data

Once the scope of the project is defined, we need to define what data is needed and the source of that data. For this use case, text data is needed and the source of the data is YouTube. YouTube Data Api is used to collect data. The text data is collected from several videos of various categories. There are five classes **(“Suggestion”, “praise”, “Request”, “Question”, “Misc”)** and the datashould be collected such that there is no class imbalance.

Initially, 700 samples were collected to check if the model works. Data should be accrued and while collecting, the meta data about the collected data should be maintained so it would help us in the error analysis. Once the data is collected, we need to label each comment. A clear definition of what each class means should be understood so that each instance will be labeled without any inconsistency. Label inconsistency is one of the major reasons for poor model performance. If labeling is done by a group of people, then proper instructions should be given to label the data consistently. If the data is labeled with consistent and enough diverse data is collected then the results should be better than improving the model hyperparameters and the below image is the evidence for the claim. This report was given by Dr.Andrew Ng.



	Steel defect detection	Solar panel	Surface inspection
Baseline	76.2%	75.68%	85.05%
Model-centric	+0% (76.2%)	+0.04% (75.72%)	+0.00% (85.05%)
Data-centric	+16.9% (93.1%)	+3.06% (78.74%)	+0.4% (85.45%)

Fig 1.2 Data-centric Vs Model-centric results comparison

3. Train Model

In this phase, we choose the best model for the given task. For the use case, bert model with 6 layers and two attention heads was used. After training, error analysis has to be done. We need to inspect why the model predicted certain comments incorrectly. Inspection is done on the data rather than on the model. Model interpretation tool, LIME, was used to carry out error analysis for the use case. We can know what features the model picked for classifying a comment to a certain class. If the model was picking up wrong words or there are fewer examples of sentences with words that contribute to the correct class, then we need to collect more data.

LIME method is one of the tools for performing error analysis and this is the method followed in the YouTube comment intent classification use-case. Error analysis methods changes based on the type of data and use case. For example, in speech recognition use-case, doing error analysis on falsely predicted data involves checking for background noise in the speech. If a falsely predicted speech input contains car noise in the background, then more data with car noise should be added. LIME will tell us which words in a sentence contributed to predict a certain class. Based on the model interpretability, we can decide what information is lacking and why the model predicted incorrectly. We can collect more data after the analysis. It will be very useful to version the data and store the metadata about the collected data so that we can trace the source of the error that we would find in the error analysis. The data versioning can be done in excel. The below image is an example of performing error analysis on the YouTube comment intent classification use case.

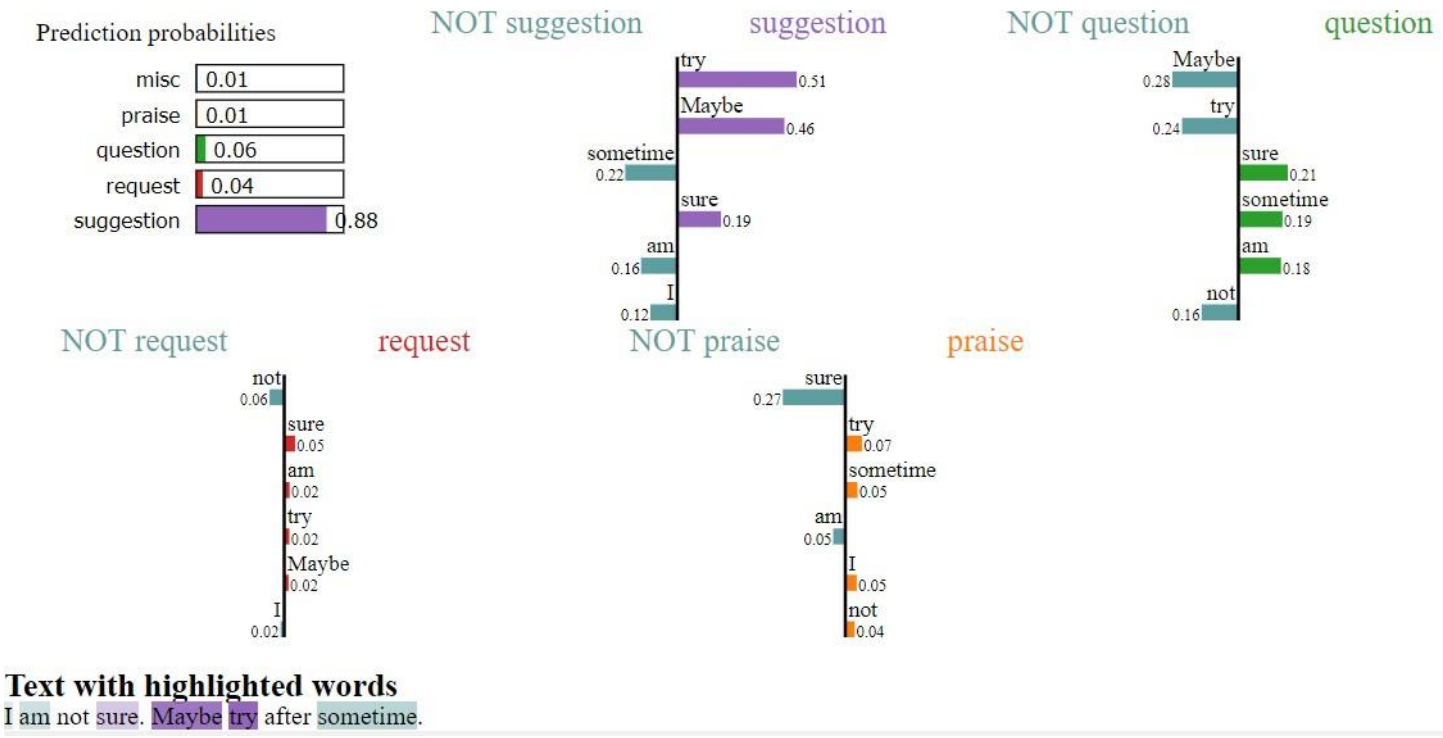


Fig 1.3 LIME result for correct prediction

From the result, it is evident that the model takes the words "Maybe", and "try" as a strong indicator of the class "suggestion" while the word "sure" is less strong. The words "sometime", "am", and "I" are the words that does not contribute to the score of "suggestion".

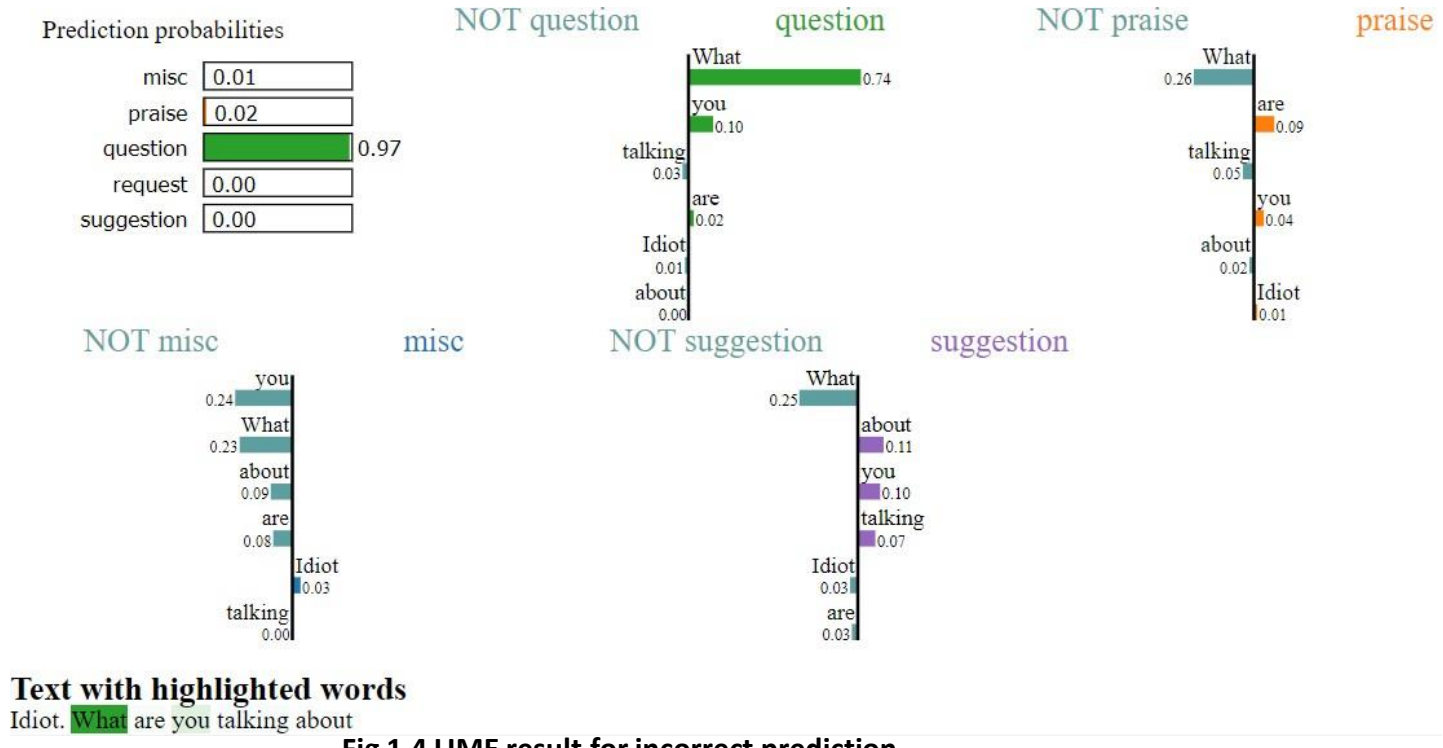


Fig 1.4 LIME result for incorrect prediction

The comment “Idiot, what are you talking about” might look like a question but it’s a negative comment. It belongs to **misc** class. It is evident from the result that the model looks at the word “What” to classify the comment as “Question”. The model does not recognize the word “Idiot” as negative; hence more data should be collected containing criticizing words along with questions/suggestion/request to let the model learn.

Once the error in the data is found we can go back to the **collect data** phase, then train the model and again perform **error analysis**. This is how we iteratively improve the data thereby improving the model performance. The model **bert** with 6 layers and 2 attention heads is kept constant here.

4. Deploy in production

Once we get good model performance, we need to deploy the model in production. Getting a good model is only 50% of the work done. The machine learning system needs to be maintained after deployment. The model was trained on the data with certain distribution, but this distribution tends to drift away in the future that would deteriorate the performance of the trained model. Cloud services like Amazon Web Service have model monitoring services to monitor the trained model and raise alarm if it identifies concept drift and data drift.

Once the drift is detected, the new data has to be trained again, and then error analysis needs to beperformed.

Conclusion

The data-centric approach will help companies to get the most value out of data, and will make the machine learning model produce sensible predictions. All data scientist and machine learning engineers should follow this approach to get the most value from the data thereby providing morevalue to the business.