**Opening statement along with first slide from power point**

In this video we will be taking a look at how we build the customer feedback sentiment analysis predictive model using AI and Machine learning, and also, we will show this in action.

For this prototype we are using Jupyter notebook environment, including Scikit-learn for classical machine learning and TensorFlow for Deep Learning.

Let’s go through the steps of our framework and see how we build this predictive model using Jupyter notebook.

Stage 1: Business Understanding. This is about building a predictive model that can predict the positive or negative sentiment of customer feedback. We look to achieve accuracy of at least ~85%.

From machine learning view point, this is a binary classification machine learning problem.

Stage 2: Data Understanding which is about understanding the raw data use for building the model. In this case we have customer feedback record data set having 3.4 million of records from Amazon.

Here are two raw customer feedback data point examples

We import the data using **Pandas** and we need set up appropriate positive or negative reviews so that we can use this for training our model.

If you look at the data, the column overall is the key data point to determine the customer sentiment based on their ratings

Moving into stage 3: Data Preparation. All the ratings generally in the range of 1 to 5.

As this is a binary classification problem, we need to convert ratings to 1 or 0, 1 for positive and 0 for negative, hold the value in a derived column review\_class for to train our model.

In order for this data to make sense to our machine learning algorithm, we’ll need to convert each customer feedback review to a numeric representation, which is called vectorization.

In machine learning, items are represented by their features. In our case, customer feedback reviews are represented by their words, so we will use words as features. Typically, we assign appropriate weights to different words, and In this case we are using TF-IDF, TF-IDF is one of the most common weighting schemes used in natural language processing.

In Machine Learning, we always divide our data into training and testing part, that means, we train our model on training data

and then we check the accuracy of a model on testing data. In this case we split as 70:30.

Stage 4. Modeling: For demonstration purpose, we build two models. The first model is classical with Scikit learn and second one is Deep Learning with TensorFlow. Both models serve the same purpose, but they are different architecture

Stage 5

Evaluation.

Evaluating machine learning algorithm is an essential part.

This visual called the ROC curve, which measures the true positive rate against the false positive rate at various threshold; and here is AUC curve to evaluate our model

We are able to achieve EIGHTY-NINE PERCENT accuracy.

This visual is called confusion matrix , which shows you numerically how the model performed.

If you look at the numbers, 11550/80931 true positive and true negative are classified correctly.

The classification report, which displays the different score of the model such precision, recall, F1 score, and support.

Moving to stage 6: Deployment. The last process of our framework is the deployment.

We have trained our model and achieved very good accuracy score ~89%.

Now we will use this model for our real time demo.

Let’s move into demo, we have developed a simple API which takes input as customer feedback review text; and pass the information to model to predict.

Our model will predict the customer sentiment in zero or one. One for positive and zero for negative.

We will see them in action in a moment.

Two models have been deployed, model one is classical and model 2 is deep learning.

Both models serve the same purpose, but they are architectural different.

Let’s test our models, Easiest thing I can do is to find some review about BlueShield of California instead of typing. Let’s search yelp review.

Here is the positive one. will pass this customer review text to classical model. Ah ha happy customer.

To test negative sentiment, let’s find a negative one, here you go. Let’s pass this customer review text to our model to see the result. Ok, Pretty sad customer.

Let’s also test our deep learning model, pass positive review, got it.

Now pass negative one to see, here you.

By applying machine learning, on our customer data, we can easily find the sentiment of our customer.