**Opening statement:**

In this video we will be talking look at how we build the customer feedback sentiment analysis predictive model using AI and Machine learning.

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

Today we will be taking a look at building a sentiment analysis classifier with python, Scikit-learn, TensorFlow and Keras.

The machine learning library for python such as Scikit Learn, TensorFlow and Keras libraries for Deep Learning is being used for this exercise. Keras Deep Learning library is built on top of TensorFlow as backend.

First let’s talk about the environment that we are looking at. Jupyter notebook, which allows you to create and share documents that contain live code.

Let’s walk through the steps of our framework and explore how we built this classifier.

**Model Building**

**Model Demo**

Stage 1: Business Understanding. This is about building a predictive model that can predict the positive or negative sentiment of customers based on their feedback. We wish to achieve a model accuracy of ~89%

Stage 2: Data Understanding. We start with raw data of customer product review from amazon. The dataset has 3.4 millions of record.

First Step: Data importing using Pandas, Pandas is great for data manipulation and analysis.  Pandas DataFrame gives massive functionality to work on data.

If you look at the data, it has 9 columns, the third attributes known for overall. This column has been used to represent feedback of customer.

As this is binary classification, Customer feedback has one of 2 possible classes either positive or negative.

If we study the “overall” column rating data, here is the graph to see how data has been plotted across. The Data set contains more positive review than negative.

Now, we understood by looking at the data, if there are a good five-star ratings, it okay to assume, definitely has some positive words in the review. if the rating is below average that’s means it’s not positive feedback and negative words in review text.

Moving into stage 3 Data Preparation. Let’s take a closer look at these two columns such as overall and reviewText. The overall contains rating of the product and reviewText it contains comments.

Let’s apply our assumption to prepare our data set, so that we feed the data set our model to train. if rating is greater than three then return one for positive else zero for negative.

We have created a separate additional column known as “review\_class” to and existing data set to hold positive or negative value.

Now let’s plot graph to see how its looks like to get some kind of intuition.

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

In classification, items are represented by their features. In our case, documents are represented by their words, so we will use words as features. Typically, we give appropriate weights to different words, and [TF-IDF](http://en.wikipedia.org/wiki/Tf%E2%80%93idf) is one of the most common weighting schemes used in text analytics applications.

So, TF-IDF is stands for Term-Frequency-Inverse-Document Frequency. weights down the common words occurring in almost all the documents and give more importance to the words that appear in a subset of documents.

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.

Testing your model on testing data will only help you evaluate the efficiency of model. Train test split ratio in this case is 70:30.

Stage 4 Modeling:

To build classifier, we applied two Machine Learning techniques, one is Logistics Regression with Scikit learn for classical, and Artificial Neural Network for Deep Learning for building two models.

**Stage 5 Evaluation.** Evaluating your machine learning algorithm is an essential part, we will be discussing different types of evaluation metrics of our model such as ROC, Accuracy, Confusion Matrix and classification report

The ROC (Receiver Operating Characteristic) curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. Area Under Curve (AUC) is one of the most widely used metrics for evaluation. It is used for binary classification problem

Accuracy: Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

Confusion Matrix: A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") with two rows and two columns that report the number of false positive and false negative

**Classification Report:** The classification report visualizer displays the precision, recall, F1, and support scores for the model. In order to support easier interpretation and problem detection, the report integrates numerical scores with a color-coded heatmap.

**Stage 6 Deployment:**

The last process of the entire framework is the deployment. We have trained our model and achieved very good accuracy score ~89%. Now let’s save the model for deployment.

We have developed a simple web application to interact with our trained models. We can see them in action in a moment.

**Application Demo:**

Here is the web application, two models have been deployed to test the sentiment of customers using real world data.

Easiest thing I can do is to find some review about BlueShield of California instead of typing. Let’s search yelp review. let’s test in action. Ahha happy customer.

To test negative sentiment, look for negative review, here is one. Here you go.

Using machine learning technique, we can easily reach out the unhappy customers.