Script

Good afternoon Everyone. me and my team are here to present our Hackathon Project  
"AI-Enabled Interactive Analytics Platform� for UPS Call Centers".  
Our team name is "You show it and we know it" and you will soon know why

===================================  
Lets quickly go thru the roadmap for this project,  
After showing our business case, we will show you the architecture we designed.   
We coded and implemented 3 artificial intelligence solutions: voice emotion detection,   
conversion of speech to text, and sentiment analysis of text.   
We present these in a Streamlit web app.

===================================  
We all know that -   
UPS’s motto is: Customer First, People Led, Innovation Driven.  
We want to listen to our customers – literally \*listen\* - to how we can avoid churn and improve customer experience.

===================================  
Existing customer experience initiatives are improving our bottom line. Expanding our analytics on the \*voice\* of the customer can help.   
UPS’ current analytics platform leverages the Verint platform. 10 million calls are transcribed,   
with covered languages expanding. It does report text-based sentiment, but we understand that it does not directly analyze audio.   
Our Hackathon work demonstrates that UPS has in-house the technical capabilities to produce customized and flexible solutions   
to UPS business cases involving audio and text.

===================================  
For the Hackathon we focused only on analyzing calls after they occur. However, this work can be extended in many ways:  
•Incorporating voiced emotion with post-call surveys  
•Offering real-time agent assistance  
•Agent monitoring, evaluation, and script improvement  
•The data from the calls can be used to improve the UPS Chatbot and social media services  
•Support diverting calls to alternate channels  
•Continue customization by sharing information with Operations, Marketing, and Sales

===================================  
We aim to build a dashboard to show the emotional states of callers at the beginning and end of calls and relate these to metadata about customer experience  
Our product enables:  
\*identification of customers at risk for churn   
\*identify content of calls  
Monitor customer sentiment over time and place and by characteristics of customers and shipments  
Improve call center operations to build stronger script for training agents

Thanks, Sayali for the introduction. About the architecture of the product, In AI Algorithms side:

•We first extract voice features out of customer audio and use those features to predict the emotions of the speaker

•On the other hand, we transcribe what customer said into text and also do the text feature extraction to predict emotion of the speaker, then we push everything to the user interface side which is an interactive dashboard in our project , and we let analytic user interact with dashboard for further insight. NEXT SLIDE PLEASE !!!

======================

Regarding the detail on how we detect emotion from voice:

•First, the emotion detection system takes in audio tracks, and extracts features like frequency, cadence, and some other sound features

•And then it put those features into a neural network multilayer perceptron model, and output of the model will be an emotion label which is either Negative or Positive. And Now I pass to Hao for the Text Sentiment Analysis

Slide 8 – thanks Quan

* Transcribing audio data involves an API request. Multiple cloud providers such as IBM, Google, and Azure offer these APIs. We compared these providers according to two metrics, Word Error Rate and Word Information Preserved, and we found that Azure performed the best.

Slide 9: next slide please

* The features extracted from the transcription will use for text analysis. There are several packages that we have tested. Ultimately, we choose Flair because it is ver/sa/tile and giving us the best accuracy.

Slide 10: next slide please

* Here is an example of testing how our algorithms perform at labeling the emotion in one tricky text: ‘I’m not afraid of failing’. Both algorithms were able to correctly label the emotion as positive.

Slide 11: next slide please

* Data scientists do not typically build front ends. BUT Bringing the front-end prototypes into data science can be beneficial. For example,
  + It can speed up the prototyping process
  + facilitate communication with the business
  + allow data science concerns to be reflected in design (which includes facilitating user inputs and interactivity)
  + and reduce project cost
* While using a Python-based framework may have advantages for integration, Streamlit is one of the easiest to learn, which we did specifically for the Hackathon. For another iteration, we could consider Dash, Flask, or Django (jango).

And over to Katherine for the demo

The business analyst wants to be able to customize how they interact directly with the data.

Here, the data is synthetic as use of UPS data is not allowed in the Hackathon.

The raw data can be accessed in this interactive dataframe, you can choose which columns to display.

To identify which calls, deserve further attention, we sort by sentiment. If you would like to drill down on a call, you can enter the call number, listen to the call, view the transcription, and record your notes for follow-up here.

**Paste this: “Check with operations, then have Sales contact Acme to follow-up on missed pickups”**

Call sentiment can be tracked by customer, shipment, or consignee characteristics to reveal in which groups we have customer experience concerns.

A word cloud is a window into the content of the call. Streamlit allows us to update the word cloud according to the sentiment.

We can also track the sentiment of the calls by location and months into Covid, and with real data these points would update.