WEAK SUPERVISION AND ACTIVE LEARNING WITH TEXT DATA

TRAINING A MULTI-CLASS CLASSIFIER WITH WEAK LABELS AND IMPROVING IT WITH ACTIVE LEARNING

TEAM 44 - FSDL 2022

Aleks Hiidenhovi

Bernardo García

Kushal Ramaiya

Edoardo Abati

Juan Manuel

Diego Quintana

Sal El Kafrawy

TABLE OF CONTENTS

01

INTRODUCTION

TRAINING A MULTI-CLASS CLASSIFIER WITH WEAK LABELS AND IMPROVING IT WITH ACTIVE LEARNING

INTRODUCTION

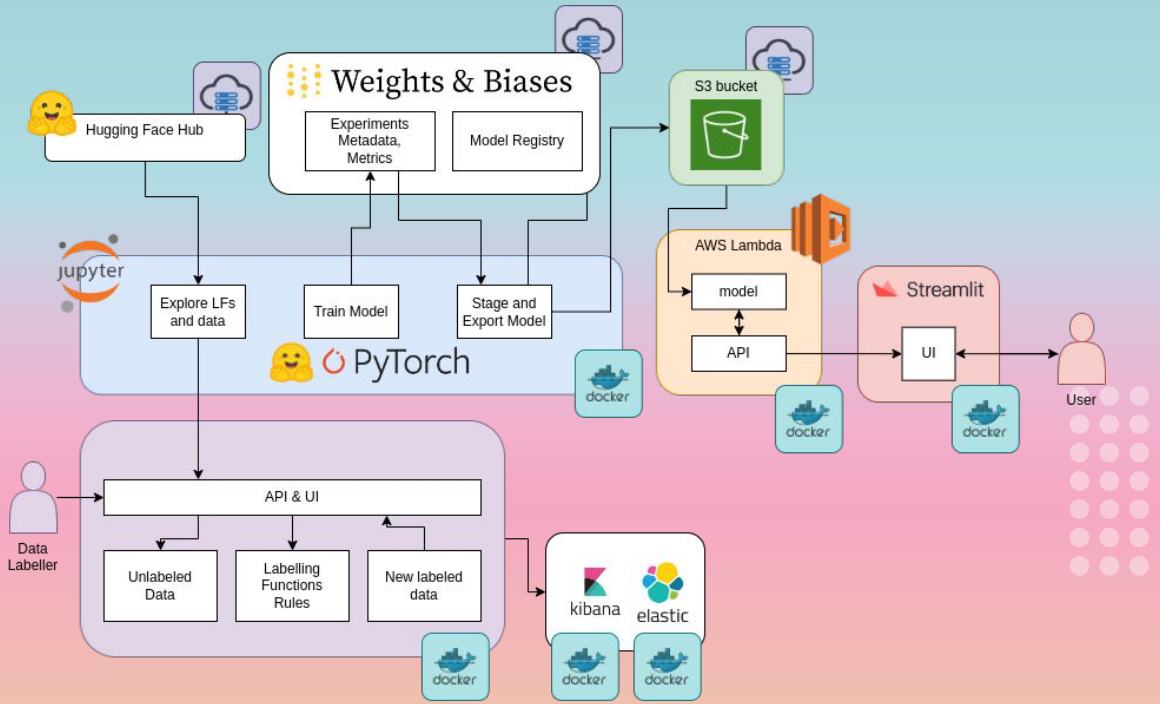
- **News classifier** → Multi-class classification
- We combined **weak supervision** and **active learning** to **minimise** the amount of time for human labelling
- **Data** → We used the AG news dataset without labels for the training stage
- We used an open-source library **Rubrix** to create an efficient workflow
- The classifier consists of a fine-tuned **DistilBERT**:
  - The model was trained with **PyTorch** and **HuggingFace**
  - experiments were tracked with **Weights & Biases**
  - The versions of the model were stored on **W&B's model registry**
  - We deployed the model using **AWS Serverless Lambdas**
  - We implemented a UI using **Streamlit**
- **Code is available at** https://github.com/EdAbati/fsdl-2022-weak-supervision-project

02

DEV ENVIRONMENT SETUP

Development environment

- Six containers on **docker-compose**
- **Streamlit** for UI
- **Development container** with jupyter and dependencies for training and deploying models
- **Rubrix service** for supporting labelling and active learning loop
- **Elasticsearch** and **Kibana** to support Rubrix
- **AWS Lambda API** for serving the model



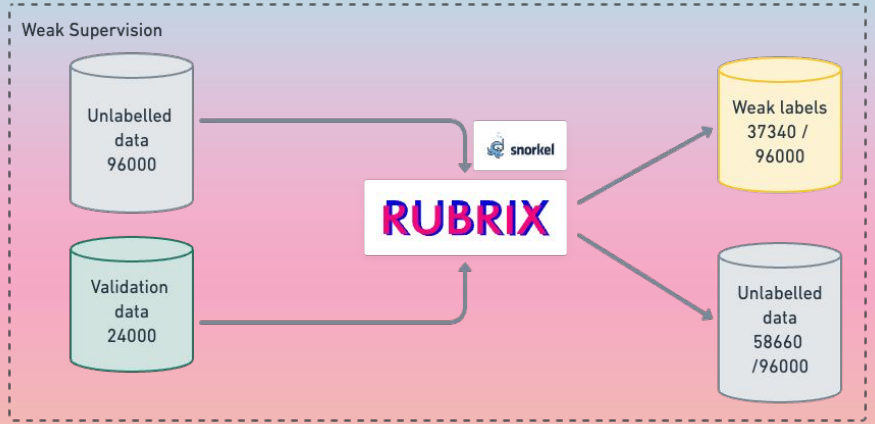
03

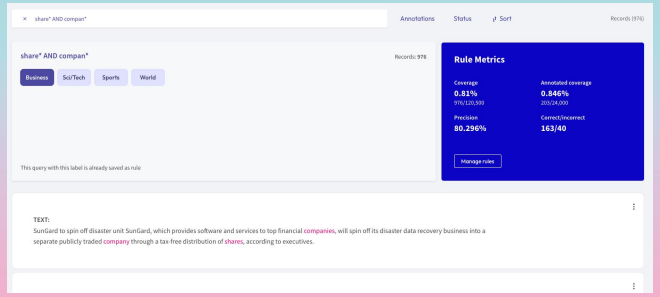
WEAK SUPERVISION AND ACTIVE LEARNING

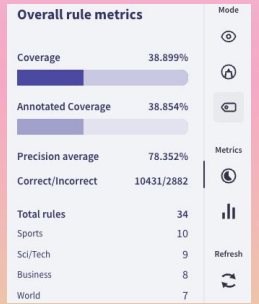
OPTIMISING HUMAN LABELLING WITH WEAK SUPERVISION AND ACTIVE LEARNING

WEAK SUPERVISION

- We wrote **34 labelling functions (LFs)** to create weak labels → Mainly **Pattern Matching LFs**
  - **4 classes:** World, Sports, Sci/Tech and Business
- **Rubrix UI** was very helpful to check the performance of the LFs on the validation set*
- We denoised the weak labels with **Snorkel's label model** to obtain probabilistic labels for the final weakly supervised dataset





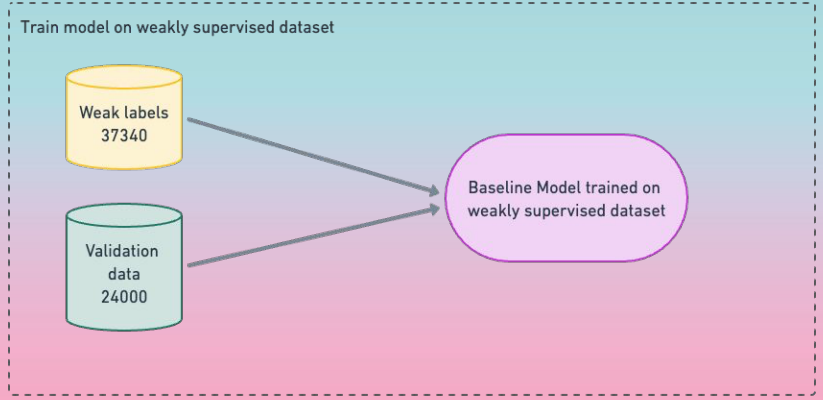


- 40% coverage
- 78% avg precision on validation set

**We used 20% of the original AG news dataset as validation set to simulate a real world scenario where this data would have been human-labelled*

ACTIVE LEARNING LOOP

- Model trained on **weakly supervised dataset → No need to label a training dataset** to initialise the active learning loop



- **Active learning** allows us to identify data points that **are confusing the model (uncertainty sampling)** or **are novel to the model (diversity sampling),** thus **optimising the amount of data that needs to be human-labelled**

Uncertainty Sampling Comparison

Least Confidence

- Picks samples where the **most likely class has the lowest probabilities**
- **Benefits**: very simple and intuitive
- **Shortcomings**: this method is **not very robust**, and doesn't provide the best signal for uncertainty

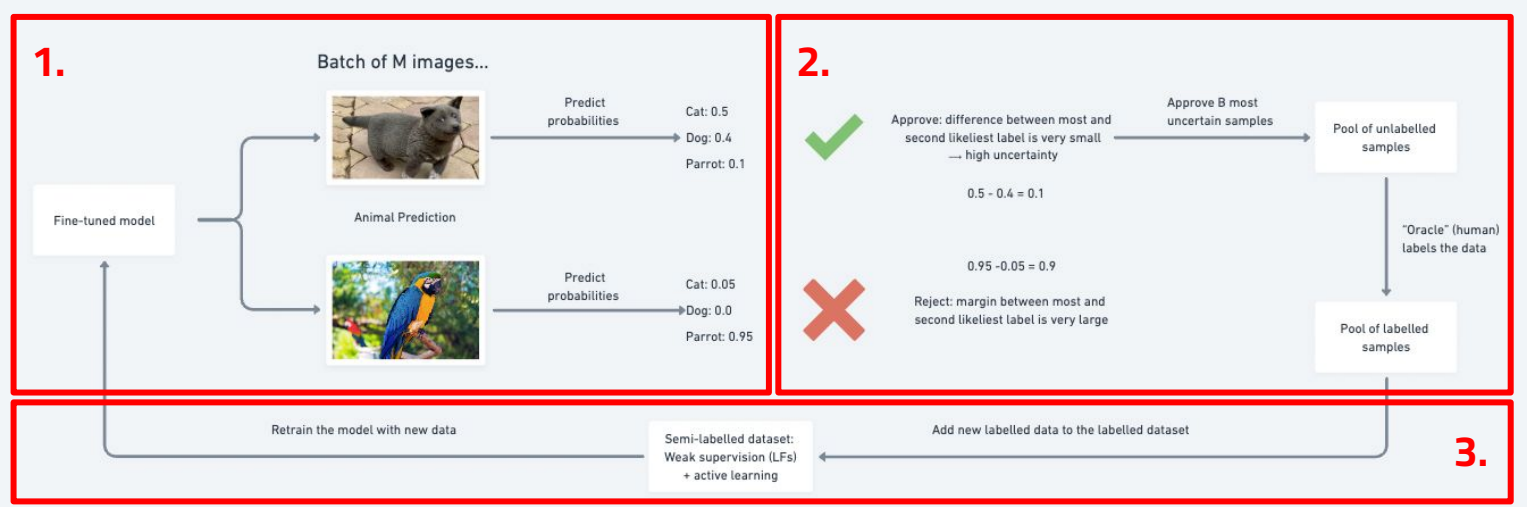
Margin Sampling

- Picks samples where the **two most likely classes have similar probabilities**
- **Benefits**: also **simple** and **more robust** than least confidence
- **Shortcomings**: **still only uses the two most probable classes**, other classes don't have an effect on uncertainty

Entropy Sampling

- **Averages the uncertainty** across all the classes
- **Benefits**: **robust**, averages the uncertainty throughout all the classes
- **Shortcomings**: **math** behind entropy sampling is a bit **more involved** compared to the other two methods

Active Learning Loop with Margin Sampling



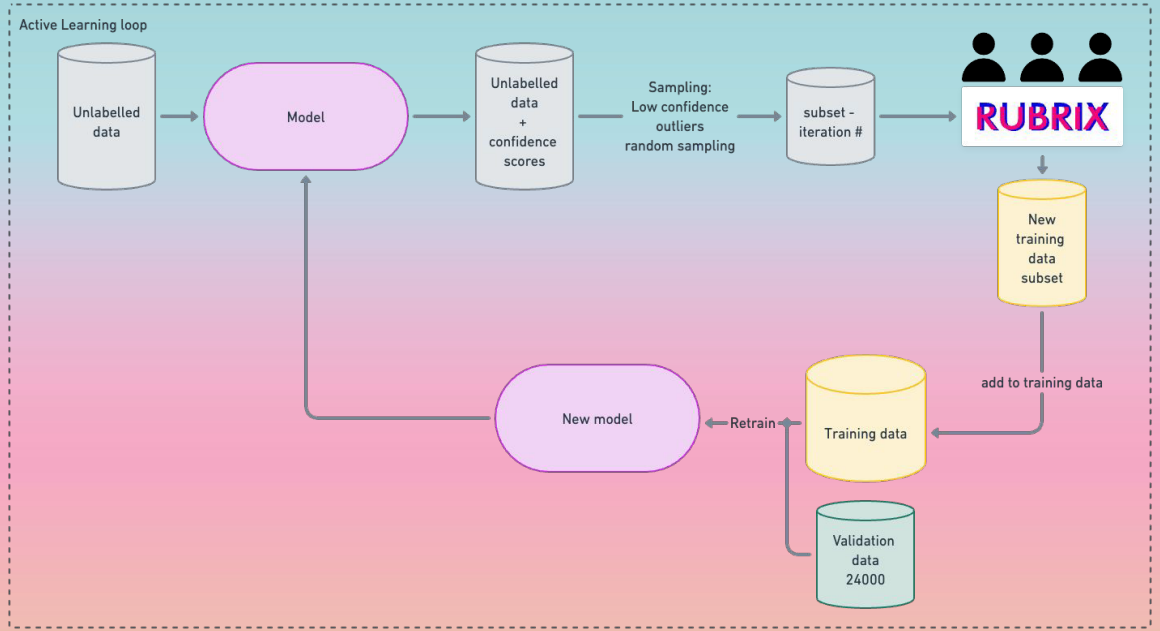
Step 1: Take a pool of M images, use the model to predict the probabilities for the images

Step 2: Calculate uncertainty scores for all n images, choose B most uncertain samples for human labeling

Step 3: Add newly labelled samples to your training set and retrain your model

ACTIVE LEARNING LOOP - IMPLEMENTATION

1. Model is used to **get confidence scores** from the pool of unlabeled data
2. A new **subset of X samples** is created based on **uncertainty sampling**: **Margin sampling**
3. The **new subset** is **human-labeled**
4. The new subset is **added to the training set** and the **model retrained** to obtain and new model
5. The new model is now used to **get confidence scores** and **select samples** from the remaining unlabeled data



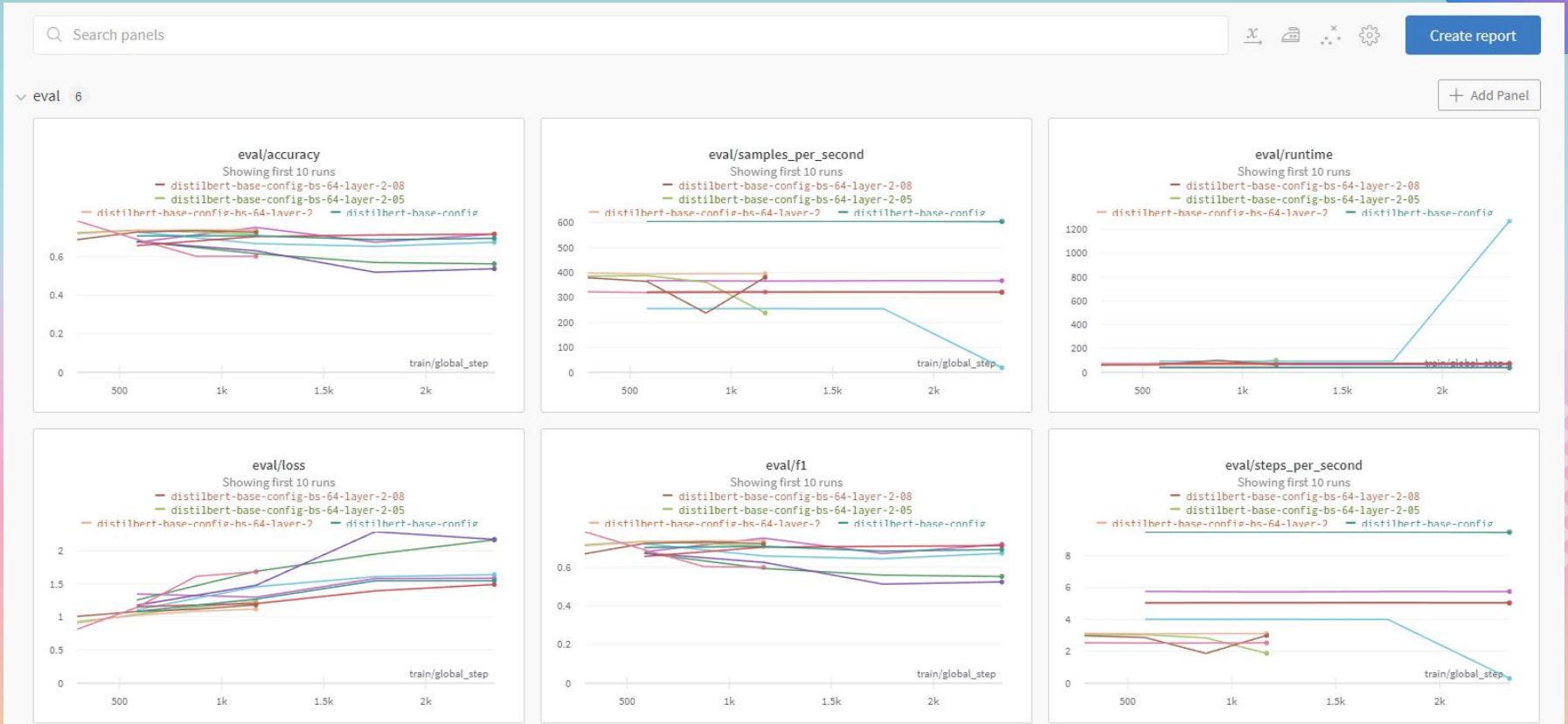
04.

MODEL TRAINING

Model Training Loop

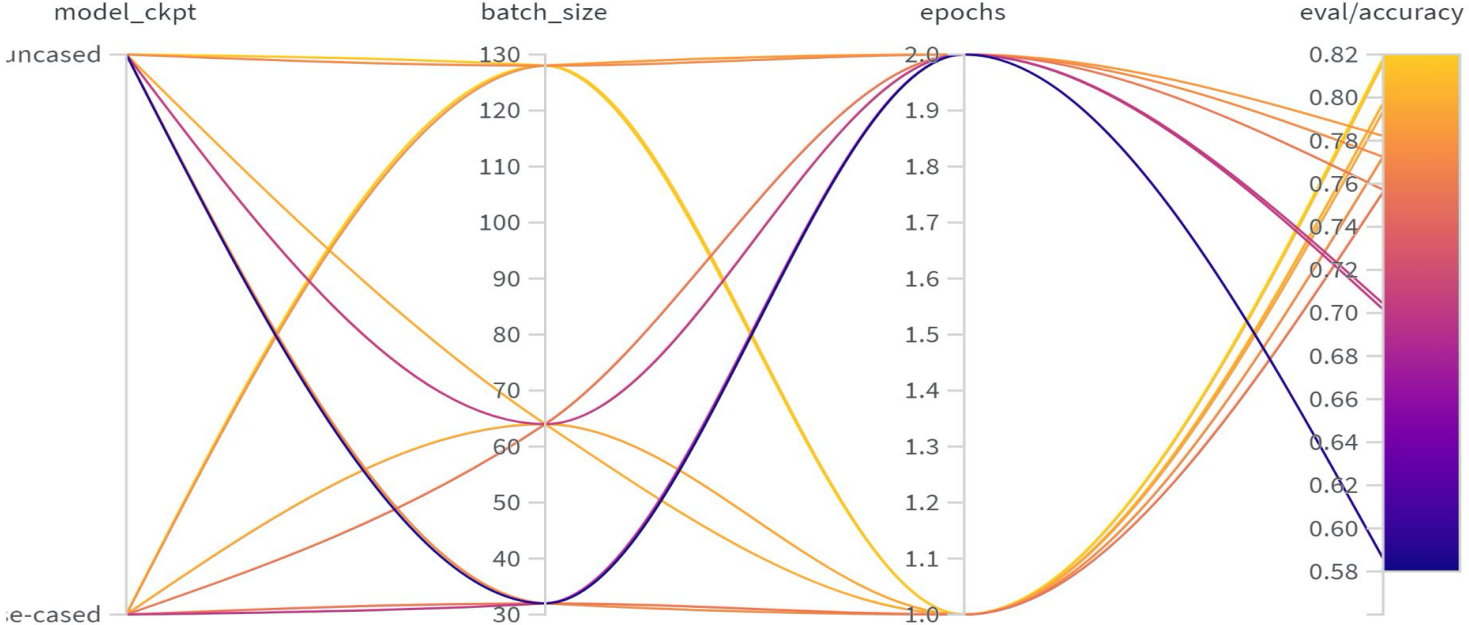
- **Natural Language Processing Text MultiClass Classifier**
- Loaded data with weak labels hosted on 🤗 Datasets, generated using Rubrix
- The data is **tokenized** using the **subword tokenization** technique.
- We load the model using `.from_pretrained` passing the name of the model as a parameter which loads from 🤗Models.
- Each run is logged into Weights & Biases, where we can compare different runs and their accuracy.
- Further W&B Sweeps are also implemented to automate hyperparameter search along with interactive dashboards.

WandB Individual Runs Combined



https://wandb.ai/kushalramaiya/huggingface?workspace=user-diegoquintana

Weight And Biases Sweep (Example Run)



https://wandb.ai/team_44/huggingface/sweeps/phsok7we?workspace=user-diegoquintana

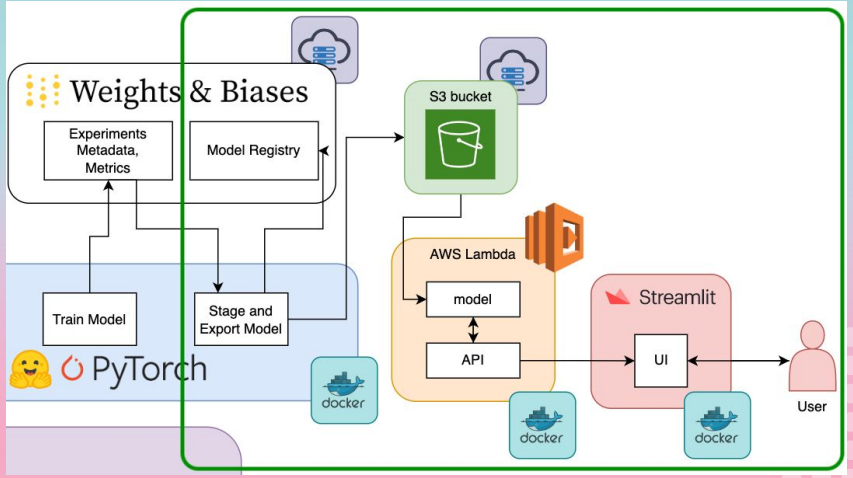
05.

MODEL DEPLOYMENT

HOW THE MODEL WAS DEPLOYED AND
DEPLOYMENT STRATEGY

Model deployment

- The **Weight and Biases model registry** is used to register models and flag the one that is ‘production-ready’
- The model selected for deployment is converted into **TorchScript** and uploaded to **S3 Bucket**
- The model inference runs in a **AWS Lambda** function
- User can query the model using a **Streamlit** app. This send requests to the model via the AWS Lambda API endpoint.



User Interface

- Streamlit link

Model deployment

- A model is stored in the **W&B model registry** with the label **prod**
- After a new model is trained, its accuracy **("eval/f1") score is compared against the prod model on the test set***
- **If the score is higher** than the production model, the **challenger model is registered as prod**
- A **traced version** of the model and **its metadata** is uploaded to the S3 bucket
- The **lambda container is rebuilt** so that it fetches the new model

**We used the test set of the original AG news dataset to simulate a real world scenario where this data would have been human-labelled*

06

FURTHER WORK

HOW TO IMPROVE IT?

FURTHER WORK

- **Weak supervision** → More comprehensive EDA to build more domain knowledge and create more sophisticated Labeling Functions (LFs)
- **Active learning strategy** → Use Entropy Sampling instead of Margin Sampling or combine different active learning strategies. E.g. 70% margin sampling + 20% Outliers + 10% Random sampling
- **Model training and deployment** → Automate model retraining and deployment after each iteration of the active learning loop
- **Model monitoring** → Come up with relevant projections for monitoring the performance of the news classifier and push them to Gantry
- **Product & UI/UX** → Build a simple product were the model could be useful for people

THANKS!

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**