

# E-commerce Product Classification

Yuqing Zhang

# Project Description

## Objective:

1. classify e-commerce products into 27 categories to achieve a high classification accuracy
2. Build a user interface for the image repository

## Data:



for each product:

Training set: ~20,000 (5-fold Cross-Validation)

Test set: ~20,000

Y: category

X: 1. categorical features 2. a noisy text description 3. a noisy image

id	category	gender	baseColour	season	usage	noisyTextDescription	
10008	Topwear	Women	Black	Fall	Casual	United Colors of Benetton Women Solid Organizer Etra	
10013	Bags	Women	Black	Winter	Casual	Murcia Women Casual Black Handbag	

output                      categorical                      text                      image

# High Level

**Models:** I trained those 4 base models independently using the categorical/image/text data:

1. categorical data: xgboost
2. image data: resnet18
3. text data: fast-text
4. text data: lstm

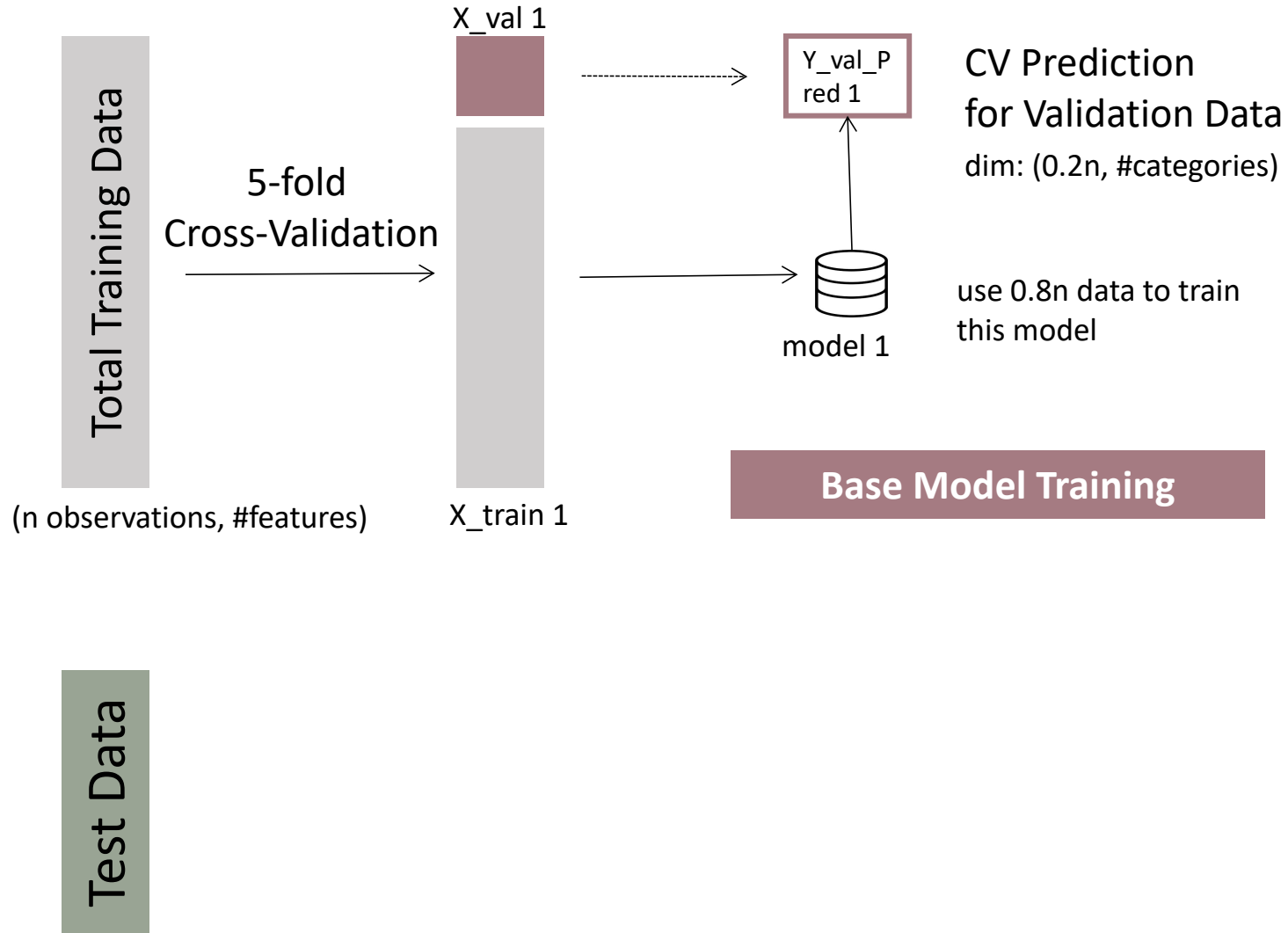
Each model will output a 27-dim vector of probabilities

**Challenge:** How to aggregate the output across the models? Majority Voting does not yield the best results

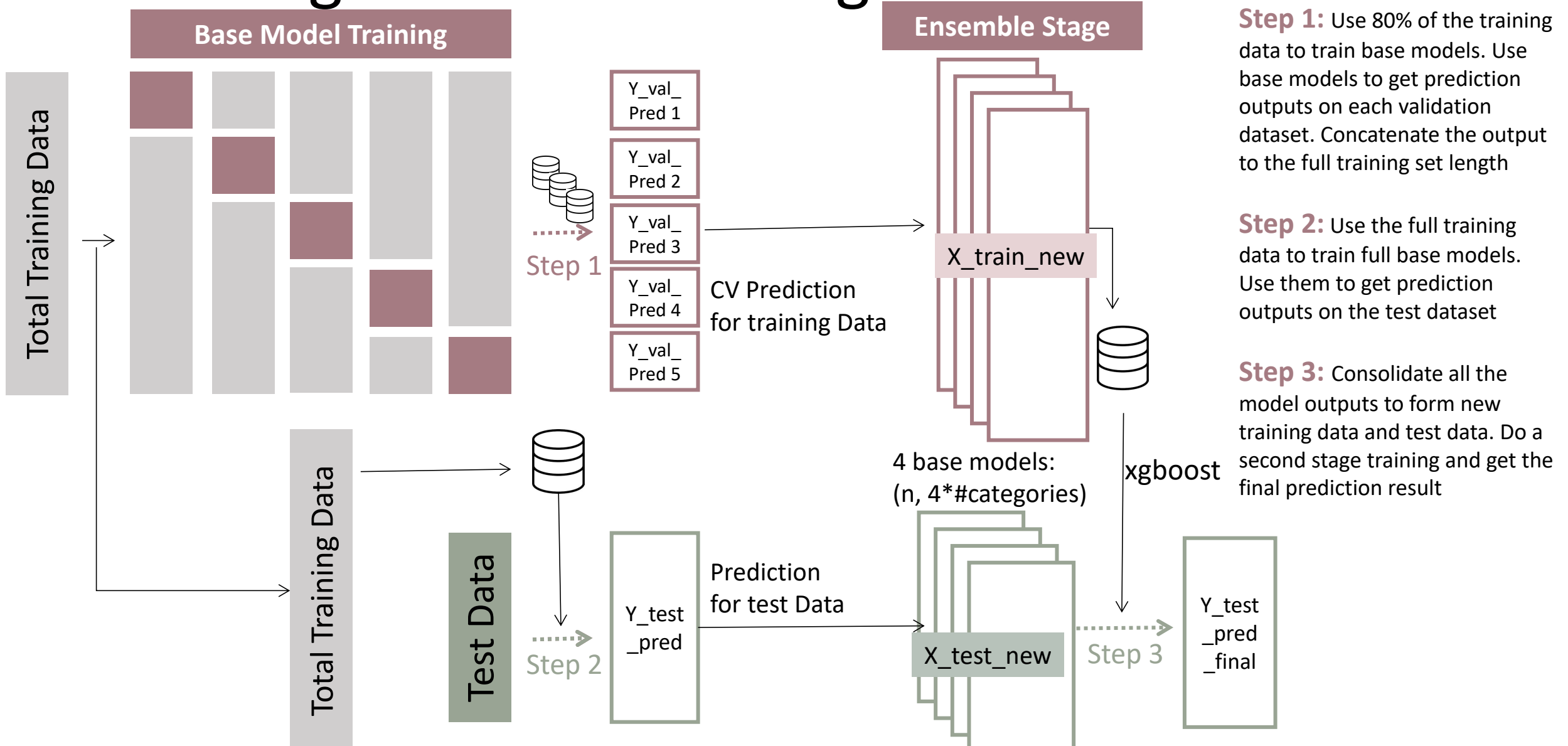
**Training Process:** stacking for two-stage training

**Result:** The final prediction accuracy is around 98%

# Training Process: Stacking



# Training Process: Stacking



# Data Preprocessing

1. **Categorical Features:** factorize to numerical

2. **Text Description:**

- remove symbols, stopwords; make them lowercase
- vectorize text by turning each text into a sequence of integers
- limit the word set to the top 50,000 words
- set the max number of words in each description at 20

3. **Image:**

- read pixel 60 \* 80
- pad zeros around the image
- randomly crop the image to 48 \* 48
- randomly rotate horizontally
- normalize each R,G,B channel to mean 0.5 and std 0.5

# Model

## Categorical Data: xgboost

- Boosting: increase the weight of misclassified items
- Gradient-boosted decision tree

## Image Data: resnet18 (CNN)

- Residual connection: skip layers to avoid gradient vanishing
- Use pretrained model to improve the effectiveness

## Text Data: LSTM (RNN)

- Long short-term memory: Use gates to obtain feedback connections in the cells.
- This helps LSTM to learn long term dependencies.

## Text Data: Fast-Text

- Hierarchical softmax: the classes are arranged in a tree distribution instead of a flat, list-like structure.
- Depth-First Search
- n-grams representation

# An Image Repository User Interface

An image repository for e-commerce products using PyQt5

There are two kinds of uses:

1. *Search*: given the input (categorical data or text description), search for the images in the repository.
2. *Classification*: given a new image, classify it to the corresponding category, and find similar products in the same category.



# Illustrative Results

- [Link to Github page](#)
- PyQt5
- Qt Designer

# Improvement...

- Build more features, like word count in each text description
- For the classification, get the confusion matrix to see which categories are misclassified with each other
- Use database to build the image repo...

Thank you :)

# Model - categorical data: xgboost

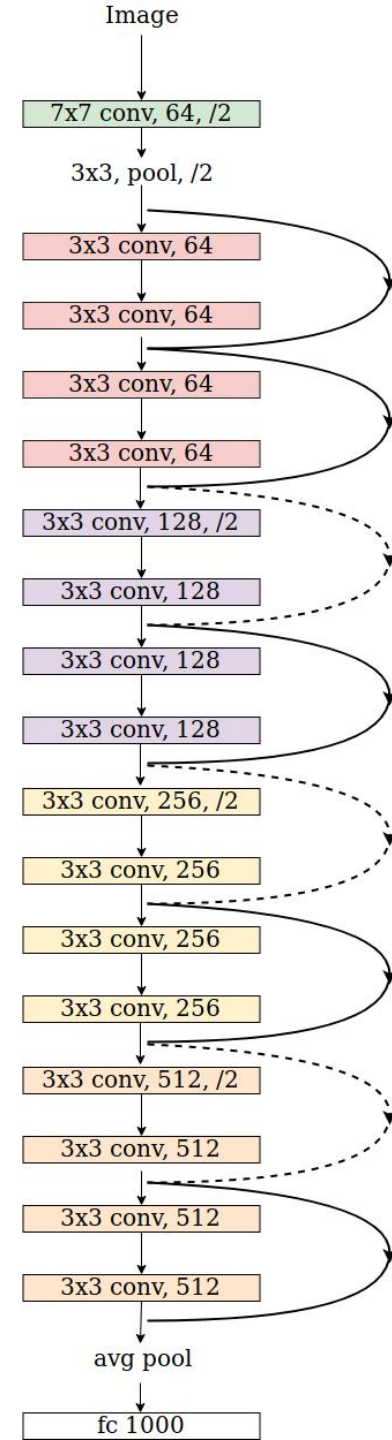
Boosting: when an instance is misclassified by a hypothesis, increase its weight so that the next hypothesis is more likely to classify it correctly

eXtreme Gradient Boosting: gradient-boosted decision tree

Implemented through sklearn in Python

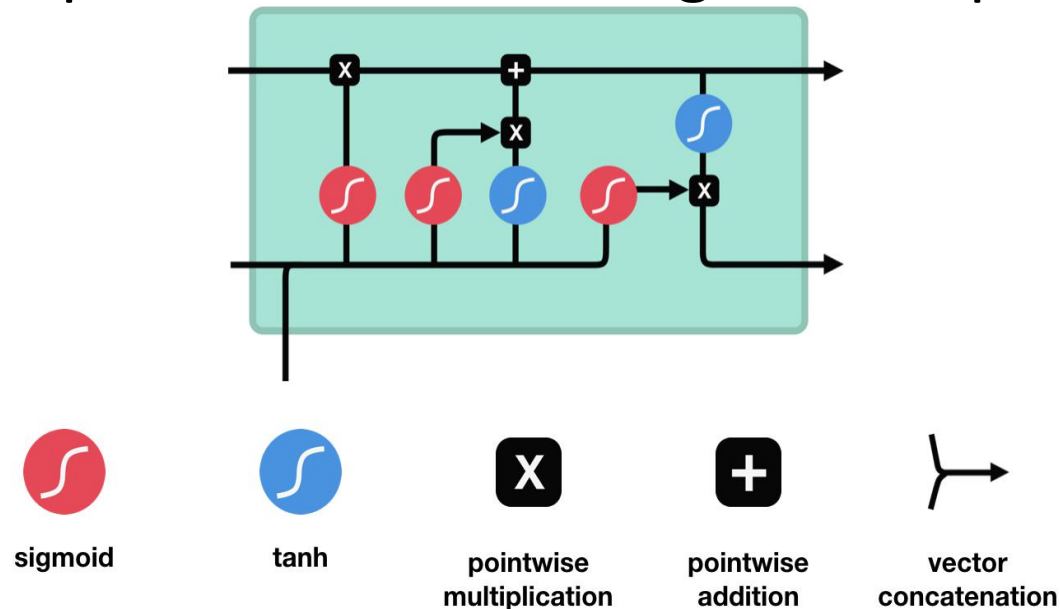
# Model - image data: resnet18

- CNN: alternative convolution and pooling layer
- Residual connection: skip layers to avoid gradient vanishing
- Use pretrained model to improve the effectiveness
- Scheduler: Cosine Annealing with Restart
- Loss: Cross Entropy Loss
- Optimizer: SGD



# Model - text data: LSTM

- RNN: outputs can be fed back to the network as inputs
- Variable length data
- Long short-term memory (LSTM): Use **gates** to obtain feedback connections in the cells. This helps LSTM to learn long term dependencies.



# Model - text data: Fast-Text

- Hierarchical softmax: a loss function that approximates the softmax with a much faster computation time.
- Advantages: there are a large number of categories and there is a class imbalance present in the data. Here, the classes are arranged in a tree distribution instead of a flat, list-like structure.
- Depth-First Search speeds up the classification process significantly.
- n-grams representation: the concatenation any n consecutive tokens.