IEMS 5780

Question 1:

* Machine learning application
* Classification task
  + Determine whether a customer will borrow money from the bank
* Types of data
  + Customers’ demographic data (age, education level, job nature, etc.)
  + Historical transactions of the customers
  + Different information about the products in the bank
  + ..
* How to prepare the data
  + We need to collect data about the customers that we are going to apply the model
  + Prepare the data, such that we have the X (features) and y (labels → 0=not borrow, 1=borrow)
  + Split the data into training and test dataset (validation dataset) → evaluate
* Representation of the customers:
  + Gender, age, education level
  + Whether he/she has borrowed money before / if he/she has already repaid the money
  + Average spending of the customer every month (compared to his monthly income)
* Input: feature vector (features of the customer, but can also contain other features: period/season of the year, general economic environment, etc.)
* Output:
  + Probability of whether a customer will borrow money
  + Or it can just be 0 or 1
* You conduct experiments using train/test split
  + Metrics: accuracy, precision and recall, AUC, etc.
* Additional data:
  + GDP, stock market performance, number of new companies registered

Question 2

* Text classification
* Pre-defined categories
* Multi-class classification (not only 1 or 0)
* Whether each comment can only be assigned to one category
  + Single-label vs. multi-label
* Information that you can extract from the small dataset
  + Languages (e.g. only English, mostly English with some Chinese? mixed?)
  + Average length of the comments
  + Whether users use emoji
  + Whether there some abusive comments
* Preprocessing
  + Tokenization
  + Normalization of the texts (lowercasing, truecasing, etc.)
  + Remove unwanted characters or words
  + Stemming
* Representation
  + Bag-of-words
  + Vector space model
  + Tf-idf
* Two ML models:
  + Logistic regression (linear combination of the weights and input)
  + Naive Bayes (probability)
  + Neural network / random forest
* Accuracy / Precision / Recall
  + Example: when the target class has a very small probability in the dataset, then a high accuracy may not be good enough, we should look into precision and recall

Question 3

* Scenario:
  + We have user reviews
  + We DON’T have user ratings (we don’t know who like what explicitly)
  + Target: whether user thinks POS or NEG about a book
  + Suggested solution:
    - Classification of the user reviews → POS or NEG (can also consider NEUTRAL class)
    - How to come up with this model?
      * We ask people to label of the review data
      * Use some external data set with labels to train model (sentiment analysis) (domain might be different, risk that the model does not generalize to your own data)
* Two algorithms:
  + User-based / item-based collaborative filtering
  + Matrix factorization
* How to evaluate the performance of the model
  + Regression task
  + Mean absolute error (MAE) or RMSE
  + MAE: |y - y’|, RMSE
* How to generate recommended books for new users
  + Cold start problem
  + Recommend the most popular books / the newest books
  + Ask the user actively to rate 10 books first before they use the app

Question 4

(d)

* All programs are the same (will produce identical output if given the same input)
* They run in parallel
* Load balancing machine
  + Centralized point to get requests from the clients
  + It will distribute requests to all the servers evenly
* Put every requests into a queue
  + Consume the requests from the queue (a server will only handle a request when it is idle)

(e)

* Global interpreter Lock (GIL) in Python
* A single process: only one thread can be executed at a time (or use the CPU)
* → use multiprocessing
* CPU-intensive or CPU-bound task
* Parallelism → speed up processing the data

(f)

* Asynchronous programming
* It tries to switch to another task, if the current task cannot be continued for now (e.g. it’s waiting for loading from the network, reading file from harddisk, etc)
* Example: server that processes requests from a large number of clients
* Ref: the asynchronous TCP server (Lecture 9)