

1. Зошто Attention ги подобрува seq2seq моделите?
 - Attention му овозможува на моделот да „внимава“ на најважните зборови од влезната реченица при секој чекор на излезот. Наместо да користи само еден вектор за целата реченица, attention пресметува тежини и извлекува најрелевантни информации. Ова го подобрува преводот, резимирањето и другите NLP задачи, особено кога влезовите се долги.
2. Што е Random Forest и кои се неговите предности?
 - Random Forest е ансамбл модел составен од повеќе дрва на одлука. Користи bagging (семплирање со враќање) и избира случајно подмножество на features за секое дрво. Тоа води до стабилен, помалку overfit-ирачки модел. Во споредба со едно Decision Tree, Random Forest подобро генерализира, е поробустен и автоматски ја мери важноста на features.
3. Класификација на вести со LSTM (NLP задача)
 - Ова е мултикласна NLP класификација.
 - Чекори:
 - Претпроцесирање: чистење, токенизација, embeddings.
 - Модел: Embedding \rightarrow LSTM \rightarrow Dense (softmax).
 - Тренирање: cross-entropy loss, Adam optimizer.
 - Overfitting: Dropout, early stopping.
4. Хиперпараметри во XGBoost:
 - **learning_rate**: колку брзо учи.
 - **max_depth**: колку длабоко се градат дрвата.
 - **n_estimators**: број на дрва.
 - **subsample**, **colsample_bytree**: контрола на податоци/фичери по дрво.
 - **gamma**, **lambda**, **alpha**: regularization параметри.
5. Разлика помеѓу RNN и LSTM:
 - RNN користи повторувачка структура за обработка на секвенци, но има проблем со долгорочна меморија (vanishing gradients). LSTM го решава тоа со посебни „врати“ (gates) кои контролираат што се памети и што се заборува. Затоа LSTM е подобар за подолги секвенци.
6. Што е точно за Feed Forward Neural Network:
 - FFNN е најосновен тип на невронска мрежа каде податоците се движат само во една насока – од влез кон излез, без циклуси. Се користи за задачи како класификација или регресија и добро се справува со поедноставни или статички податоци.

7. Како да определам кластеринг според слика:
 - Гледаш дали кластерите имаат неправилна форма (тогаш DBSCAN/HDBSCAN се подобри), или се јасно одделени и кружни (K-means е соодветен). Ако има outliers, DBSCAN е подобар бидејќи ги игнорира.
8. Кој модел да се избере за болница со 1000 статии и медицински зборови?
 - Најдобар избор е fine-tuned Clinical BERT, бидејќи е веќе трениран на медицински текстови и бара помалку податоци за да постигне добри резултати.
9. Што е точно за lags?
 - Lag features ги користат вредностите од минатото (n чекори наназад) за да се предвиди сегашноста или иднината. Клучни се за временски серии.
10. Што значи `return_sequences=True` во LSTM?
 - Ова значи дека LSTM ќе врати **целата секвенца од скриени состојби** за секој временски чекор, наместо само последната. Ова е важно кога следниот слој (на пример друг LSTM или TimeDistributed Dense) има потреба од **излез за секој чекор во времето**.
11. Имаш 100000 позитивни и 500 негативни примери, моделот ги промашува негативните – што правиш?
 - Ова е класичен имбалансирен сет. Може да се примени undersampling на позитивните или oversampling на негативните, или користење на специфични метрики како recall/F1 score.
12. Тренирање на модел со финансиски и временски податоци од берза:
 - Најдобар избор е FastModel трениран на доменски податоци, бидејќи ги разбира финансиските патерни. Временските податоци бараат модели кои ја разбираат секвенцата, како LSTM, но предтренираниот доменски модел е побрз и поточен.
13. Како DBSCAN се справува со overfitting наспроти K-means:
 - DBSCAN не бара број на кластери однапред и е отпорен на outliers. Не се води од форма или големина. Пример: ако имаш географски податоци со групи во различна густина, DBSCAN ќе најде кластери подобро од K-means, кој би ги поделил неправилно.
14. Зошто се користат длабоки (deep) мрежи наместо плитки (shallow)?
 - Длабоките мрежи можат да научат сложени, хиерархиски патерни и апстракции кои плитките мрежи не можат. Иако теориски една слој мрежа може да апроксимира функции, во пракса длабоките мрежи се поефикасни и поизразити.

15. Трансформери – како ги енкодираат зборовите
- Трансформерите ги енкодираат зборовите преку embedding слој кој секој збор го претставува како вектор.
 - Потоа, се додава positional encoding за да се задржи редоследот на зборовите, бидејќи трансформерите немаат рекурентна структура.
 - Овие вектори се обработуваат преку self-attention механизми кои откриваат кои зборови се важни за интерпретација на другите во контекстот.
16. One Hot Encoder – предности и слабости
- Предности: едноставна имплементација, не воведува редослед меѓу категории.
 - Слабости: неефикасна за голем број категории (curse of dimensionality), не пренесува семантичка блискост меѓу категории.
17. Seq2Seq – позитивни и негативни страни
- Позитивни: добар за задачи со различна должина на влез и излез (превод, резиме), ја задржува секвенцијалната структура.
 - Негативни: класичниот seq2seq (без attention) не памти долго, performance опаѓа кај долги секвенци.
18. Разлика помеѓу Boosting и Bagging – кој подобро се справува со overfit?
- Bagging (пример: Random Forest) го намалува overfitting со усреднување (averaging) на независни модели.
 - Boosting (пример: XGBoost) гради модели последователно, подобро се справува со bias, но може полесно да доведе до overfit.
 - Значи: Bagging подобро за overfit, Boosting за underfit.
19. Кој алгоритам креира празни кластери и зошто?
- K-Means може да креира празни кластери ако два центроида се иницијализираат блиску еден до друг и не "фатат" доволно блиски точки при некоја итерација.
 - Алгоритмот тогаш не добива нови точки во тој кластер и тој останува празен.
20. Еднонасочна vs двонасочна RNN – и дали може да има длабока RNN
- Еднонасочна RNN гледа само нанапред во секвенцата, додека двонасочна RNN ги чита податоците и напред и назад.
 - Да, може. Длабока RNN има повеќе слоеви еден над друг, што ѝ овозможува да учи покомплексни шаблони.
21. Label Encoder – предности и слабости
- Предности: компактно претставување, лесен за имплементација.
 - Слабости: внесува лажен редослед (броеви), што може да биде проблем кај не-нумерички категории.

22. Алгоритам за кластерирање со проблем на локален оптимум
- K-Means често запаѓа во локален оптимум поради случајна иницијализација на центроидите. Ако тие се лошо поставени на почетокот, крајниот резултат може да биде некавалитетен.
23. Gini index vs Information Gain
- Gini index е побрз за пресметка и често се користи во Random Forest.
 - Information Gain (базиран на ентропија) дава подобра информациска вредност но е побавен.
24. Сличности и разлики – Seq2Seq и Transformers
- Сличности: двата модели работат со влез/излез секвенции и имаат encoder-decoder структура.
 - Разлики: Seq2Seq (LSTM/RNN) е рекурентен, трансформерите се базирани на attention и се попаралелизирани, што ги прави побрзи и подобри за долги секвенци.
25. Агломеративно vs KNN (кластерирање)
- Агломеративно е хиерархиско – гради кластери од долу нагоре. KNN не е кластерирање, туку класификациски алгоритам кој предвидува класа базирано на блиски соседи.
26. Кога хиерархиско кластерирање дава лоши перформанси?
- Кога има многу податоци или кога скалите на карактеристиките се многу различни. Без нормализација, алгоритмот ќе дава приоритет на атрибути со поголеми вредности.
27. FFNN кај трансформери – за што служи?
- Feed-Forward Neural Network кај трансформери се користи после attention блокот за да се трансформираат и збогатат добиените вектори. Се применува посебно на секој збор.
28. DBSCAN vs K-Means – outliers
- DBSCAN природно ги открива outliers бидејќи не ги смета точките што не припаѓаат на густ регион.
 - K-Means мора насилно да класифицира сè, па и outliers се доделуваат на некој кластер – што често ги расипува резултатите.
29. Зошто нормализација е важна за хиерархиско кластерирање?
- Хиерархиското кластерирање користи растојанија (на пр. Euclidean), па ако карактеристиките се на различна скала (нпр. возраст 0-100, приход 0-100000), ќе се фаворизираат големите броеви.
 - Пример: алгоритмот ќе кластеризира само по приход ако не се нормализираат податоците.

30. За даден модел е дадена информацијата дека има "ассигасу" од 95%. Опишете ги поребните предуслови за да можете да кажете дека моделот не е добар.
- Еден предуслов за ова е ако датасетот е небалансиран. На пример ако се работи за нумерички податоци, ако голем дел од податоците се позитивни, моделот при секое тестирање ќе заклучува дека "classifiers" ќе бидат позитивни и со тоа не може да се заклучи дали навистина е добар моделот. Исто така во ассигасу не спаѓаат false positive примероците (тие кои биле точни а сме рекле дека се грешни), а тие податоци што фалат можат да значат многу за моделот.
31. Како вработен во финтек компанија добивате задача да ги оцените вестите дали се спортски, политички, финансиски, економски или забавни. Притоа, добивате насоки да користите LSTM невронски мрежи за да го изградите моделот. За каков тип на NLP таск (задача) станува збор?
- Наведете ги сите потребни претпроцесиращки чекори, архитектурата на моделот кој што би го користеле, процесот на тренирање и евалуација, како и техника со која би го избегнале overfitting.
- Станува збор за мултикласна класификација на текст (Multi-class Text Classification)
 - Претпроцесирање: Чистење, токенизација, embeddings, padding, label encoding.
 - Модел: Embedding → LSTM → Dropout → Dense(softmax).
 - Тренирање: Crossentropy loss, Adam, метрики: Accuracy + F1.
 - Anti-overfitting: Dropout, Early Stopping, Regularization.
32. Накратко објаснете ја мерката R-squared, кога и како би ја користеле.
- R-squared мери колку добро моделот ја објаснува варијацијата во податоците.
 - Се користи кај регресиони модели (линеарна/логистичка регресија) и кога сакаме да измериме колку добро моделот се вклопува во податоците
 - Се движи од 0-1
 - $R^2 = 1$ → моделот совршено ја објаснува варијацијата
 - $R^2 = 0$ → моделот не ја објаснува варијацијата
 - $R^2 < 0$ → моделот е полош од предвидување со средна вредност