Predictive Modeling (19 questions)

1. (Given a Dataset) Analyze this dataset and give me a model that can predict this response variable.

- Problem Determination -> Data Cleaning -> Feature Engineering -> Modeling
- Benchmark Models
    - Linear Regression (Ridge or Lasso) for regression
    - Logistic Regression for Classification
- Advanced Models
    - Random Forest, Boosting Trees, and so on
        - Scikit-Learn, XGBoost, LightGBM, CatBoost
- Determine if the problem is classification or regression
- Plot and visualize the data.
- Start by fitting a simple model (multivariate regression, logistic regression), do some feature engineering accordingly, and then try some complicated models. Always split the dataset into train, validation, test dataset and use cross validation to check their performance.
- Favor simple models that run quickly and you can easily explain.
- Mention cross validation as a means to evaluate the model.

2. What could be some issues if the distribution of the test data is significantly different than the distribution of the training data?

- The model that has high training accuracy might have low test accuracy. Without further knowledge, it is hard to know which dataset represents the population data and thus the generalizability of the algorithm is hard to measure. This should be mitigated by repeated splitting of train vs. test dataset (as in cross validation).
- When there is a change in data distribution, this is called the dataset shift. If the train and test data has a different distribution, then the classifier would likely overfit to the train data.
- This issue can be overcome by using a more general learning method.
- This can occur when:
    - P(y|x) are the same but P(x) are different. (covariate shift)
    - P(y|x) are different. (concept shift)
- The causes can be:
    - Training samples are obtained in a biased way. (sample selection bias)
    - Train is different from test because of temporal, spatial changes. (non-stationary environments)
- Solution to covariate shift
    - importance weighted cv

3. What are some ways I can make my model more robust to outliers?

- We can have regularization such as L1 or L2 to reduce variance (increase bias).
- Changes to the algorithm:
    - Use tree-based methods instead of regression methods as they are more resistant to outliers. For statistical tests, use non parametric tests instead of parametric ones.
    - Use robust error metrics such as MAE or Huber Loss instead of MSE.
- Changes to the data:
    - Winsorizing the data
    - Transforming the data (e.g. log)
    - Remove them only if you’re certain they’re anomalies and not worth predicting

4. What are some differences you would expect in a model that minimizes squared error, versus a model that minimizes absolute error? In which cases would each error metric be appropriate?

- MSE is more strict to having outliers. MAE is more robust in that sense, but is harder to fit the model for because it cannot be numerically optimized. So when there are less variability in the model and the model is computationally easy to fit, we should use MAE, and if that's not the case, we should use MSE.
- MSE: easier to compute the gradient, MAE: linear programming needed to compute the gradient
- MAE more robust to outliers. If the consequences of large errors are great, use MSE
- MSE corresponds to maximizing likelihood of Gaussian random variables

5. What error metric would you use to evaluate how good a binary classifier is? What if the classes are imbalanced? What if there are more than 2 groups?

- Accuracy: proportion of instances you predict correctly.
    - Pros: intuitive, easy to explain
    - Cons: works poorly when the class labels are imbalanced and the signal from the data is weak
- ROC curve and AUC: plot false-positive-rate (fpr) on the x axis and true-positive-rate (tpr) on the y axis for different threshold. Given a random positive instance and a random negative instance, the AUC is the probability that you can identify who's who.
    - Pros: Works well when testing the ability of distinguishing the two classes.
    - Cons: can't interpret predictions as probabilities (because AUC is determined by rankings), so can't explain the uncertainty of the model, and it doesn't work for multi-class case.
- logloss/deviance/cross entropy:
    - Pros: error metric based on probabilities
    - Cons: very sensitive to false positives, negatives
- When there are more than 2 groups, we can have k binary classifications and add them up for logloss. Some metrics like AUC is only applicable in the binary case.

6. What are various ways to predict a binary response variable? Can you compare two of them and tell me when one would be more appropriate? What's the difference between these? (SVM, Logistic Regression, Naive Bayes, Decision Tree, etc.)

- Things to look at: N, P, linearly separable, features independent, likely to overfit, speed, performance, memory usage and so on.
- Logistic Regression
    - features roughly linear, problem roughly linearly separable
    - robust to noise, use l1,l2 regularization for model selection, avoid overfitting
    - the output come as probabilities
    - efficient and the computation can be distributed
    - can be used as a baseline for other algorithms
    - (-) can hardly handle categorical features
- SVM
    - with a nonlinear kernel, can deal with problems that are not linearly separable
    - (-) slow to train, for most industry scale applications, not really efficient
- Naive Bayes
    - computationally efficient when P is large by alleviating the curse of dimensionality
    - works surprisingly well for some cases even if the condition doesn't hold
    - with word frequencies as features, the independence assumption can be seen reasonable. So the algorithm can be used in text categorization
    - (-) conditional independence of every other feature should be met
- Tree Ensembles

  - good for large N and large P, can deal with categorical features very well
  - non parametric, so no need to worry about outliers
  - GBT’s work better but the parameters are harder to tune
  - RF works out of the box, but usually performs worse than GBT
- Deep Learning
  - works well for some classification tasks (e.g. image)
  - used to squeeze something out of the problem

7. What is regularization and where might it be helpful? What is an example of using regularization in a model?

- Regularization is useful for reducing variance in the model, meaning avoiding overfitting.
- For example, we can use L1 regularization in Lasso regression to penalize large coefficients and automatically select features, or we can also use L2 regularization for Ridge regression to penalize the feature coefficients.

8. Why might it be preferable to include fewer predictors over many?

- When we add irrelevant features, it increases model's tendency to overfit because those features introduce more noise. When two variables are correlated, they might be harder to interpret in case of regression, etc.
- curse of dimensionality
- adding random noise makes the model more complicated but useless
- computational cost
- Ask someone for more details.

9. Given training data on tweets and their retweets, how would you predict the number of retweets of a given tweet after 7 days after only observing 2 days worth of data?

- Build a time series model with the training data with a seven day cycle and then use that for a new data with only 2 days data.
- Ask someone for more details.
- Build a regression function to estimate the number of retweets as a function of time t
- to determine if one regression function can be built, see if there are clusters in terms of the trends in the number of retweets
- if not, we have to add features to the regression function
- features + # of retweets on the first and the second day -> predict the seventh day
- https://en.wikipedia.org/wiki/Dynamic_time_warping

10. How could you collect and analyze data to use social media to predict the weather?

- We can collect social media data using twitter, Facebook, instagram API’s.
- Then, for example, for twitter, we can construct features from each tweet, e.g. the tweeted date, number of favorites, retweets, and of course, the features created from the tweeted content itself.
- Then use a multivariate time series model to predict the weather.
- Ask someone for more details.

11. How would you construct a feed to show relevant content for a site that involves user interactions with items?

- We can do so using building a recommendation engine.
- The easiest we can do is to show contents that are popular other users, which is still a valid strategy if for example the contents are news articles.

- To be more accurate, we can build a content based filtering or collaborative filtering. If there's enough user usage data, we can try collaborative filtering and recommend contents other similar users have consumed. If there isn't, we can recommend similar items based on vectorization of items (content based filtering).

12. How would you design the people you may know feature on LinkedIn or Facebook?

- Find strong unconnected people in weighted connection graph
    - Define similarity as how strong the two people are connected
    - Given a certain feature, we can calculate the similarity based on
        - friend connections (neighbors)
        - Check-in's people being at the same location all the time.
        - same college, workplace
        - Have randomly dropped graphs test the performance of the algorithm
- Ref. News Feed Optimization
    - Affinity score: how close the content creator and the users are
    - Weight: weight for the edge type (comment, like, tag, etc.). Emphasis on features the company wants to promote
    - Time decay: the older the less important

13. How would you predict who someone may want to send a Snapchat or Gmail to?

- for each user, assign a score of how likely someone would send an email to
- the rest is feature engineering:
    - number of past emails, how many responses, the last time they exchanged an email, whether the last email ends with a question mark, features about the other users, etc.
- Ask someone for more details.
- People who someone sent emails the most in the past, conditioning on time decay.

14. How would you suggest to a franchise where to open a new store?

- build a master dataset with local demographic information available for each location.
    - local income levels, proximity to traffic, weather, population density, proximity to other businesses
    - a reference dataset on local, regional, and national macroeconomic conditions (e.g. unemployment, inflation, prime interest rate, etc.)
    - any data on the local franchise owner-operators, to the degree the manager
- identify a set of KPIs acceptable to the management that had requested the analysis concerning the most desirable factors surrounding a franchise
    - quarterly operating profit, ROI, EVA, pay-down rate, etc.
- run econometric models to understand the relative significance of each variable
- run machine learning algorithms to predict the performance of each location candidate

15. In a search engine, given partial data on what the user has typed, how would you predict the user's eventual search query?

- Based on the past frequencies of words shown up given a sequence of words, we can construct conditional probabilities of the set of next sequences of words that can show up (n-gram). The sequences with highest conditional probabilities can show up as top candidates.
- To further improve this algorithm,
    - we can put more weight on past sequences which showed up more recently and near your location to account for trends
    - show your recent searches given partial data

- Personalize and localize the search
    - Use the user's historical search data
    - Use the historical data from the local region

16. Given a database of all previous alumni donations to your university, how would you predict which recent alumni are most likely to donate?

- Based on frequency and amount of donations, graduation year, major, etc, construct a supervised regression (or binary classification) algorithm.

17. You’re Uber and you want to design a heatmap to recommend to drivers where to wait for a passenger. How would you approach this?

- Based on the past pickup location of passengers around the same time of the day, day of the week (month, year), construct
- Ask someone for more details.
- Based on the number of past pickups
    - account for periodicity (seasonal, monthly, weekly, daily, hourly)
    - special events (concerts, festivals, etc.) from tweets

18. How would you build a model to predict a March Madness bracket?

- One vector each for team A and B. Take the difference of the two vectors and use that as an input to predict the probability that team A would win by training the model. Train the models using past tournament data and make a prediction for the new tournament by running the trained model for each round of the tournament
- Some extensions:
    - Experiment with different ways of consolidating the 2 team vectors into one (e.g concantenating, averaging, etc)
    - Consider using a RNN type model that looks at time series data.

19. You want to run a regression to predict the probability of a flight delay, but there are flights with delays of up to 12 hours that are really messing up your model. How can you address this?

- This is equivalent to making the model more robust to outliers.
- See Q3.