① How would you tackle multicollinearity in multiple linear regression?

Multiple linear regression is a method that uses several independent variables to predict or explain the dependent variable we are interested in. When using this technique, we assume that the independent or explanatory variables are also independent from one another (i.e., their value do not affect one another).

Multicollinearity occurs when different independent variables are correlated, and if the correlation between variables is high enough, this can cause problems in fitting the linear regression model.

Ways to tackle multicollinearity:

1. Determine if you actually care about this problem

If you are solely interested in having a good predictor and are satisfied that your model does well on both the training and test sets or if you are below a certain threshold of correlation between variables, you may be able to leave this problem alone.

2. Reduce the number of independent variables

(a) Remove some or most of the correlated variables and re-run your analysis

(b) Use dimensionality reduction technique, such as PCA or a clustering approach, to lower the number of variables.

3. Standardize your independent variables.

(2) How would you interpret coefficients of logistic regression for categorical and boolean variables?

- for boolean variables, we can interpret coeffcients in the following way:
- The magnitude of the coeffcient is directly correlated to its effect on the outcome probability.
- The sign of the coefficients tells you whether the variable is directly or inversely correlated with the outcome probability.

for categorical variables, one-hot encode them to boolean variables & use the above guidelines.

(3) When to use Random Forest over SVM and vice versa?

RF is intrinsically suited for multiclass problems, while SVM is intrinsically two class. for multiclass problem you will need to reduce it into multiple binary classification problems.

- RF works well with a mixture of numerical and categorical features. When features are on various scales, it is also fine. Roughly speaking, with RF, you can use data as it is. SVM maximizes the margin and thus relies on the concept of distance between different points. It is up to you to

decide if "distance" is meaningful. As a consequence, one-hot encoding for categorical features is a must-do. Further, min-max or other scaling is highly recommended at preprocessing step.

- If you have data with n points and m features, an intermediate step in SVM is constructing an nxn matrix. Therefore, SVM is hardly scalable beyond 10^5 points. Large number of features is generally not a problem.
- for a classification problem, RF gives you the probability of belonging to a class. SVM gives you distance to the boundary, you still need to convert it to probability somehow.
- for those problems where SVM applies, it generally performs better than random forest.
- SVM gives you "support vectors", i.e., points in each class closest to the boundary between classes. They may be of interest by themselves for interpretation.

(4) When to use Lasso, Ridge and Elastic Net?

- Lasso and Elastic Net tend to give sparse weights (most zeros), because the L1 regularization ~~drives~~ cares equally about driving down big weights to small weights, or small weights to zero. If you have a lot of predictors (features), and you suspect that not all of them are that important, Lasso and Elastic Net

may be a good idea to start with.

- Ridge tends to give small but well distributed weights, because L2 regularization cares more about driving big weights to small weights, instead of driving small weights to zero. If you only have a few predictors, and you are confident that all of them should be really relevant for predictions, try Ridge as a good regularized linear regression model.
- You will need to scale your data before using these regularized linear regression models.

(5) What is the relation between the log likelihood loss function for logistic regression and maximum likelihood estimation?

In order to chose values for the parameters of logistic regression, we use maximum likelihood estimation. The log likelihood equation is:

$$LL(\theta) = \sum_{i=1}^{n} y^{(i)} \log \sigma\left(\theta^T x^{(i)}\right) + \left(1 - y^{(i)}\right) \log \left[1 - \sigma\left(\theta^T x^{(i)}\right)\right].$$

MLE chooses the parameters θ that minimize $LL(\theta)$.

$$\theta_j^{(new)} = \theta_j^{(old)} - \eta \cdot \frac{\partial LL\left(\theta^{(old)}\right)}{\partial \theta_j^{(old)}}$$

$$= \theta_j^{(old)} - \eta \cdot \sum_{i=1}^{n} \left[y^{(i)} - \sigma\left(\theta^T x^{(i)}\right) \right] x_j^{(i)}$$

What is the loss function for SVM ?

Linear SVM classifies prediction

$$\hat{y} = \begin{cases} 0, & \text{if } w^T x + b < 0 \\ 1, & \text{if } w^T x + b \geq 0. \end{cases}$$

Hard margin linear SVM classifier objective

$$\underset{w,\, b}{\text{Minimize}} \quad \frac{1}{2} w^T w$$

subject to $t^{(i)}\left(w^T x^{(i)} + b\right) \geq 1$ for $i = 1, 2, \cdots, m$

$t^{(i)} = -1$ for negative instances (if $y^{(i)} = 0$) and $t^{(i)} = 1$ for positive instances (if $y^{(i)} = 1$).

Soft margin linear SVM classifies objective

$$\underset{w,\, b,\, \zeta}{\text{Minimize}} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{m} \zeta^{(i)}$$

subject to $t^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \zeta^{(i)}$ and $\zeta^{(i)} \geq 0$ for $i = 1, 2, \cdots, m$.

equivalently

$$\underset{w,\, b}{\text{Minimize}} \quad \frac{1}{2} w^T w + C \underbrace{\sum_{i=1}^{m} \max\left(0, 1 - t^{(i)}\left(w^T x^{(i)} + b\right)\right)}_{\text{hinge loss}}$$

$\zeta^{(i)}$ measures how much the ith instance is allowed to violate the margin.

(7) When to use logistic regression vs SVMs ?

n = number of features ($x \in \mathbb{R}^{n+1}$)

m = number of training examples.

- If n is large (relative to m) (e.g., $n \geq m$, $n = 10,000$, $m = 10, \cdots, 1000$)

Use logistic regression, or SVM without a kernel (linear kernel)

- If n is small, m is intermediate ($n = 1-1000$, $m = 10-10000$)

Use SVM with Gaussian kernel

- If n is small, m is large ($n = 1-1000$, $m = 50,000+$)

Create/add more features, then use logistic regression or SVM without a kernel.