**Statement**: "This model cannot be estimated with OLS as it involves a non-linear *t*2 term. OLS only can estimate linear equations."

**Answer**: B. False

**Reasoning**:

1. **Linear in Parameters**:
   * The model *y*(*t*)=*y*0​+*v*0​*t*−21​*gt*2 is quadratic with respect to the variable *t*.
   * However, it's linear with respect to the parameters *y*0​, *v*0​, and *g*.
   * OLS requires the equation to be linear in its parameters, not necessarily in its variables. This model meets that criterion.
2. **Polynomial Regression with OLS**:
   * OLS can handle polynomial regressions (like linear, quadratic, cubic, etc.).
   * Our given equation is a type of polynomial regression, specifically quadratic in terms of *t*.
3. **Estimation Process**:
   * To use OLS for this model:
     + Create a design matrix with columns for 0*t*0 (which is just a column of ones), *t*, and 2*t*2.
     + Use OLS to estimate the coefficients for these columns.
     + These coefficients will provide the estimates for *y*0​, *v*0​, and −21​*g* respectively.

**Conclusion**: While the equation is quadratic with respect to *t*, it's linear with respect to its parameters. Hence, OLS can be used to estimate this model.

**k-Nearest Neighbors regression with 5-fold cross-validation on a training dataset of 100 observations**:

1. **5-Fold Cross-Validation**:
   * The data is divided into 5 "folds" or partitions.
   * Each fold serves as a validation set exactly once while the other 4 folds together make up the training set.
   * So, for a dataset with 100 observations, each fold contains 20 observations.
2. **Training Process for Each *k***:
   * Train the model on 80 observations (4 folds) and validate on the remaining 20 observations (1 fold).
   * Repeat this 5 times (once for each fold as the validation set).
   * Therefore, for each *k* value, the model is trained 5 times.
3. **Total Number of Trainings for all *k* values**:
   * We're comparing 15 different *k* values (from 1 to 15).
   * For each *k*, the model is trained 5 times (because of 5-fold cross-validation).
   * Total number of trainings = 15×5=7515×5=75.
4. **Number of Observations Involved in Each Training**:
   * As already mentioned, each training uses 4 out of 5 folds.
   * So, each training involves 45×100=8054​×100=80 observations.

**Answers**:

* Model must be trained 75 times.
* Each fit involves 80 observations.

So, the correct option is:

**D. 75, 80**.

The answer is:

**B. False**

**Reasoning**:

K-Nearest Neighbors (KNN) & QDA is none linear, thus KNN does not inherently produce linear decision boundaries. Instead, the decision boundaries of KNN can be quite irregular and are based on the local structure of the data. The boundaries depend on the distribution of the training samples in the feature space. If neighboring points belong to different classes, the boundary will be more irregular to classify new points based on their proximity to the training points.

In contrast, methods like logistic regression or linear SVM produce linear decision boundaries. KNN's boundaries are determined by the data itself and can be non-linear.

Standardization, which means scaling features to have zero mean and unit variance, is particularly important for algorithms that compute distances between data points or rely on the magnitude of features. Let's evaluate the given methods:

A. **KNN**:

* K-Nearest Neighbors (KNN) calculates distances between data points to find the 'k' closest points (or neighbors). Features with larger scales can disproportionately impact the distance computation, making some features dominate the decision over others.
* **Standardization is crucial for KNN.**

B. **OLS**:

* Ordinary Least Squares (OLS) regression is primarily concerned with the relationships between variables rather than the scale. While standardization can help in interpreting coefficients especially when comparing them, it's not crucial for the algorithm to function properly.
* **Standardization is not crucial for OLS.**

C. **Logistic**:

* Logistic Regression, like OLS, is not inherently dependent on the scale of the features. However, standardizing can help in cases where regularization is used (like in Ridge or Lasso logistic regression) and for interpretation purposes.
* **Standardization is not crucial for basic Logistic Regression but can be helpful in regularized versions.**

D. **LDA**:

* Linear Discriminant Analysis (LDA) aims to find the linear combination of features that best separates classes. LDA assumes that the features have the same variance. If features are on different scales, it might affect LDA's ability to find the best linear combination.
* **Standardization can be important for LDA.**

**Answer**: Among the given options, standardization is especially crucial for: A. KNN and D. LDA.

The described procedure is an instance where a dataset is split into two subsets: one for training the model and another for testing its performance. This specific method is called:

**D. Holdout Method**

**Reasoning**:

A. **10 fold Cross Validation**: In this method, the data is divided into 10 equal parts. One part is used for validation and the other 9 parts are used for training. This process is repeated 10 times, each time with a different part used as the validation set.

B. **Leave One Out Cross Validation (LOOCV)**: In LOOCV, the model is trained on all data points except one, which is used for validation. This process is repeated for each data point in the dataset.

C. **Parallel Computing**: This pertains to the simultaneous execution of multiple tasks or processes, typically used to speed up computational tasks. It doesn't directly relate to a method of splitting or evaluating datasets.

D. **Holdout Method**: The data is divided into two parts - one for training and one for testing, which matches the procedure described in the question.

So, the correct answer is: **D. Holdout Method**.

The decomposition of the expected prediction error in the context of bias-variance trade-off is given by:

Expected Prediction Error=Bias2+Variance+Irreducible Error

Where:

* **Bias** is the error due to overly simplistic assumptions in the learning algorithm.
* **Variance** is the error due to too much complexity in the learning algorithm.
* **Irreducible Error** is the noise inherent in any real-world data collection process.

Given the options:

A. Misclassification Error

B. Training Error

C. Expected Prediction Error

D. Mean Squared Error only include Bias and variance, not irreducible error

The correct answer is:

**C. Expected Prediction Error**.

High Bias/Low Variance models:

* **High Bias**: Means the model makes strong assumptions about the form of the underlying data distribution and often ends up being too simplistic. It might not capture the true patterns in the data.
* **Low Variance**: Means the model will produce similar outputs given different training sets. It's not sensitive to fluctuations in the training data.

Given the options:

A. they overlook useful trends. B. they too closely fit patterns in the training data that are not inherent in the data and therefore don’t generalize well.

The characteristic of High Bias is that the model overlooks useful trends in the data, meaning it misses relevant relations between features and target outputs.

So, the correct answer is:

**A. they overlook useful trends.**

Let's evaluate each of the statements regarding Ridge Regression:

A. **reduces bias at the expense of higher variance.**

* Ridge regression can reduce variance, especially when multicollinearity is present, at the potential cost of introducing a bit of bias.

B. **often sets several of the regression coefficients to zero.**

* This is not true for Ridge Regression. Lasso Regression (L1 regularization) has the property of setting some coefficients to zero. Ridge Regression (L2 regularization) tends to shrink the coefficients toward zero but doesn't typically set them to zero.

C. **has no tuning parameters.**

* This is not true. Ridge and Lasso regression has a tuning parameter, often referred to as *λ* or *α*, which determines the strength of the regularization. This parameter is crucial and is often determined using cross-validation.

D. **adds a squared/quadratic penalty on the parameters.**

* This is true. Ridge Regression adds an L2 penalty which is a squared penalty on the coefficients. The objective function for Ridge Regression is: RSS+*λ*∑*i*=1*p*​*βi*2​ where RSS is the residual sum of squares, *βi*​ are the model coefficients, *p* is the number of predictors, and *λ* is the regularization strength.

Given the above evaluations, the correct answer is:

**D. adds a squared/quadratic penalty on the parameters.**

Let's understand the behavior of Lasso regression with respect to the tuning parameter *λ*:

Lasso regression involves an L1 penalty on the regression coefficients. As *λ* increases:

1. The penalty on the coefficients increases.
2. More coefficients are pushed towards zero, increasing the simplicity of the model (more regularization).
3. With *λ*=0, Lasso regression is just the ordinary least squares, and as *λ* increases, more shrinkage is applied, and more coefficients are set to zero.

Given the options:

A. **at first increase with *λ* and then decrease.**

* This isn't typical behavior for Lasso. As *λ* increases from 0, we would expect the training error to increase, not decrease.

B. **at first decrease with *λ* and then increase.**

* This isn't correct. As *λ* starts at 0 (no regularization), the training error is at its minimum. As *λ* increases, the training error typically increases due to the model becoming simpler and having more coefficients shrunk toward zero.

C. **decrease with *λ*.**

* This isn't correct. As *λ* increases, the model is more regularized, and training error typically increases.

D. **increase with *λ*.**

* This is correct. As *λ* increases, the model becomes more regularized, and more coefficients are pushed towards zero. This typically leads to an increase in training error since the model becomes more biased.

So, the correct answer is:

**D. increase with *λ*.**

Elastic Net is a regularized regression method that combines both L1 and L2 penalties of the Lasso and Ridge methods.

Given the options:

A. **Lasso and Best Subsets**:

* Best Subsets is a procedure to find subsets of predictors that predict the response well, but Elastic Net is not a combination of Lasso and Best Subsets.

B. **Lasso and OLS**:

* OLS is the standard linear regression without regularization. Elastic Net doesn't specifically combine Lasso and OLS; it combines the penalties of Lasso and Ridge.

C. **Lasso and Ridge**:

* This is correct. Elastic Net combines the L1 penalty from Lasso and the L2 penalty from Ridge to regularize regression models.

D. **Ridge and OLS**:

* As explained above, Elastic Net combines the penalties of Lasso and Ridge, not Ridge and OLS.

So, the correct answer is:

**C. Lasso and Ridge**.

In logistic regression, the relationship between the predictor(s) and the log-odds of the response being 1 is linear. The equation is:

log(1−*pp*​)=*β*0​+*β*1​*x*

Where:

* *p* is the probability that *Y*=1.
* 1−*pp*​ is the odds of *Y*=1.
* *β*0​ is the intercept.
* *β*1​ is the coefficient for predictor *x*.

Given this, the coefficient *β*1​ represents the change in the log-odds of *Y*=1 for a one-unit change in *x*.

Given the options: Put in formula sheet

A. **the log-odds of Y = 1**.

* This is correct. As explained above, *β*1​ directly represents the change in the log-odds for a one-unit change in *x*.

B. **the probability of Y = 1**.

* Not directly. While the log-odds and probability are related, *β*1​ directly impacts the log-odds, not the probability itself.

C. **The odds of Y = 1**.

* The relationship is in the log-odds scale, not the odds scale.

D. **None of the above**.

* Given the correct answer is option A, this choice is not accurate.

So, the correct answer is:

**A. the log-odds of Y = 1**.

Let's evaluate each of the options in the context of using Ordinary Least Squares (OLS) for classification:

A. **Masking**:

* Masking typically refers to situations where an outlier is not detected because its effect is "masked" by other outliers. It's an issue more commonly discussed in the context of outlier detection and not specifically a problem of OLS for classification.

B. **Probabilities greater than 1**:

* One problem with using OLS for classification is that the output can fall outside the [0,1] range. This makes it possible to get values greater than 1, which doesn't make sense when interpreting as probabilities.

C. **Probabilities less than 0**:

* Similarly, the output can also be less than 0 when using OLS for classification. This doesn't make sense in the context of probabilities.

D. **Slow to fit/train**:

* OLS, especially in the context of linear regression, is actually known for being computationally efficient. More complex models like some neural networks or ensemble methods are typically slower to train. OLS is not known for being particularly slow.

Given these considerations, the correct answer is:

**D. Slow to fit/train** as this is NOT a problem with OLS for classification.

Given the description "A has a bigger curve than B", I will interpret this as curve A being more towards the top-left corner than curve B on an ROC plot.

Given this interpretation:

An ROC curve that is closer to the top-left corner (and thus having a larger area underneath it compared to the diagonal) indicates better classifier performance. It means that the classifier has a higher true positive rate for a given false positive rate, which is desirable.

So, if curve A is more towards the top-left corner compared to curve B, then:

**A. A** represents better classifier performance.

Quadratic Discriminant Analysis (QDA) and Linear Discriminant Analysis (LDA) are both linear classifiers, but they make different assumptions about the covariance matrices of the classes:

A. **all classes have the same covariance matrix**:

* This is the assumption made by LDA. LDA assumes that each class has the same covariance matrix, which is why the decision boundaries are linear.

B. **each class has the same number of training samples**:

* Neither QDA nor LDA makes this assumption.

C. **each class has its own distribution function**:

* Both LDA and QDA assume that observations from each class are drawn from a Gaussian distribution, but this doesn't uniquely differentiate QDA from LDA.

D. **each class has its own covariance matrix**:

* This is the assumption made by QDA. Unlike LDA, which assumes that every class has the same covariance matrix, QDA assumes that each class can have its own covariance matrix, leading to quadratic decision boundaries.

So, the correct answer that differentiates QDA from LDA is:

**D. each class has its own covariance matrix**.

When *C* is increased, fewer observations will satisfy *P*(*Y*=1∣*X*=*xi*)>*C*. This means that fewer observations will be classified as class 1, and more will be classified as class 0.

Let's analyze the two matrices:

**Matrix A:** Predicted Group 0: 10 + 6 = 16 Predicted Group 1: 3 + 11 = 14

**Matrix B:** Predicted Group 0: 12 + 12 = 24 Predicted Group 1: 1 + 5 = 6

Comparing the two matrices, we can see that Matrix B has more observations classified as class 0 and fewer observations classified as class 1 compared to Matrix A.

Therefore, Matrix B is the result of a larger cutoff *C*.

The answer is: B. B