[redacted]

4 Bias-Variance for Ridge Regression (24 points)

Consider the scalar data-generation model:

$$Y = xw^* + Z$$

where x denotes the scalar input feature, Y denotes the scalar noisy measurement, $Z \sim \mathcal{N}(0,1)$ is standard unit-variance zero-mean Gaussian noise, and w^* denotes the true generating parameter that we would like to estimate.

We are given a set of n training samples $\{x_i, y_i\}_{i=1}^n$ that are generated by the above model with i.i.d. Z_i and distinct x_i. Our goal is to fit a linear model and get an estimate $\widehat{w}$ for the true parameter w^*. For all parts, assume that x_i's are given and fixed (not random).

For a given training set $\{x_i, y_i\}_{i=1}^n$, the ridge-regression estimate for w^* is defined by

$$\widehat{w}_\lambda = \arg\min_{w \in \mathbb{R}} \lambda w^2 + \sum_{i=1}^{n} (y_i - x_i w)^2 \qquad \text{with } \lambda \geq 0.$$

For the rest of the problem, assume that this has been solved and written in the form:

$$\widehat{w}_\lambda = \frac{S_{xy}}{s_x^2 + \lambda} \tag{2}$$

where $S_{xy} = \sum_{i=1}^n x_i Y_i$ and $s_x^2 = \sum_{i=1}^n x_i^2$.

(This is given, no need to rederive it).

(a) (8 pts) **Compute the squared-bias of the ridge estimate $\widehat{w}_\lambda$ defined as follows**

$$\text{Bias}^2(\widehat{w}_\lambda) = (\mathbb{E}[\widehat{w}_\lambda] - w^*)^2. \tag{3}$$

It is fine if your answer depends on w^* or s_x, but it should not depend directly or indirectly on the realizations of the random Z noise. (So, no S_{xy} allowed.)

Hint: First compute the expectation of the estimate $\widehat{w}_\lambda$ over the noises Z in the observation.

(b) (8 pts) **Compute the variance of the estimate $\widehat{w}_\lambda$ which is defined as**

$$\text{Var}(\widehat{w}_\lambda) = \mathbb{E}[(\widehat{w}_\lambda - \mathbb{E}[\widehat{w}_\lambda])^2]. \tag{4}$$

Hint: It might be useful to write $\widehat{w}_\lambda = \mathbb{E}[\widehat{w}_\lambda] + R$ for some random variable R.

(c) (8 pts) **Describe how the squared-bias and variance of the estimate $\widehat{w}_\lambda$ change as we change the value of λ? What happens as $\lambda \to 0$? $\lambda \to \infty$? Is the bias increasing or decreasing? Is the variance increasing or decreasing? In what sense is there a bias/variance tradeoff?**

5 Hospital (25 points)

You work at hospital A. Your hospital has collected patient data to build a model to predict who is likely to get sepsis (a bad outcome). Specifically, the data set contains the feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, and associated real number labels $\mathbf{y} \in \mathbb{R}^n$, where n is the number of patients you are learning from and d is the number of features describing each patient. You plan to fit a linear regression model $\widehat{y} = \mathbf{w}^\top \mathbf{x}$ that will enable you to predict a label for future, unseen patients (using their feature vectors).

However, your hospital has only started collecting data a short time ago. Consequently the model you fit is not likely to be particularly accurate. Hospital B has exactly the same set up as your hospital (i.e., their patients are drawn from the same distribution as yours and they have the same measurement tools). For privacy reasons, Hospital B will not share their data. However, they tell you that they have trained a linear model on their own sepsis-relevant data: ($\mathbf{X}_B$ and $\mathbf{y}_B$) and are willing to share their learned model $\widehat{y} = \widehat{\mathbf{w}}_B^\top \mathbf{x}$ with you. In particular, Hospital B shares their entire Gaussian posterior distribution on $\mathbf{w}$ with you: $\mathcal{N}(\widehat{\mathbf{w}}_B, \mathbf{\Psi})$.

(a) (10 pts) Assume that we use the posterior from Hospital B as our own prior distribution for $\mathbf{w} \sim \mathcal{N}(\widehat{\mathbf{w}}_B, \mathbf{\Psi})$. Suppose that our Hospital A model is given by $\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}$, where the noise, $\boldsymbol{\epsilon}$, has an assumed distribution $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. **Derive the MAP estimate $\widehat{\mathbf{w}}$ for $\mathbf{w}$ using Hospital A's data $\mathbf{X}, \mathbf{y}$ and the prior information from Hospital B.**

HINT: Recall that traditional ridge regression could be derived from a MAP perspective, where the parameter $\mathbf{w}$ has a zero mean Gaussian prior distribution with a scaled identity covariance. How could you use reparameterization (i.e. change of variables) for the problem here?

(b) (15 pts) Now, for simplicity, consider $d = 1$ so that the w is a scalar parameter. Suppose that instead of giving you their posterior distribution, Hospital B only gave you their mean $\widehat{w}_B$. How can you use this information to help fit your model? **Describe in detail how you should use your own hospital's patient data and combine it with the mean $\widehat{w}_B$ from Hospital B in a procedure to find your own $\widehat{w}$ for predicting sepsis in Hospital A.**

Hint 1: You might want to consider introducing an appropriate hyperparameter and doing what you usually do with hyperparameters.

Hint 2: What does the λ hyperparameter in ridge-regression correspond to from a probabilistic perspective?

6 Ridge regression vs. PCA (24 points)

Assume we are given n training data points $(\mathbf{x}_i, y_i)$. We collect the target values into $\mathbf{y} \in \mathbb{R}^n$, and the inputs into the matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ where the rows are the $d-$dimensional feature vectors $\mathbf{x}_i^\top$ corresponding to each training point. Furthermore, assume that $\frac{1}{n}\sum_{i=1}^n \mathbf{x_i} = \mathbf{0}$, $n > d$ and $\mathbf{X}$ has rank d.

In this problem we want to compare two procedures: The first is ridge regression with hyperparameter λ, while the second is applying ordinary least squares after using PCA to reduce the feature dimension from d to k (we give this latter approach the short-hand name k-PCA-OLS where k is the hyperparameter).

Notation: The singular value decomposition of $\mathbf{X}$ reads $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ where $\mathbf{U} \in \mathbb{R}^{n\times n}$, $\mathbf{\Sigma} \in \mathbb{R}^{n\times d}$ and $\mathbf{V} \in \mathbb{R}^{d\times d}$. We denote by $\mathbf{u}_i$ the n-dimensional column vectors of $\mathbf{U}$ and by $\mathbf{v}_i$ the $d-$dimensional column vectors of $\mathbf{V}$. Furthermore the diagonal entries $\sigma_i = \Sigma_{i,i}$ of $\mathbf{\Sigma}$ satisfy $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d > 0$. For notational convenience, assume that $\sigma_i = 0$ for $i > d$.

(a) (6 pts) It turns out that the ridge regression optimizer (with $\lambda > 0$) in the $\mathbf{V}$-transformed coordinates

$$\widehat{\mathbf{w}}_{\text{ridge}} = \arg\min_{\mathbf{w}} \|\mathbf{X}\mathbf{V}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$$

has the following expression:

$$\widehat{\mathbf{w}}_{\text{ridge}} = \text{diag}(\frac{\sigma_i}{\lambda + \sigma_i^2})\mathbf{U}^\top \mathbf{y}. \tag{5}$$

Use $\widehat{y}_{test} = \mathbf{x}_{test}^\top \mathbf{V}\widehat{\mathbf{w}}_{\text{ridge}}$ to denote the resulting prediction for a hypothetical $\mathbf{x}_{test}$. Using (5) and the appropriate scalar $\{\beta_i\}$, this can be written as:

$$\widehat{y}_{test} = \mathbf{x}_{test}^\top \sum_{i=1}^d \mathbf{v}_i \beta_i \mathbf{u}_i^\top \mathbf{y}. \tag{6}$$

What are the $\beta_i \in \mathbb{R}$ for this to correspond to (5) from ridge regression?

(b) (12 pts) Suppose that we do k-PCA-OLS — i.e. ordinary least squares on the reduced k-dimensional feature space obtained by projecting the raw feature vectors onto the $k < d$ principal components of the covariance matrix $\mathbf{X}^\top\mathbf{X}$. Use $\widehat{y}_{test}$ to denote the resulting prediction for a hypothetical $\mathbf{x}_{test}$,

It turns out that the learned k-PCA-OLS predictor can be written as:

$$\widehat{y}_{test} = \mathbf{x}_{test}^\top \sum_{i=1}^{d} \mathbf{v}_i \beta_i \mathbf{u}_i^\top \mathbf{y}. \tag{7}$$

Give the $\beta_i \in \mathbb{R}$ coefficients for k-PCA-OLS. Show work.

Hint 1: some of these β_i will be zero. Also, if you want to use the compact form of the SVD, feel free to do so if that speeds up your derivation.

Hint 2: some inspiration may be possible by looking at the next part for an implicit clue as to what the answer might be.

(c) (6 pts) For the following part, $d = 5$. The following $\boldsymbol{\beta} := (\beta_1, \dots, \beta_5)$ (written out to two significant figures) are the results of OLS (i.e. what we would get from ridge regression in the limit $\lambda \to 0$), λ-ridge-regression, and k-PCA-OLS for some $\mathbf{X}, \mathbf{y}$ (identical for each method) and $\lambda = 1, k = 3$. **Write down which procedure was used for each of the three sub-parts below.**

We hope this helps you intuitively see the connection between these three methods.

Hint: It is not necessary to find the singular values of $\mathbf{X}$ *explicitly, or to do any numerical computations at all.*

(i) $\boldsymbol{\beta} = (0.01, 0.1, 0.5, 0.1, 0.01)$

(ii) $\boldsymbol{\beta} = (0.01, 0.1, 1, 0, 0)$

(iii) $\boldsymbol{\beta} = (0.01, 0.1, 1, 10, 100)$

7 Kernel PCA (24 points)

In lectures, discussion, and homework, we learned how to use PCA to do dimensionality reduction by projecting the data to a subspace that captures most of the variability. This works well for data that is roughly Gaussian shaped, but many real-world high dimensional datasets have underlying low-dimensional structure that is not well captured by linear subspaces. However, when we lift the raw data into a higher-dimensional feature space by means of a nonlinear transformation, the underlying low-dimensional structure once again can manifest as an approximate subspace. Linear dimensionality reduction can then proceed. As we have seen in class so far, kernels are an alternate way to deal with these kinds of nonlinear patterns without having to explicitly deal with the augmented feature space. This problem asks you to discover how to apply the "kernel trick" to PCA.

Let $\mathbf{X} \in \mathbb{R}^{n \times \ell}$ be the data matrix, where n is the number of samples and ℓ is the dimension of the raw data. Namely, the data matrix contains the data points $\mathbf{x}_j \in \mathbb{R}^\ell$ as rows

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix} \in \mathbb{R}^{n \times \ell}. \tag{8}$$

(a) (5 pts) **Compute $\mathbf{X}\mathbf{X}^\top$ in terms of the singular value decomposition $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ where $\mathbf{U} \in \mathbb{R}^{n \times n}$, $\mathbf{\Sigma} \in \mathbb{R}^{n \times \ell}$ and $\mathbf{V} \in \mathbb{R}^{\ell \times \ell}$.** Notice that $\mathbf{X}\mathbf{X}^\top$ is the matrix of pairwise Euclidean inner products for the data points. **How would you get $\mathbf{U}$ if you only had access to $\mathbf{X}\mathbf{X}^\top$?**

(b) (7 pts) Given a new test point $\mathbf{x}_{test} \in \mathbb{R}^\ell$, one central use of PCA is to compute the projection of $\mathbf{x}_{test}$ onto the subspace spanned by the k top singular vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$.

Express the scalar projection $z_j = \mathbf{v}_j^\top \mathbf{x}_{test}$ onto the j-th principal component as a function of the inner products

$$\mathbf{X}\mathbf{x}_{test} = \begin{pmatrix} \langle \mathbf{x}_1, \mathbf{x}_{test} \rangle \\ \vdots \\ \langle \mathbf{x}_n, \mathbf{x}_{test} \rangle \end{pmatrix}. \tag{9}$$

Assume that all diagonal entries of $\mathbf{\Sigma}$ are nonzero and non-increasing, that is $\sigma_1 \geq \sigma_2 \geq \cdots > 0$.

Hint: Express $\mathbf{V}^\top$ in terms of the singular values $\mathbf{\Sigma}$, the left singular vectors $\mathbf{U}$ and the data matrix $\mathbf{X}$. If you want to use the compact form of the SVD, feel free to do so.

(c) (12 pts) How would you define kernelized PCA for a general kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ (to replace the Euclidean inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$)? For example, the RBF kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\delta^2}\right)$.

Describe this in terms of a procedure which takes as inputs the training data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^\ell$ and the new test point $\mathbf{x}_{test} \in \mathbb{R}^\ell$, and outputs the analog of the previous part's z_j coordinate in the kernelized PCA setting. You should include how to compute U from the data, as well as how to compute the analog of $\mathbf{X}\mathbf{x}_{test}$ from the previous part.

Invoking the SVD or computing eigenvalues/eigenvectors is fine in your procedure, as long as it is clear what matrix is having its SVD or eigenvalues/eigenvectors computed. The kernel $k(\cdot, \cdot)$ can be used as a black-box function in your procedure as long as it is clear what arguments it is being given.

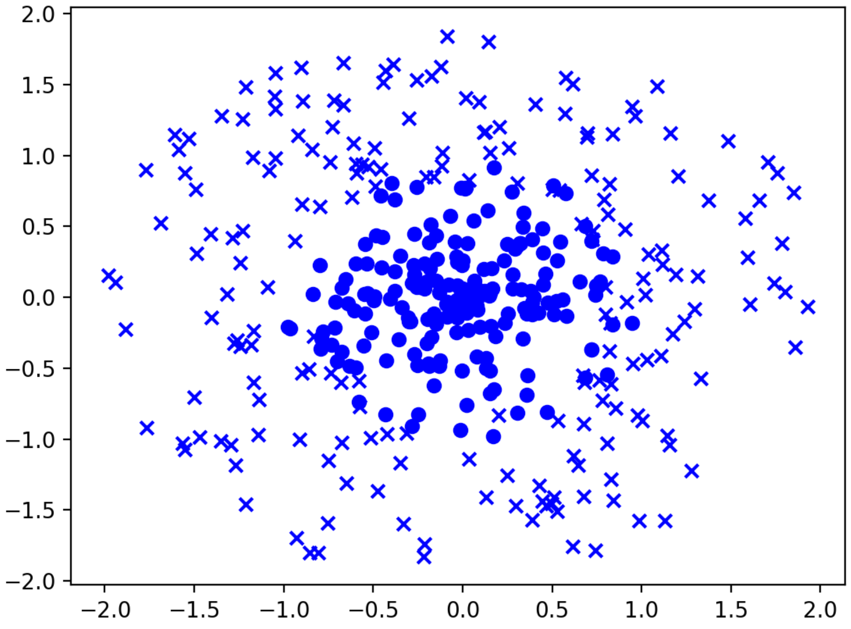
8 Multiple Choice Questions (14 points)

For these questions, select **all** the answers which are correct. You will get full credit for selecting all the right answers. On some questions, real-valued partial credit will be assigned. You will be graded on your **best seven of nine, so feel free to skip up to two of them.**

(a) Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ with $n \geq d$. Suppose $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is the singular value decomposition of $\mathbf{X}$ where $\sigma_i = \Sigma_{i,i}$ are the diagonal entries of $\mathbf{\Sigma}$ and satisfy $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d$ while $\mathbf{u}_i$ and $\mathbf{v}_i$ are the ith columns of $\mathbf{U}$ and $\mathbf{V}$ respectively. **Which of the following is the rank k approximation to X that is best in the Froebenius norm.** That is, which low rank approximation, $\mathbf{X}_k$, for $\mathbf{X}$ yields the lowest value for $||\mathbf{X} - \mathbf{X}_k||_F^2$?

○ $\sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_{n-i}^\top$ ○ $\sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ ○ $\sum_{i=d-k+1}^d \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ ○ $\sum_{i=1}^k \sigma_i \mathbf{u}_{n-i} \mathbf{v}_i^\top$

(b) Consider a simple dataset of points $(x_i, y_i) \in \mathbb{R}^2$, each associated with a label b_i which is -1 or $+1$. The dataset was generated by sampling data points with label -1 from a disk of radius 1.0 (shown as filled circles in the figure) and data points with label $+1$ from a ring with inner radius 0.8 and outer radius 2.0 (shown as crosses in the figure). **Which set of polynomial features would be best for performing linear regression, assuming at least as much data as shown in the figure?**



○ $1, x_i$

○ $1, x_i, y_i$

○ $1, x_i, y_i, x_i^2, x_iy_i, y_i^2$

○ $1, x_i, y_i, x_i^2, x_iy_i, y_i^2, x_i^3, y_i^3, x_i^2y_i, x_iy_i^2$

(c) **Which of the following is a valid kernel function** for vectors of the same length, $\mathbf{x}$ and $\mathbf{y}$?

○ $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$

○ $k(\mathbf{x}, \mathbf{y}) = e^{-\frac{1}{2}||\mathbf{x}-\mathbf{y}||_2^2}$

○ $k(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}^\top \mathbf{y})^p$ for some degree p

○ $k(\mathbf{x}, \mathbf{y}) = k_1(\mathbf{x}, \mathbf{y}) - k_2(\mathbf{x}, \mathbf{y})$ for valid kernels k_1 and k_2.

(d) During training of your model, both independent variables in the matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and dependent target variables $\mathbf{y} \in \mathbb{R}^n$ are corrupted by noise. At test time, the data points you are computing predictions for, $\mathbf{x}_{test}$, are noiseless. **Which method(s) should you use to estimate the value of $\hat{\mathrm{w}}$ from the training data in order to make the most accurate predictions $\mathbf{y}_{test}$ from the noiseless test input data, $\mathbf{X}_{test}$?** Assume that you make predictions using $\mathbf{y}_{test} = \mathbf{X}_{test}\hat{\mathbf{w}}$.

○ OLS

○ Ridge regression

○ Weighted Least Squares

○ TLS

(e) Assume you have n input data points, each with d high quality features ($\mathbf{X} \in \mathbb{R}^{n \times d}$) and associated labels ($\mathbf{y} \in \mathbb{R}^n$). Suppose that $d \gg n$ and that you want to learn a linear predictor. **Which of these approaches would help you to avoid overfitting?**

○ Preprocess $\mathbf{X}$ using $k \ll n$ random projections

○ Preprocess $\mathbf{X}$ using PCA with $k \ll n$ components.

○ Preprocess $\mathbf{X}$ using PCA with n components.

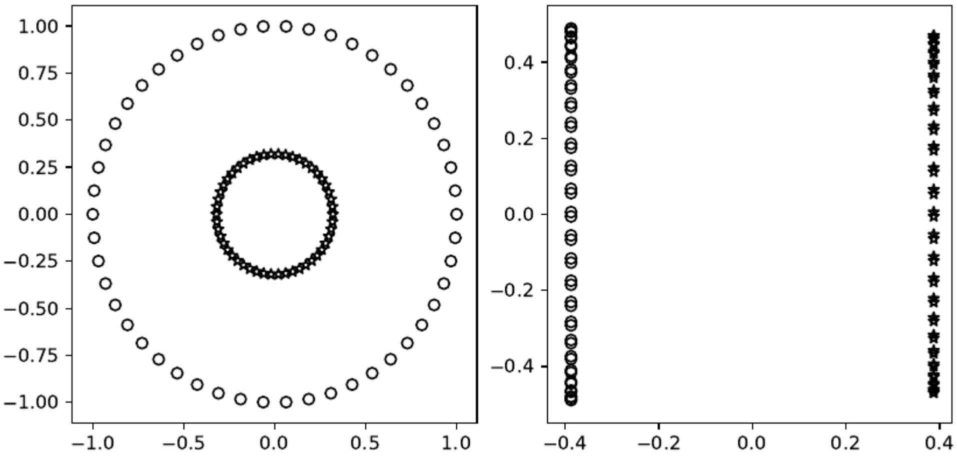
○ Add polynomial features

○ Use a kernel approach

○ Add a ridge penalty to OLS

○ Do weighted least squares

(f) **Which methods could yield a transformation to go from the two-dimensional data on the left to the two-dimensional data on the right?**



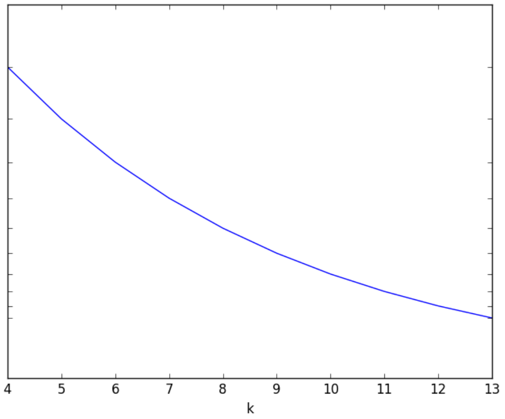
○ Random projections

○ PCA

○ Use of a kernel

○ Adding polynomial features

(g) Your friend is training a machine learning model to predict disease severity based on k different health indicators. She generates the following plot, where the value of k is on the x axis.



Which of these might the y axis represent?

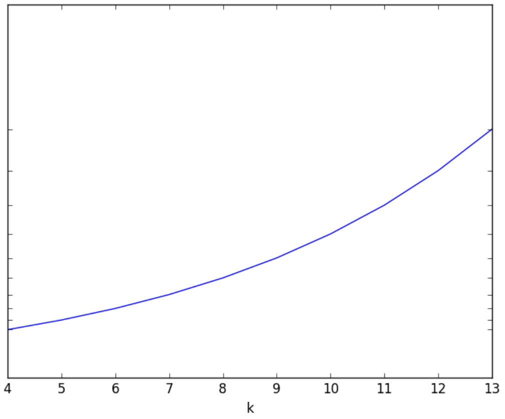
○ Training Error

○ Validation Error

○ Bias

○ Variance

(h) Your friend is training a machine learning model to predict disease severity based on k different health indicators. She generates the following plot, where the value of k is on the x axis.



Which of these might the y axis represent?

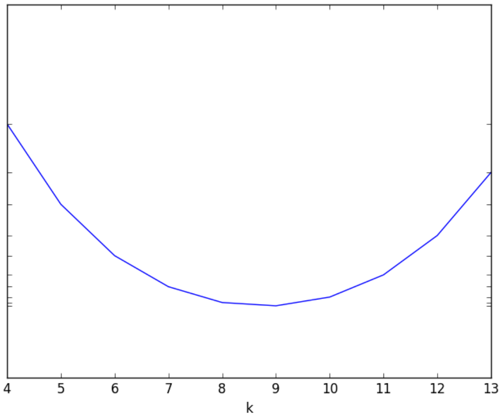
○ Training Error

○ Validation Error

○ Bias

○ Variance

(i) Your friend is training a machine learning model to predict disease severity based on k different health indicators. She generates the following plot, where the value of k is on the x axis.



Which of these might the y axis represent?

○ Training Error

○ Validation Error

○ Bias

○ Variance

[redacted]