compsci671_HW3

2022-10-26

Agreement 1) This assignment represents my own work. I did not work on this assignment with others. All coding was done by myself. Agreement 2) I understand that if I struggle with this assignment that I will reevaluate whether this is the correct class for me to take. I understand that the homework only gets harder.

1 Hoeffding and Beyond: Concentration Inequalities in Statistical Learning

1.1

https://homepage.cs.uiowa.edu/~sriram/5360/fall18/notes/9.10/week4Notes.pdf

let X be a non negative random variable, define random variable $Y = \begin{cases} 1, X \geq \epsilon \\ 0, X < \epsilon \end{cases}$

There are two cases: Case 1: $X < \epsilon$ then by definition $Y = 0$. Since X, ϵ are both non-negative then $Y \leq \frac{X}{\epsilon}$ Case 2: $X \geq \epsilon$ then by definition $Y = 1$. Therefore $Y \leq \frac{X}{\epsilon}$ so we can generalize that for all X, $Y \leq \frac{X}{\epsilon}$. This implies $E[Y] \leq E[\frac{X}{\epsilon}] = \frac{E[X]}{\epsilon}$ and since Y is an indicator random variable, $E[Y] = P(X \geq \epsilon)$. Therefore $P(X \geq \epsilon) \leq \frac{E[X]}{\epsilon}$

1.2

https://homepage.cs.uiowa.edu/~sriram/5360/fall18/notes/9.10/week4Notes.pdf

$P(|X - \mu| \geq \epsilon) = P((X - \mu)^2 \geq \epsilon^2)$. Lets define a random variable $Y = (X - \mu)^2$, then by the markov inequality $P(Y \geq \epsilon^2) \leq \frac{E[Y]}{\epsilon^2} = \frac{E[(X-\mu)^2]}{\epsilon^2} = \frac{\sigma^2}{\epsilon^2}$. Therefore $P(|X - \mu| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2}$

1.3

https://cims.nyu.edu/~sling/MATH-SHU-236-2020-SPRING/MATH-SHU-236-Lecture-17-18-Concentration.pdf

$P(|X - \mu| \geq \epsilon) = P(|X - \mu|^k \geq \epsilon^k)$. Lets define a random variable $Y = |X - \mu|^k$, then by the markov inequality $P(Y \geq \epsilon^k) \leq \frac{E[Y]}{\epsilon^k} = \frac{E[|X-\mu|^k]}{\epsilon^k}$. Therefore $P(|X - \mu| \geq \epsilon) \leq \frac{E[|X-\mu|^k]}{\epsilon^k}$

1.4

https://cims.nyu.edu/~sling/MATH-SHU-236-2020-SPRING/MATH-SHU-236-Lecture-17-18-Concentration.pdf

$P(X - \mu \geq \epsilon) = P(\lambda(X - \mu) \geq \lambda\epsilon) = P(e^{\lambda(X-\mu)} \geq e^{\lambda\epsilon})$ since $\lambda > 0$ and $\epsilon \in \mathbb{R}$ and the exponential function is monotonically increasing. Lets define a random variable $Y = e^{\lambda(X-\mu)}$, then by the markov inequality $P(Y \geq e^{\lambda\epsilon}) \leq \frac{E[Y]}{e^{\lambda\epsilon}} = \frac{E[e^{\lambda(X-\mu)}]}{e^{\lambda\epsilon}} = \inf_{\lambda \geq 0} \frac{M_{X-\mu}(\lambda)}{e^{\lambda\epsilon}}$. Therefore $P(X - \mu \geq \epsilon) \leq \inf_{\lambda \geq 0} \frac{M_{X-\mu}(\lambda)}{e^{\lambda\epsilon}}$

1.5

https://cims.nyu.edu/~sling/MATH-SHU-236-2020-SPRING/MATH-SHU-236-Lecture-17-18-Concentration.pdf

$$P\left(\left(\frac{1}{n}\sum_{i=1}^{n}X_i\right)-\mu\geq\epsilon\right)=P\left(\left(\frac{1}{n}\sum_{i=1}^{n}X_i\right)-\left(\frac{1}{n}\sum_{i=1}^{n}\mu\right)\geq\epsilon\right)=P\left(\frac{1}{n}\sum_{i=1}^{n}(X_i-\mu)\geq\epsilon\right)=P\left(\sum_{i=1}^{n}(X_i-\mu)\geq n\epsilon\right)$$

By the Chernoff Bound

$$P\left(\sum_{i=1}^{n}(X_i-\mu)\geq n\epsilon\right)\leq E\left[\exp\left(\lambda\sum_{i=1}^{n}(X_i-\mu)\right)\right]e^{-\lambda n\epsilon}$$

By independence of $X_i's$

$$E\left[\exp\left(\lambda\sum_{i=1}^{n}(X_i-\mu)\right)\right]e^{-\lambda n\epsilon}=e^{-\lambda n\epsilon}\prod_{i=1}^{n}E\left(e^{\lambda(X_i-\mu)}\right)$$

By the Hoeffding lemma

$$\leq e^{-\lambda n\epsilon}\prod_{i=1}^{n}e^{\frac{\lambda^2(b-a)^2}{8}}=\exp\left(-\lambda n\epsilon+\frac{n\lambda^2(b-a)^2}{8}\right)$$

When we minimize the exponent over λ to get the tighter bound we get:

let $f=\left(-\lambda n\epsilon+\frac{n\lambda^2(b-a)^2}{8}\right)$ then when we set $f'=0$ and solve for λ^* we get $f'=n\epsilon+\frac{n\lambda(b-a)^2}{4}=0$ so $\lambda^*=\frac{4\epsilon}{(b-a)^2}$. Plugging in λ^* for λ we get

$$\min_{\lambda\geq0}\exp\left(-\lambda n\epsilon+\frac{n\lambda^2(b-a)^2}{8}\right)=\exp\left(\frac{2n\epsilon^2}{(b-a)^2}\right).$$

1.6

a.

Hoeffding's inequality gives us:

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n}X_i-\mu\right|\geq\epsilon\right)\leq2\exp\left(\frac{-2n\epsilon^2}{(b-a)^2}\right)$$

then if we let $X_i=l_{01}(y_i,g(x_i))$ and $\mu=E_D[l_{01}(y,g(x))]$ and since $l_{01}(y_i,g(x_i))\in[0,1]$, we get:

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n}l_{01}(y_i,g(x_i))-E_D[l_{01}(y,g(x))]\right|\geq\epsilon\right)\leq2e^{\left(-2n\epsilon^2\right)}$$

let $\delta=2e^{\left(-2n\epsilon^2\right)}$ then by inversion

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n}l_{01}(y_i,g(x_i))-E_D[l_{01}(y,g(x))]\right|\leq\epsilon\right)\geq1-2e^{\left(-2n\epsilon^2\right)}=1-\delta$$

therefore,

$$P\left(|R_s(g)-R(g)|\leq\epsilon\right)\geq1-\delta$$

when we solve for n and we get $n = \frac{\ln \frac{2}{\delta}}{2\epsilon^2}$

b.

Chebyschev's Inequality gives us:

$$P(|X - \mu| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2}$$

Since $\text{var}(l_{01}(y_i, g(x_i))) = 1$ then the $\text{var}(\frac{1}{n}\sum_{i=1}^{n} l_{01}(y_i, g(x_i)) = n \cdot (\frac{1}{n})^2 \cdot (\sigma^2) = n \cdot \frac{1}{n^2} \cdot 1 = \frac{1}{n}$
then if we let $X = \frac{1}{n}\sum_{i=1}^{n} l_{01}(y_i, g(x_i))$ and $\mu = E_D[l_{01}(y_i, g(x_i))]$ and since $l_{01}(y_i, g(x_i)) \in [0, 1]$ and $\sigma^2 = \frac{1}{n}$, we get:

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n} l_{01}(y_i, g(x_i)) - E_D[l_{01}(y, g(x))]\right| \geq \epsilon\right) \leq \frac{\frac{1}{n}}{\epsilon^2} = \frac{1}{n\epsilon^2}$$

let $\delta = \frac{1}{n\epsilon^2}$ then by inversion

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n} l_{01}(y_i, g(x_i)) - E_D[l_{01}(y, g(x))]\right| \leq \epsilon\right) \geq 1 - \frac{1}{n\epsilon^2} = 1 - \delta$$

therefore,

$$P\left(|R_s(g) - R(g)| \leq \epsilon\right) \geq 1 - \delta$$

when we solve for n and we get $n = \frac{1}{\delta\epsilon^2}$

For a given variance, if an algorithm satisfies Hoeffding's inequality requirements, Hoeffdings upper bound will always be lower than the Chebyshev upper bound.

1.7

Since A is β-stable, let S be a set of points g_S, g_{S^k} be classifiers of A:

$$\sup_{x_1, y_1} |l_{01}(g_S(x_1), y_1) - l_{01}(g_{S^k}(x_1), y_1)| \leq \beta$$

by the triangle inequality

$$\sup_{x_1,y_1,x_2,y_2} |l_{01}(g_S(x_1), y_1) - l_{01}(g_{S^k}(x_1), y_1) + l_{01}(g_S(x_2), y_2) - l_{01}(g_{S^k}(x_2), y_2)| \leq \sup_{x_1,y_1,x_2,y_2} |l_{01}(g_S(x_1), y_1) - l_{01}(g_{S^k}(x_1), y_1)| + |l_{01}$$

therefore:

$$\sup_{x_1,y_1,x_2,y_2} \left|\sum_{i=1}^{2} l_{01}(g_S(x_i), y_i) - \sum_{i=1}^{2} l_{01}(g_{S^k}(x_i), y_i)\right| \leq 2\beta$$

If we repeat for all n points in S we have

$$\sup_{x_1,y_1,\ldots,x_n,y_n} \left| \sum_{i=1}^{n-1} l_{01}(g_S(x_i), y_i) - \sum_{i=1}^{n-1} l_{01}(g_{S^k}(x_i), y_i) + l_{01}(g_S(x_k), y_k) - l_{01}(g_{S^k}(x'_k), y'_k) \right| \leq (n-1)\beta + 1$$

if we multiply everything by $\frac{1}{n}$ we get:

$$\sup_{x_1,y_1,\ldots,x_n,y_n} \frac{1}{n} \left| \sum_{i=1}^{n-1} l_{01}(g_S(x_i), y_i) - \sum_{i=1}^{n-1} l_{01}(g_{S^k}(x_i), y_i) + l_{01}(g_S(x_k), y_k) - l_{01}(g_{S^k}(x'_k), y'_k) \right| \leq \frac{1}{n}\left[(n-1)\beta + 1\right]$$

then if we distribute the $\frac{1}{n}$,

$$\sup_{x_1,y_1,\ldots,x_n,y_n} \left| \frac{1}{n}\left[\sum_{i=1}^{n-1} l_{01}(g_S(x_i), y_i) + l_{01}(g_S(x_k), y_k)\right] - \frac{1}{n}\left[\sum_{i=1}^{n-1} l_{01}(g_{S^k}(x_i), y_i) + l_{01}(g_{S^k}(x'_k), y'_k)\right] \right| \leq \frac{(n-1)}{n}\beta + \frac{1}{n}$$

And since

$$R_A(S) := R_A((x_1, y_1), \cdots, (x_n, y_n)) := \frac{1}{n}\sum_{i=1}^{n} l_{01}(g_S(x_i), y_i)$$

Therefore:

$$\left|R_A(S) - R_A(S^k)\right| \leq \frac{(n-1)}{n}\beta + \frac{1}{n} \leq \beta + \frac{1}{n}$$

McDiarmid's inequality gives us

$$P\left(|f(X_1, \cdots, X_n) - E[f(X_1, \cdots, X_n)]| \geq \epsilon\right) \leq 2\exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{n} c_i^2}\right). \quad \text{(McDiarmid's Inequality)}$$

Since R_A maps $\mathbb{R}^n \to \mathbb{R}$ of n i.i.d random variables and using the bounded difference property of R_A

Then we can use McDiarmid's inequality and we can substitute R_A for f and substitute c_i with the upper bound of $\left|R_A(S) - R_A(S^k)\right|$

so

$$P(|R_A(S) - E[R_A(S)]| \geq \epsilon) \leq 2\exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{n}(\beta + \frac{1}{n})^2}\right) = 2\exp\left(\frac{-2\epsilon^2}{n(\beta + \frac{1}{n})^2}\right)$$

2 Classic Exercises in VC Dimension

2.1

VC dimension = 1

If we have 1 point x, let x be the point 0.5. If the true label for x is 1, then let $f \in \mathcal{F}$ s.t. $t \in (.5, 1]$. This shatters the point. If the true label for x is 0, then let $f \in \mathcal{F}$ s.t. $t \in [0, .5]$. This shatters the point as well. Since this is every permutation of labels, the VC dimension of $\mathcal{F}$ is at least 1

Now let x_1, x_2 be two arbitrary points on the interval $[0, 1]$. let the true label of x_1 be 1 and the true label of x_2 be 0. If $x_1 = x_2$ then they will be incorrectly classified by any $f \in \mathcal{F}$. Therefore, without loss of generality, let $x_1 < x_2$. If we want to classify x_1 correctly, we need an $f \in \mathcal{F}$ s.t. $t \in [x_1, 1]$. However, if $t \in [x_1, 1]$ we cannot classify x_2 correctly. therefore, $\mathcal{F}$ cannot shatter any arrangement of points of dimension 2. Thus the VC dimension of $\mathcal{F}$ is 1.

2.2

VC dimension = 2

Given two points x_1, x_2, Let $x_1 = .25$ and $x_2 = .75$. Case 1: true label of x_1, x_2 is 1. Then let $f \in \mathcal{F}$ s.t. $t_1 \in [0, .25)$ and $t_2 \in (.75, 1]$ then f shatters the points. Case 2: true label of x_1, x_2 is 0. Then let $f \in \mathcal{F}$ s.t. $t_1 \in (.75, 1)$ and $t_2 \in (.75, 1]$ then f shatters the points. Case 3: true label of x_1 is 1 and true label of x_2 is 0. Then let $f \in \mathcal{F}$ s.t. $t_1 \in (0, .25)$ and $t_2 \in (.25, .75)$ then f shatters the points. Case 4: true label of x_1 is 0 and true label of x_2 is 1. Then let $f \in \mathcal{F}$ s.t. $t_1 \in (.25, .75)$ and $t_2 \in (.75, 1)$ then f shatters the points. Therefore since $\mathcal{F}$ shatters all permutations of labels $\mathcal{F}$ has VC dimension of at 2

Now let x_1, x_2, x_3 be three arbitrary points on the interval $[0, 1]$. If any two points have the same value then then there will be some permutation of labels that cannot be shattered by $\mathcal{F}$, therefore without loss of generality, let $x_1 < x_2 < x_3$. let the true label of x_1 be 1, the true label of x_2 be 0 and the true label of x_3 be 1. If we want to classify x_1 and x_2 correctly, we need an $f \in \mathcal{F}$ s.t. $t_1 \in [0, x_1)$ and $t_2 \in (x_1, x_2)$. However, if $t_2 < x_2$ we cannot classify x_3 correctly. therefore, $\mathcal{F}$ cannot shatter any arrangement of points of dimension 3. Thus the VC dimension of $\mathcal{F}$ is 2.

2.3

vc dimension = d

Let the dimension of the vector space be d so all $x \in \mathbb{R}^d$ and the corresponding $y_i \in \{-1, 1\}$. Now lets have d points. let $x_i = \{e_i\}_{i=1}^d$ where e_i is the standard basis vector for element i. Then let $b = \sum_{i=1}^d y_i e_i$. Then $f(e_i) = \text{sign}(b^T e_i) = \text{sign}(y_i) = y_i$ therefore $f(x)$ shatters all permutations of labels of dimension d and thus has a VC dimension of at least d.

Now lets have $d + 1$ points. Then there exists at least one point, lets call x_{d+1}, where x_{d+1} is a linear combination of basis vectors $x_1, ...x_d$, so $x_{d+1} = \sum_{k=1}^d a_k x_k$ and let $y_{d+1} = -1$. Set b to be the same from the lower bound that perfectly classified the points then let the weights of a be the same as the weights of c. So $f(x_k) = y_k = \text{sign}(a_k) = \text{sign}(b^T x_k)$. Then $f(x_{d+1}) = \text{sign}(b^T \sum_{k=1}^d a_k x_k) > 0$ so $f(x_{d+1})$ is incorrectly classified and thus the VC dimension of $\mathcal{F}$ is d.

Now in our d dimensional vector space lets have $d+1$ points. Then there exists at least one point, lets call x_j, where x_j is a linear combination of multiple points, so $x_j = \sum_{k \neq j} a_k e_k$. Then $f(x_j) = \text{sign}(b^T \sum_{k \neq j} a_k e_k) = \text{sign}(\sum_{k \neq j} y_k a_k)$ and w.l.o.g $y_j = -1$. Assume $f(x_j)$ is classified correctly then $\text{sign}(y_j) = \text{sign}(\sum_{k \neq j} y_k a_k)$. Then another permutation of labels would be leaving all labels the same and setting $\text{newsign}(y_j) = \text{sign}(-y_j)$. In this case the $f(x)$ does not classify x_j correctly. therefore, $\mathcal{F}$ cannot shatter any arrangement of points of dimension d+1. Thus the VC dimension of $\mathcal{F}$ is d.

2.4

max leaves l would have a VC dimension of l

If we have a decision tree with l leaves and l unique points, then we can construct the splits of the tree so that each leaf has exactly 1 point. Then we would classify the leaf with the label of the single point occupying that leaf. therefore the VC dimension of this set of binary tree classifiers is at least l. If we have l+1 data points then at least one leaf will contain 2 points. Without loss of generality lets call these points x_1 and x_2. One permutation of labels would be that the true label of x_1 is 1 and the true label of x_2 is 1 therefore the leaf would be classified as 1 and the two points are classified correctly. However, another permutation of labels is that the true label of x_1 is 1 and the true label of x_2 is 0. In this instance, the leaf will either be 1 or 0 and the decision tree will not shatter l+1 points. Therefore, the VC dimension of binary decision trees with the number of leaves at most l is l.

A binary decision tree with the number of splits at most d has at most l leaves where $l = d + 1$ therefore the VC dimension of this set of classifiers is $d + 1$.

2.5

$\mathcal{F}$ is a finite hypothesis class of finite dimension. Therefore $\exists d$ s.t. $VCdim(\mathcal{F}) = d$ since every classifier $f \in \mathcal{F}$ labels the points in a region and since there are at least 2^d permutations of labels then there are at least 2^d different classifiers in f. Therefore, $|\mathcal{F}| \geq 2^d$ and with some algebra we get $\log_2 |\mathcal{F}| \geq d$. Thus the $VCdim(\mathcal{F}) \leq \log_2 |\mathcal{F}|$

3 Logistic Regression and Kernals

3.1

Since $\mathcal{H}$ is a reproducing kernel Hilbert space we can use the representer theorem

$f^*(\cdot) = \sum_{i=1}^n \alpha_i k(\cdot, x_i)$

Since $f_\theta = \theta^T x$ and since $\mathcal{H}$ is a Reproducing Kernel Hilbert Space, we have $f(x) = \langle k(x, \cdot)\ , f(\cdot)\ \rangle$. Since $f(\cdot) = \theta^T$ then this means that $k(x, \cdot) = x$ which indicates the optimal solution is of the form $f^*(\cdot) = \sum_{i=1}^n \alpha_i x_i$

3.2

for the primal problem we want to minimize f

$$f(\omega, \zeta) = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n} g(\zeta_i)$$

subject to the constraint:

$$y_i(\omega^T x_i) \geq \zeta_i, \forall i$$

Therefore, let $h(\zeta_i) = \zeta_i - y_i(\omega^T x_i)$:

$$\mathcal{L}(\omega, X, y, \zeta, \alpha, C) = f(\omega, \zeta) + \sum_{i=1}^{n} \alpha_i h(\zeta_i) = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n} g(\zeta_i) + \sum_{i=1}^{n} \alpha_i(\zeta_i - y_i(\omega^T x_i))$$

Therefore, given $\alpha_i \geq 0, \forall i$, for our primal problem we have:

$$\min_{\zeta,\omega}[\max_{\alpha,X,y,C} \mathcal{L}(\omega, X, y, \zeta, \alpha, C)]$$

Thus the dual formulation is:

$$\max_{\alpha,X,y,C}[\min_{\zeta,\omega} \mathcal{L}(\omega, X, y, \zeta, \alpha, C)] = \max_{\alpha,X,y,C}[\min_{\zeta,\omega}[\frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n} g(\zeta_i) + \sum_{i=1}^{n} \alpha_i(\zeta_i - y_i(\omega^T x_i))]]$$

We can then find the optimal ω and ζ_i and plug them into the dual problem

$$\frac{\partial \mathcal{L}}{\partial \omega} = \omega + \sum -\alpha_i y_i x_i = 0$$

$$\omega^* = \sum \alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial \zeta_i} = -C\frac{e^{-\zeta_i}}{1 + e^{-\zeta_i}} + \alpha_i = 0$$

$$\alpha_i(1 + e^{-\zeta_i}) = Ce^{-\zeta_i}$$

$$e^{-\zeta_i} = \frac{\alpha_i}{C - \alpha_i}$$

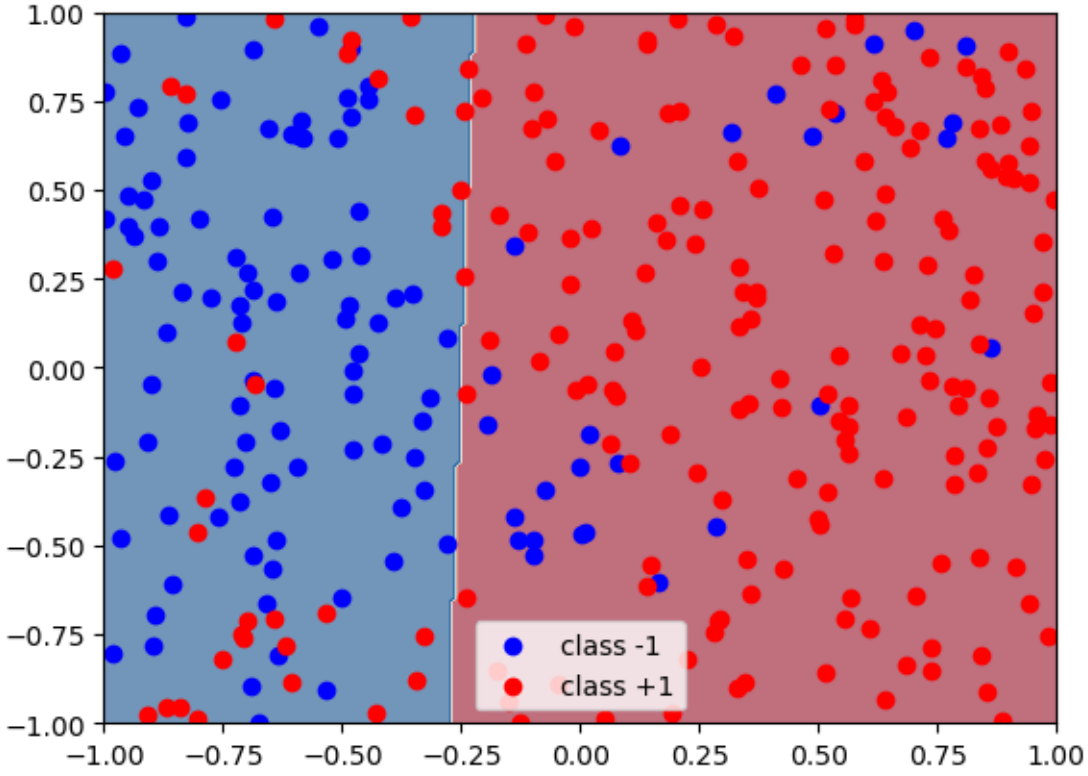
$$\zeta_i^* = \ln\left(\frac{C}{\alpha_i} - 1\right)$$

$$\max_{\alpha,X,y,C} \mathcal{L}(X, y, \alpha, C)] = \max_{\alpha,X,y,C} \left[\frac{1}{2}||\omega^*||^2 + C\sum_{i=1}^{n} g(\zeta_i^*) + \sum_{i=1}^{n} \alpha_i(\zeta_i^* - y_i(\omega^{*T} x_i))\right]$$

4 Kernel Power on SVM and Regularized Logistic Regression

4.1

Look at code for more details



4.2 Look at code for more details

4.3 Look at code for more details

4.4 Look at code for more details

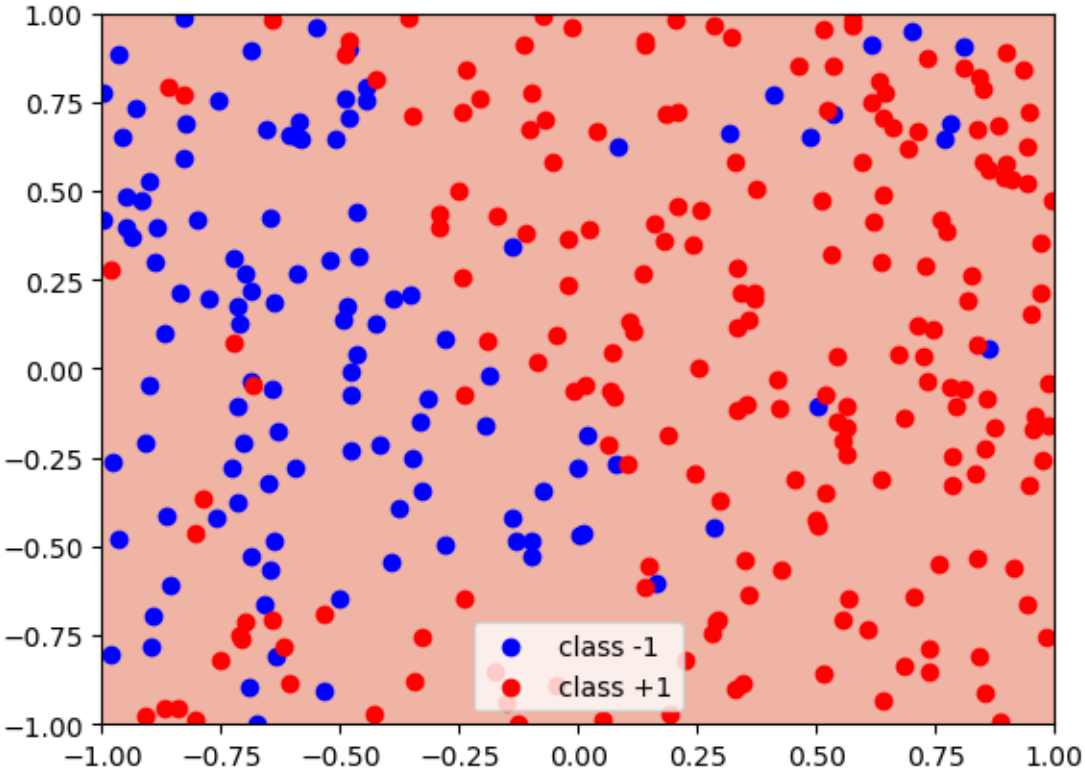


Figure 1: polynomial kernel decision boundry

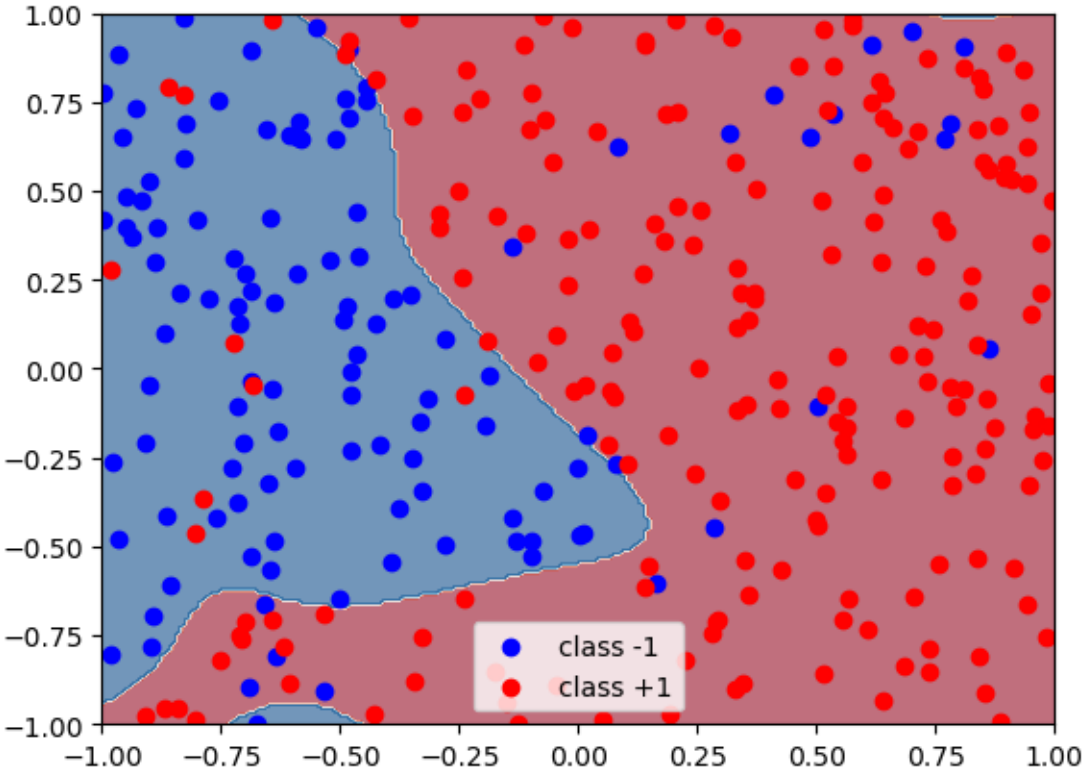
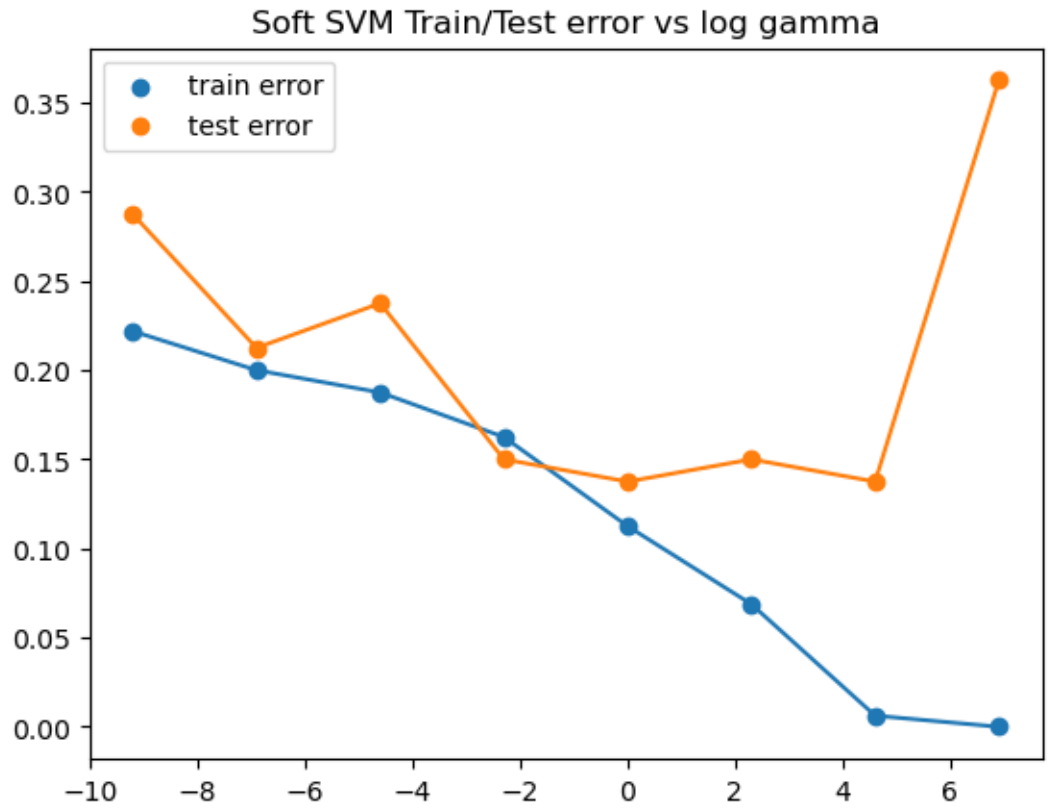


Figure 2: Gaussian kernel decision boundry



After plotting the training and testing error vs γ we noticed that as γ increases both training and testing error go down. However, as γ gets too large testing error shoots up while training error continues to decrease. This occurs because as we increase γ we are decreasing the variance of the Gaussian kernel so as we take linear combinations of these finer kernels we are able to identify finer patterns in the data. However, when γ gets too large, meaning σ gets too small, the widths of these Gaussian kernels become to fine and begin to overfit on the training data. That is why we see the training error continue to decrease as the test error rises sharply. Therefore a small γ i.e large σ corresponds to high bias, since a gaussian kernel with a large width would cover much of the feature space and would not be able to make distinctions between the points.

4.5 Look at code for more details

4.6 Look at code for more details

4.7 Look at code for more details

4.8 Look at code for more details

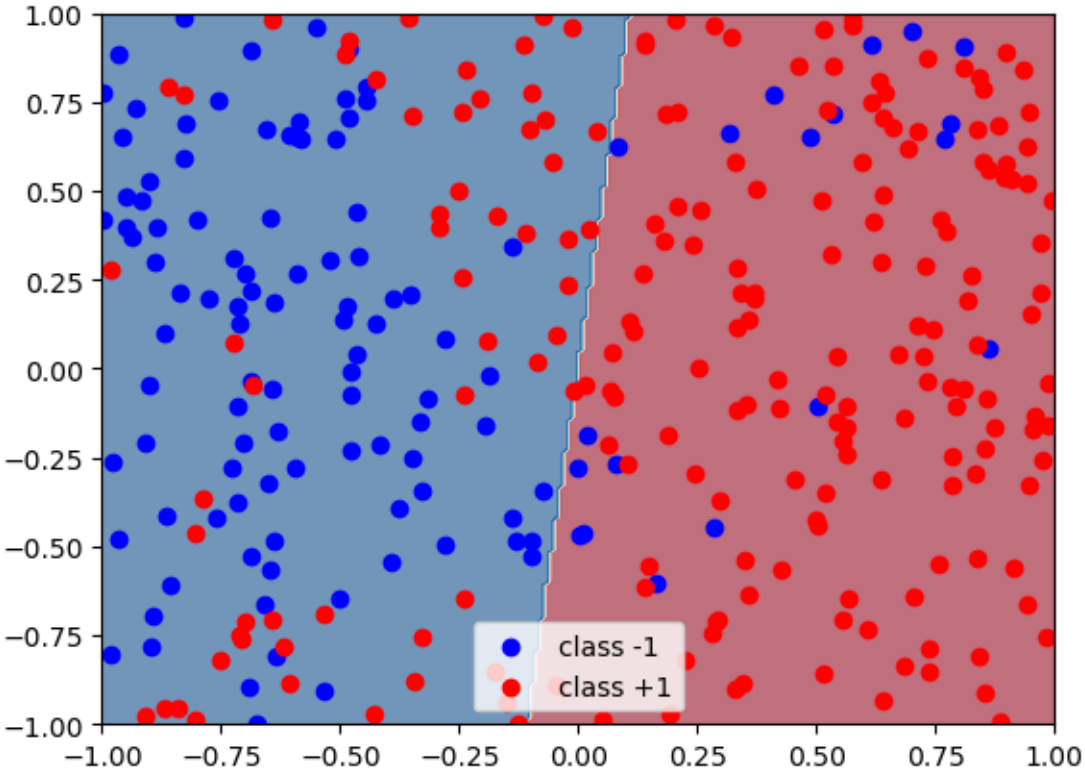


Figure 3: L2 Logistic Regression linear kernel decision boundry

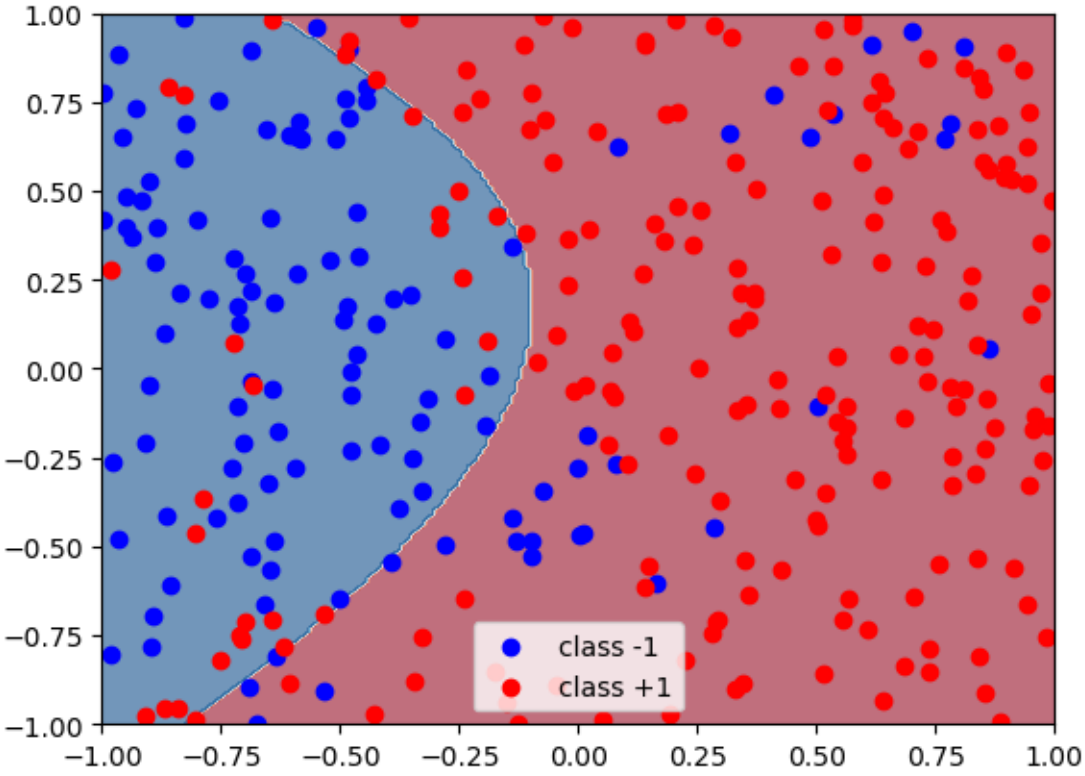


Figure 4: L2 Logistic Regression polynomial kernel decision boundry

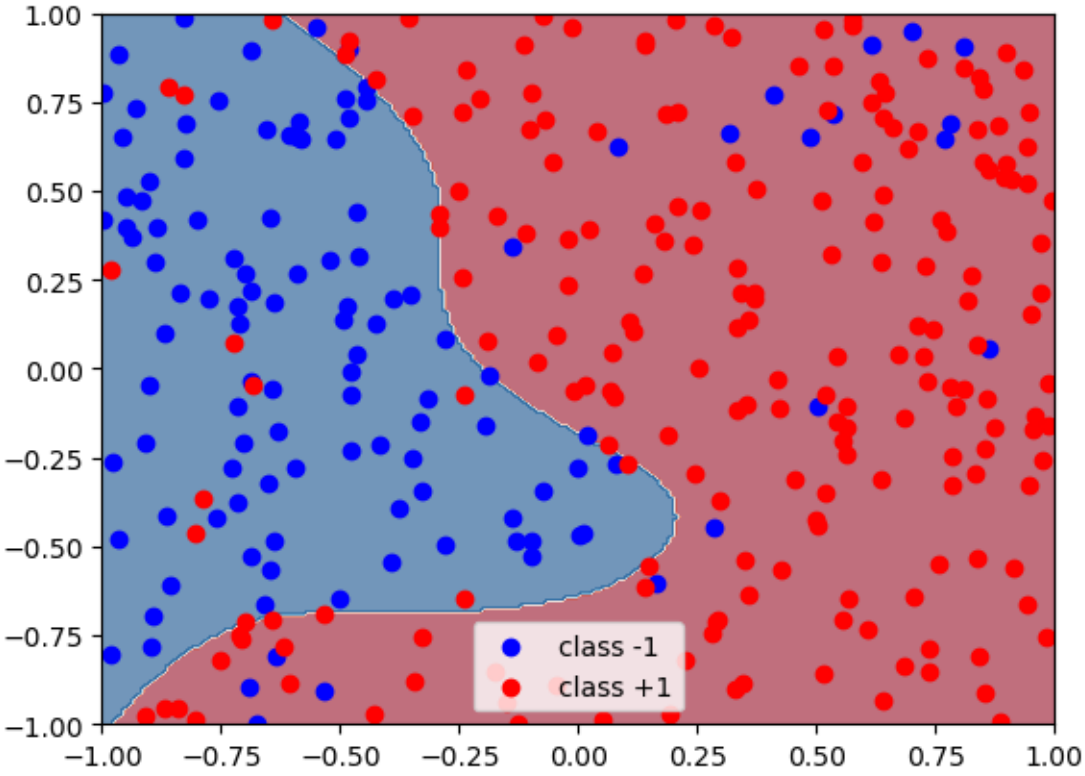
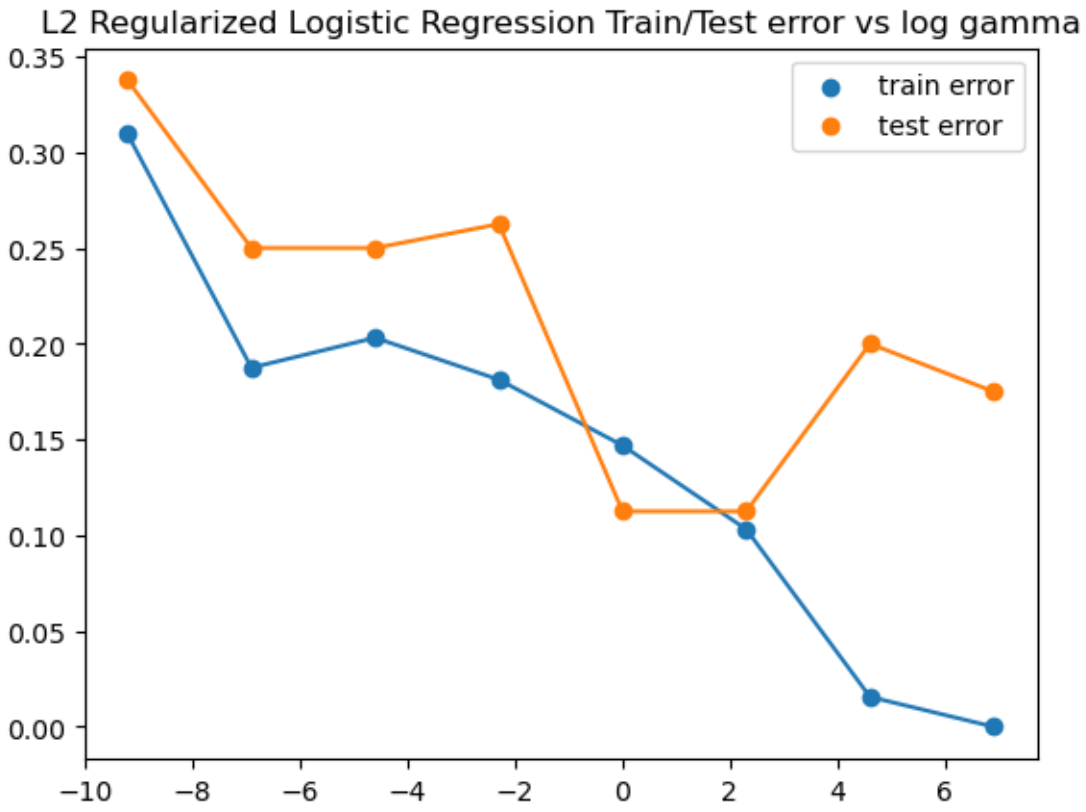


Figure 5: L2 Logistic Regression Gaussian kernel decision boundry



5 Sparse Logistic Regression

5.1

Look at code for more details

5.2

Look at code for more details

There are two components in building the 300 different subsets of features. 1. get all 44 single predictors 2. iterate over the 946 unique combinations of 2 predictors. Calculate the absolute value of the correlation for each pair. Sort the pairs of predictors by absolute correlation in ascending order. Take the first 256 pairs (those with the lowest absolute correlations).

5.3

look at code for more details

5.4

look at code for more details

L2 regularized Logistic Regression had a much higher training AUC than our optimal sparse Logistic Regression but the test AUC for the two models were essentially the same. In addition, testing accuracy was higher for the sparse logistic regression was higher than the testing accuracy for the L2 regularized logistic regression so the L2 regularized logistic regression was beginning to overfit.

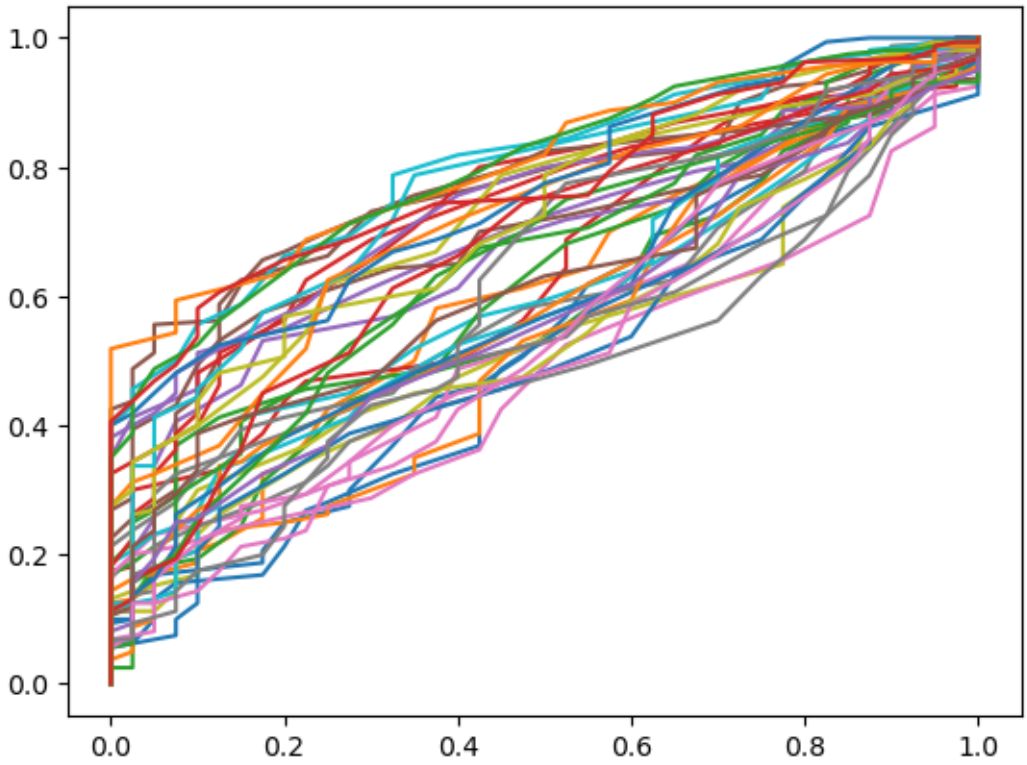


Figure 6: ROC curves for Single Variable Logistic Regression

6 Ridge Regression and its freinds

6.1

Look at code for more details

Ridge Regression took 3.5 minutes while Kernel Ridge Regression took 1.25 minutes. If we look at the closed form solutions of the two algorithms we see that Ridge Regression's form is $(X^TX + CI)^{-1}X^T$, where X is an n x p matrix, which has a matrix operation time complexity of $O(n \cdot p^2 + p^3)$. The closed form solution for Kernel Ridge Regression is $X^T(XX^T + CI)^{-1}$ which has a matrix operation time complexity of $O(p \cdot n^2 + n^3)$. Since in our example $p = 5 \cdot 10^6$ and $n = 100$ the method which was less dependent on feature size, Kernel Ridge Regression, was faster.

6.2

Look at code for more details

As λ our regularization strength got stronger i.e. λ got larger, we notice that the number of non-zero parameters for lasso regression decreases while the number of non-zero parameters for ridge regression remains the same. One obvious advatage of lasso over ridge is that it results in simpler more parsimonious models which are more interpretable. Both types of regularization help with over fitting but since lasso can take coefficients to 0 it can help with variable selection as well.

6.3

Look at code for more details

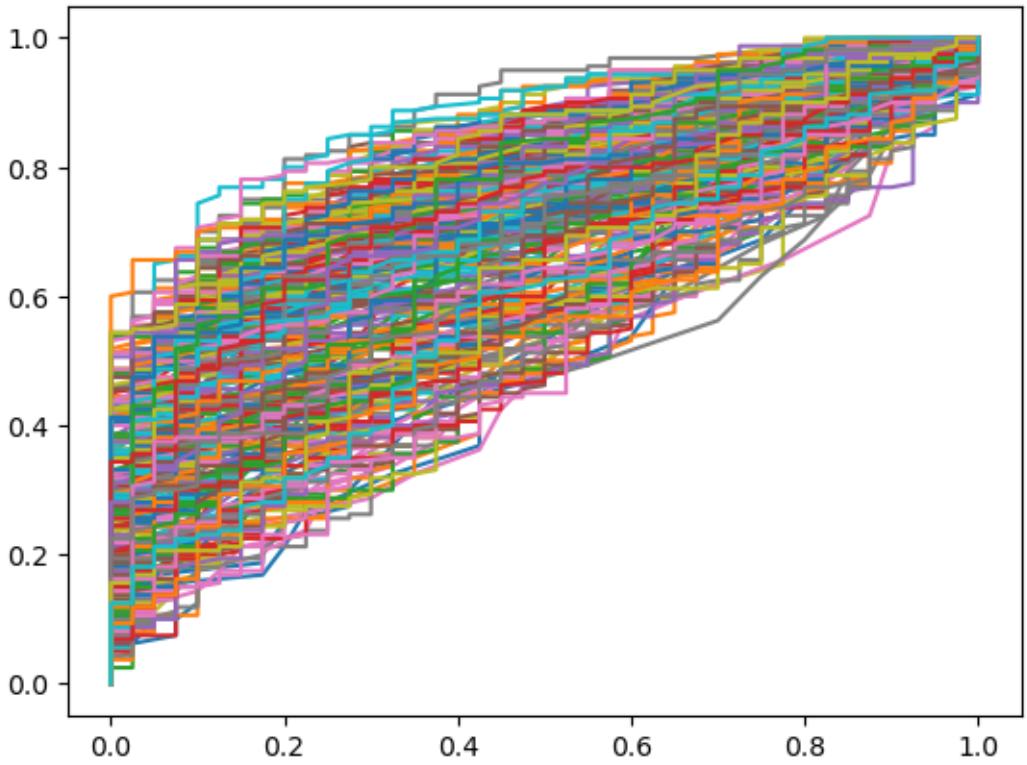
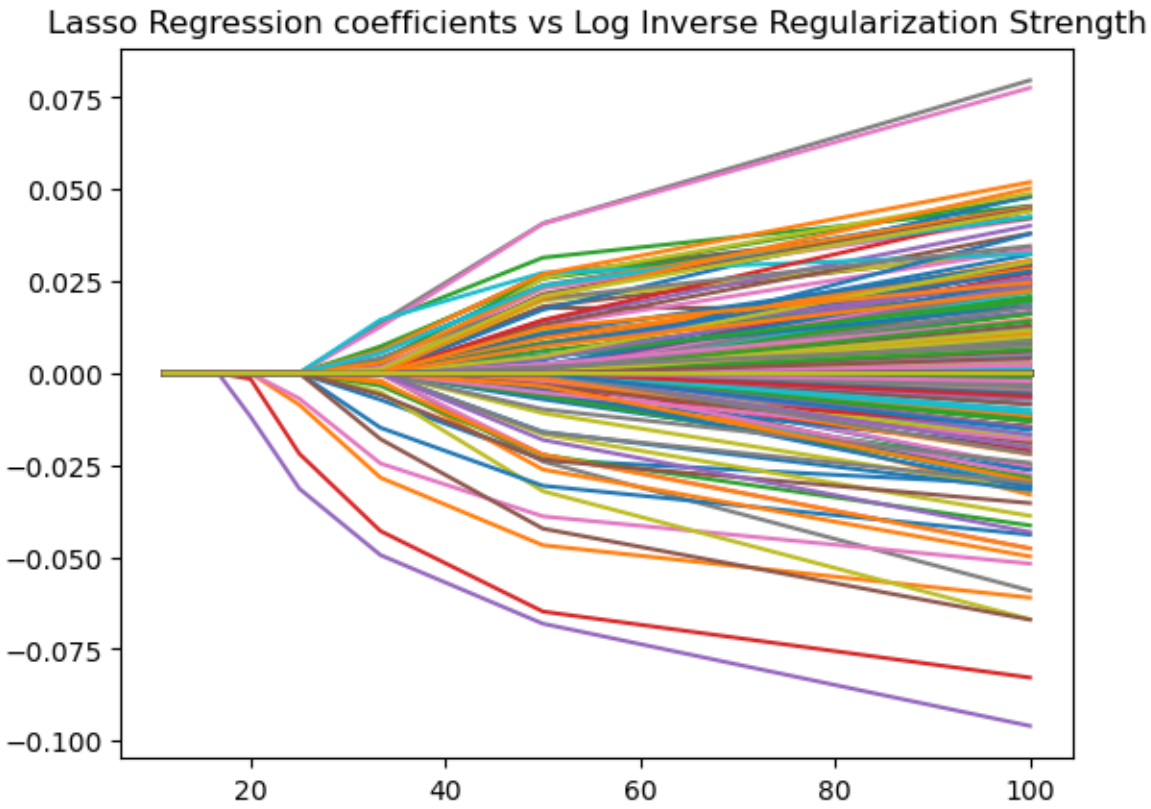


Figure 7: ROC curve for 300 selected subsets of variables



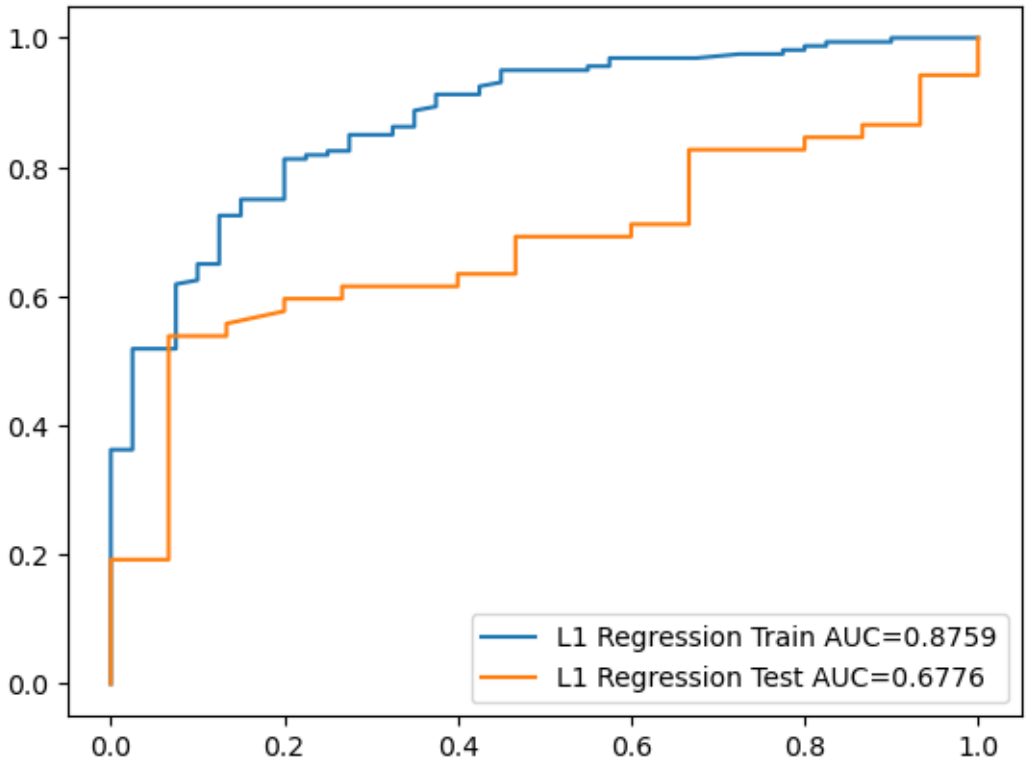
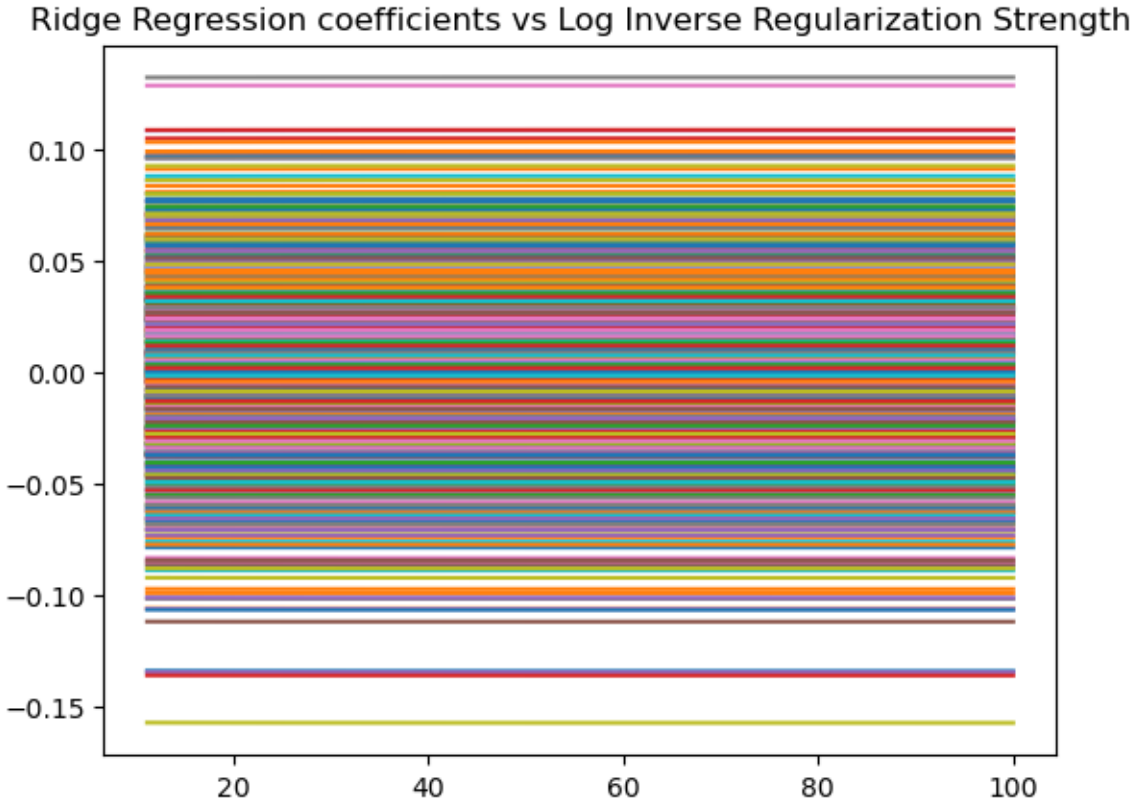


Figure 8: Best L1 Regression Train/Test ROC



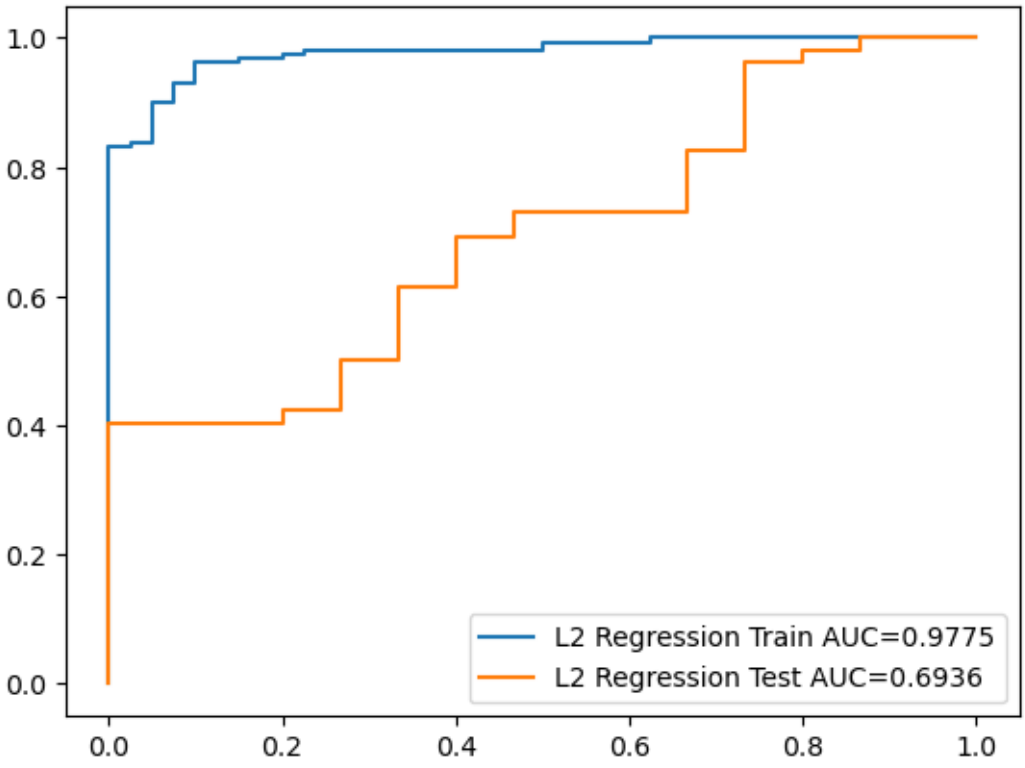


Figure 9: L2 Regression Train/Test ROC