Problem 1:

In Naive Bayes if $\Sigma$ is constrained to be diagonal then $P(X_j|Y)$ can be written as a product of $P(X_j|Y)$

$$P(X|Y) = \frac{1}{\sqrt{(2\pi)^D \det(\Sigma_Y)}} \exp\left(-\frac{1}{2}(X_j - \mu_{jY})^T \Sigma_Y^{-1} (X_k - \mu_{kY})\right)$$

$$= \prod_{j=1}^{D} \frac{1}{\sqrt{(2\pi)^D \sigma_{yjj}}} \exp\left(-\frac{1}{2\sigma_{yjj}} \|X_j - \mu_{jy}\|_2^2\right) = \prod_{j=1}^{D} P(X_j|Y)$$

So diagonal covariance matrix satisfies naive bayes assumption

Case 1: The covariance matrix is shared.

$$P(X|Y) = N(X|\mu_y, \Sigma)$$

$$P(X|Y=1) = P(X|Y=0)$$

$$\log \pi_1 - \frac{1}{2}(X-\mu_1)^T \Sigma^{-1}(X-\mu_1) = \log \pi_0 - \frac{1}{2}(X-\mu_0)^T \Sigma^{-1}(X-\mu_0)$$

$$c + x^T \Sigma^{-1} x - 2\mu_1^T \Sigma^{-1} x + \mu_1^T \Sigma^{-1} \mu_1 = x^T \Sigma^{-1} x - 2\mu_0^T \Sigma^{-1} x + \mu_0^T \Sigma^{-1}$$

$$[2(\mu_0 - \mu_1)^T \Sigma^{-1}] - x(\mu_0 - \mu_1)^T \Sigma^{-1}(\mu_0 - \mu_1) = c$$

$$\Rightarrow b_i x_i + c = 0$$

where, $b_i = 2(\mu_0 - \mu_1)^T \Sigma^{-1}$

$$c = -(\mu_0 - \mu_1)^T \Sigma^{-1}(\mu_0 - \mu_1)$$

This is a linear function

If the covariance is not shared.

$$x^T(\Sigma_1^{-1} - \Sigma_0^{-1})x - 2(\mu_1^T\Sigma_1^{-1} - \mu_0^T\Sigma_0^{-1})x + (\mu_0^T\Sigma_0\mu_0 - \mu_1^T\Sigma_1\mu_1) = C$$

$$\Rightarrow x^T a_i^\circ x + b_i x + c_i = 0$$

where. $a_i^\circ = \Sigma_1^{-1} - \Sigma_0^{-1}$

$$b_i^\circ = -2(\mu_1^T\Sigma_1^{-1} - \mu_0^T\Sigma_0^{-1})$$

$$c_i^\circ = (\mu_0^T\Sigma_0\mu_0 - \mu_1^T\Sigma_1\mu_1)$$

This is a quadratic function.

Problem 2

(a) $S_{m-1}(r)$

$= S_{2-1}(r)$ $[m=2]$

$= S_1(r)$

$= 2\pi r$

$V_m(r)$

$= V_2(r)$

$= \pi r^2$

$S_{m-1}(r)$

$= S_2(r)$ $[m=3]$

$= 4\pi r^2$

$V_m(r)$

$= V_3(r)$

$= \frac{4}{3}\pi r^3$

$V_0 = 1$

$V_1 = 2$

$S_0 = 2$

$S_1 = 2\pi$

$V_2 = \pi$

$S_2 = 4\pi$

$V_3 = \frac{4}{3}\pi$

(b) $V = \frac{4}{3}\pi r^3$

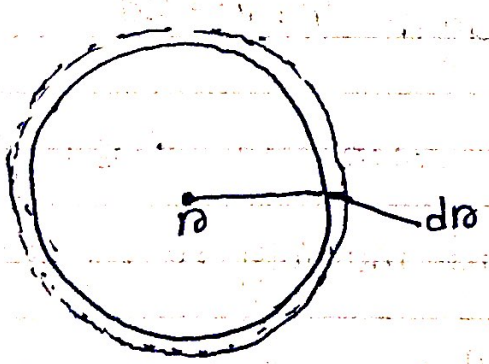
$\frac{d}{dx}[x^n] = nx^{n-1}$

$\frac{dv}{dr} = V'(r) = \frac{4}{3}\pi 3r^2$

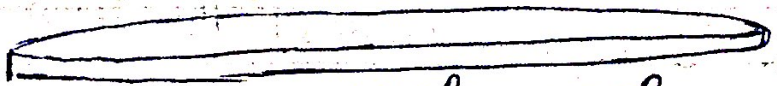
$\frac{dv}{dr} = V'(r) = 4\pi r^2$

$S = 4\pi r^2$

$\frac{dv}{dr}$ describes the change in volume with respect to the radius




A small increase in radius increase the volume of the circle
If we flatten this shell or layer created by the increase
radius it will produce a right circular cylinder




$A = \frac{4\pi r^2}{2} = 2\pi r^2$

Suppose a ball filled with air represent the shell or layer we want to find the volume of. If we begin to squish that and make that completely flattened it will look like a right circular cylinder. (In the previous page) which represents:

Top circle thickness = $dr$

Bottom " " = $dr$

Total thickness or height = $dr + dr = 2dr$

In order to find the volume we first calculate area of the base and multiply it by the height.

Area of the base of the right circular cylinder

$A_{base} = \frac{4\pi r^2}{2} = 2\pi r^2$ [half of the surface of the sphere]

So, $V = A_{base} . h = 2\pi r^2 . 2dr$

This is the changed volume of the sphere

So, $V' = 4\pi r^2 dr$

$\frac{dV}{dr} = 4\pi r^2$

So the derivative of the volume of a sphere is the surface area. Increasing the radius results in an increase in volume proportional to the surface area

(c) $S_{m-1}(r) = \overline{S}_{m-1} r^{m-1}$

$V_m = \frac{\overline{S}_{m-1}}{m}$, therefore $V_{m-2} = \frac{\overline{S}_{m-3}}{m-2}$ ---- (1)

$\overline{S}_{m-1} = 2\pi V_{m-2}$

$= \frac{2\pi \overline{S}_{m-3}}{m-2}$ [From eqn (1)]

$$\text{So, } S_{m-1}(r) = \frac{2\pi \overline{S}_{m-3}}{m-2} r^{m-1}$$

2(d)

$$P(x) = \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{\|x^2\|}{2\sigma^2}\right)$$

$$S_{m-1}(r) = \frac{2\pi S_{m-3}}{m-2} r^{m-1}$$

$$x \in S_{m-1}(r)$$

$$P_m(r) = \int_{x \in S_{m-1}(r)} P(x)\,dx$$

Things we need to do:

1. we have to chop up the region $S_{m-1}(r)$ into tiny pieces
2. Multiply the area of each piece $dx$, by the value of $P(x)$ at one of the points inside that piece
3. Add up the resulting values

So, $$P_m(r) = \int_{S_{m-1}(r)} \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{\|x^2\|}{2\sigma^2}\right) dx$$

But since P depends only on $x$ as well as $r$, it is constant on the spherical surface

So the equation $\rho_m(r)$ for the integrated density of sampled points from the gaussian distribution lying on the surface of $S_{m-1}(r)$:

$$\boxed{\rho_m(r) = r^{m-1} e^{-\frac{r^2}{2\sigma^2}}}$$

**2(e)** $\rho_m(r)$ will be maximum when

$$\frac{d\rho_m(r)}{dr} = 0$$

$$\Rightarrow e^{-\frac{r^2}{2\sigma^2}} (m-1) r^{m-1-1} - r^{m-1} e^{-\frac{r^2}{2\sigma^2}} \cdot \frac{2r}{2\sigma^2} = 0$$

$$\Rightarrow e^{-\frac{r^2}{2\sigma^2}} (m-1) r^{(m-2)} - \frac{r^m}{\sigma^2} e^{\frac{-r^2}{2\sigma^2}} = 0$$

$$\Rightarrow e^{-\frac{r^2}{2\sigma^2}} \left( (m-1) r^{m-2} - \frac{r^m}{\sigma^2} \right) = 0$$

$$\Rightarrow (m-1) r^{m-2} = \frac{r^m}{\sigma^2}$$

$$\Rightarrow r^{m-2-m} = \frac{1}{(m-1)\sigma^2}$$

$$\Rightarrow r^{-2} = \{ (m-1)\sigma^2 \}^{-1}$$

$$\Rightarrow r^2 = (m-1)\sigma^2$$

As $m$ is very large,

$m-1 \approx m$

So, $\boxed{\hat{r_0} = \sqrt{m}\sigma}$

**2(f)** $P(r_0) = r_0^{m-1} e^{-\frac{r_0^2}{2\sigma^2}}$

$\Rightarrow P(\hat{r_0}+\varepsilon) = (\hat{r_0}+\varepsilon)^{m-1} e^{\frac{-(\hat{r_0}+\varepsilon)^2}{2\sigma^2}}$

$\ln P(\hat{r_0}+\varepsilon) = (m-1)\ln(\hat{r_0}+\varepsilon) - \frac{(\hat{r_0}+\varepsilon)^2}{2\sigma^2} = S(\hat{r_0}+\varepsilon)$

Differentiating

$S'(\hat{r_0}+\varepsilon) = \frac{m-1}{\hat{r_0}+\varepsilon} - \frac{\hat{r_0}+\varepsilon}{\sigma^2}$

$S''(\hat{r_0}+\varepsilon) = -\frac{(m-1)}{(\hat{r_0}+\varepsilon)^2} - \frac{1}{\sigma^2} \leq -1$

$S'(\hat{r_0}+\varepsilon) = 0$ at $(\hat{r_0}+\varepsilon)^2 = (m-1)\sigma^2$

$\Rightarrow \hat{r_0}+\varepsilon = \sqrt{m}\sigma = \hat{r_0}$

$S''(r_0+\varepsilon) < 0$ for all $r_0$.

The taylor series expansion for $S(r+\varepsilon)$ is,

$$S(\hat{n}+\varepsilon) = S(\hat{n}) + S'(\hat{n})(\hat{n}+\varepsilon-\hat{n}) + \frac{1}{2}S''(\hat{n})(\hat{n}+\varepsilon-\hat{n})^2 + \cdots$$

Thus.

$$S(\hat{n}+\varepsilon) = S(\hat{n}) + S'(\hat{n})(\hat{n}+\varepsilon-\hat{n}) + \frac{1}{2}S''(\xi)(\hat{n}+\varepsilon-\hat{n})^2$$

For some point $\xi$ between $\hat{n}$ and $\hat{n}+\varepsilon$ since $S'(\hat{n}) = 0$, the second term vanishes

$$S(\hat{n}+\varepsilon) = S(\hat{n}) + \frac{1}{2}S''(\xi)(\hat{n}+\varepsilon-\hat{n})^2$$

Since the second derivative is always less than -1,

$$S(\hat{n}+\varepsilon) \leq S(\hat{n}) - \frac{1}{2}(\hat{n}+\varepsilon-\hat{n})^2$$

now $P(\hat{n}+\varepsilon) = e^{S(\hat{n}+\varepsilon)}$ Therefore

$$P(\hat{n}+\varepsilon) \leq e^{S(\hat{n}) - \frac{1}{2\sigma^2}(\hat{n}+\varepsilon-\hat{n})^2}$$

$$\therefore \boxed{P(\hat{n}+\varepsilon) \leq P(\hat{n})e^{-\varepsilon^2/2\sigma^2}}$$

2g

For a low dimension gaussian distribution most points are close to origin. For a high dimension gaussian distribution the mass is concentrated in a thin annulus of width $O(1)$ at radius $\sqrt{m}$, located outside the sphere.

2h

Probability density at origin:

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{(x-\mu)^2}{2\sigma^2}\right]$$

Probability density at one point on sphere $S_{m-1}(\hat{r})$

$$P(x) = \frac{1}{(2\pi)^{d/2}\sigma^d} \exp\left(-\frac{|x|^2}{2\sigma^2}\right)$$

one way to illustrate the vastness of high-dimensional space is comparing the volume:

The volume of a hypersphere:

$$\frac{2r^d \pi^{d/2}}{d\Gamma(d/2)}$$

where the volume of a hypercube:

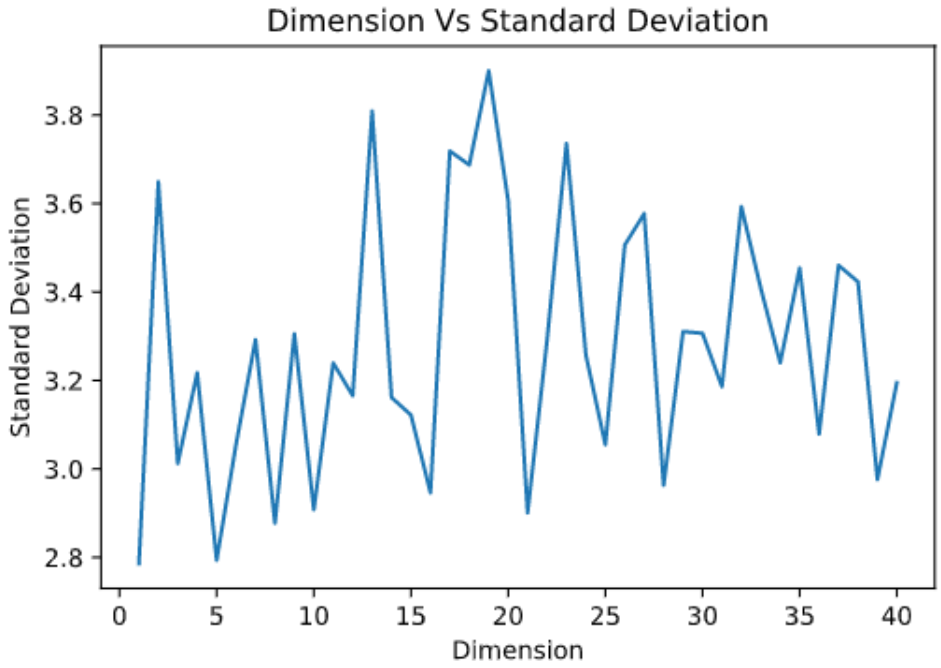
$2r$.

Comparing the proportions.

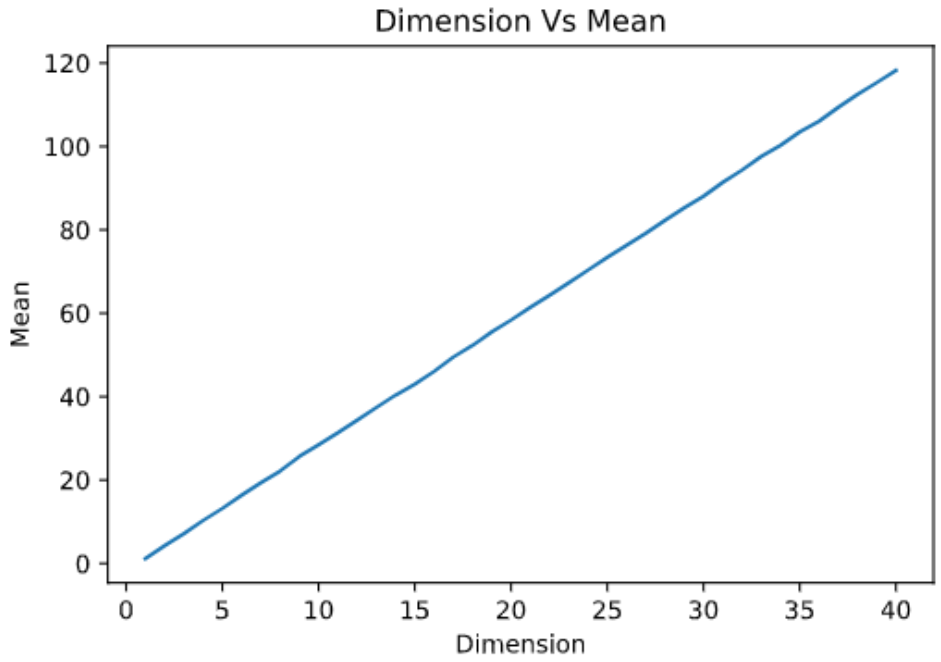
$$\frac{V_{hypersphere}}{V_{hypercube}} = \frac{\pi^{d/2}}{d\,2^{d-1}\Gamma(d/2)} \to 0 \text{ as } d \to \infty$$

# Problem 2(I)

Dimension Vs Standard Deviation

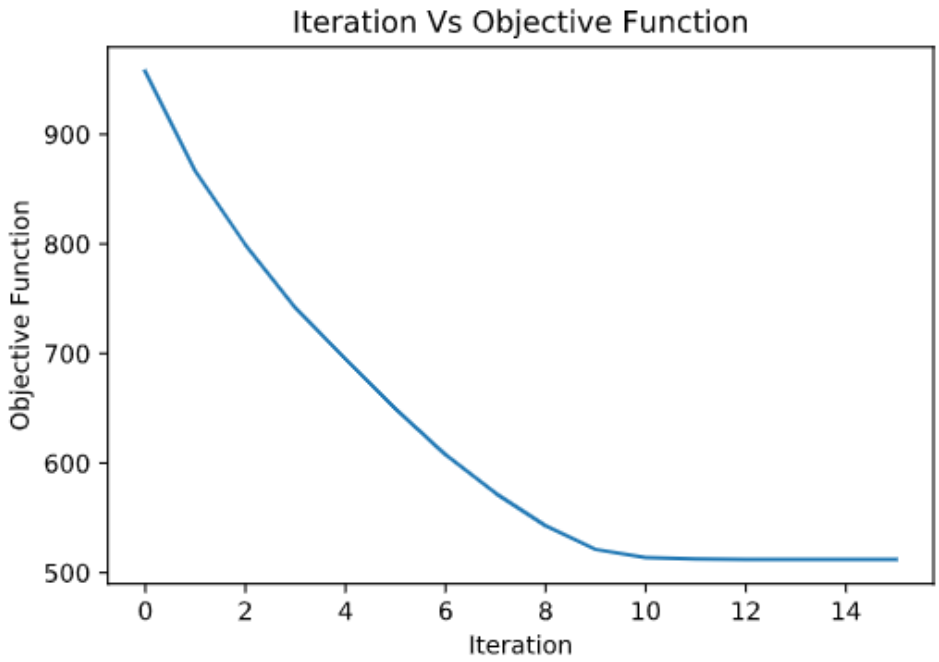



Dimension Vs Mean



The plot is consistent with our previous discussion. In high dimension, we measure the distance of gaussian samples using Euclidean distance. Therefore, each new dimension adds a non-negative term to the sum, so the distance increases with the number of dimensions for distinct vectors.
From the above figure, its proven that with the increase in dimensions, mean distance increases rapidly. As it is sampled from same gaussian distribution therefore the standard deviation is very closer.

# Problem 3(a)




Solving lasso on the generated synthetic data using the given parameters and reporting indices of non-zero weight entries: 0, 1, 2, 3, 4, 12, 16, 25, 34, 36, 52, 63, 71

# Problem 3(b)

Root Mean Square Error = 0.8591716751012152

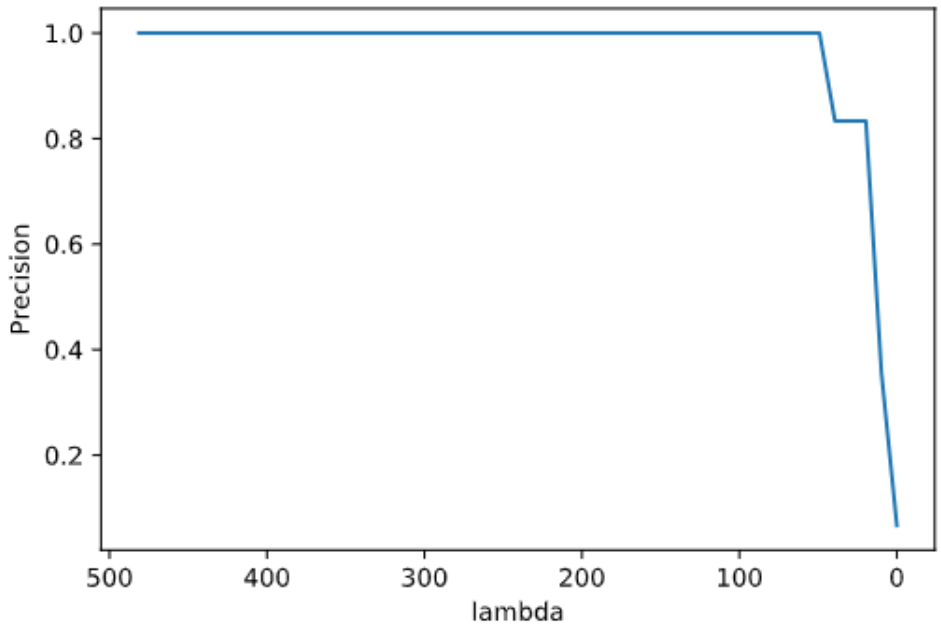
Sparsity = 13

Precision = 0.38461538461538464

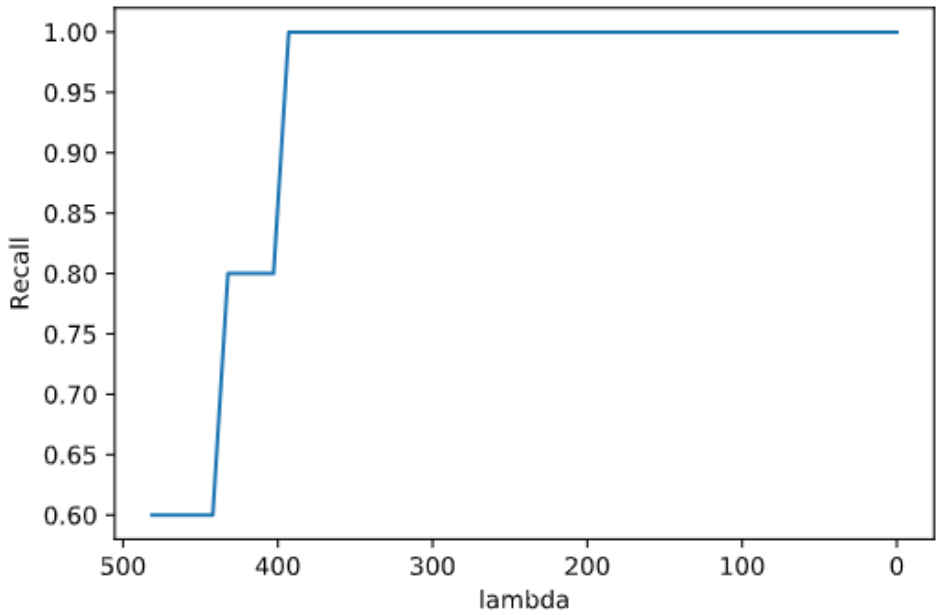
Recall = 1.0

# Problem 3(c)

Precision Vs Lambda



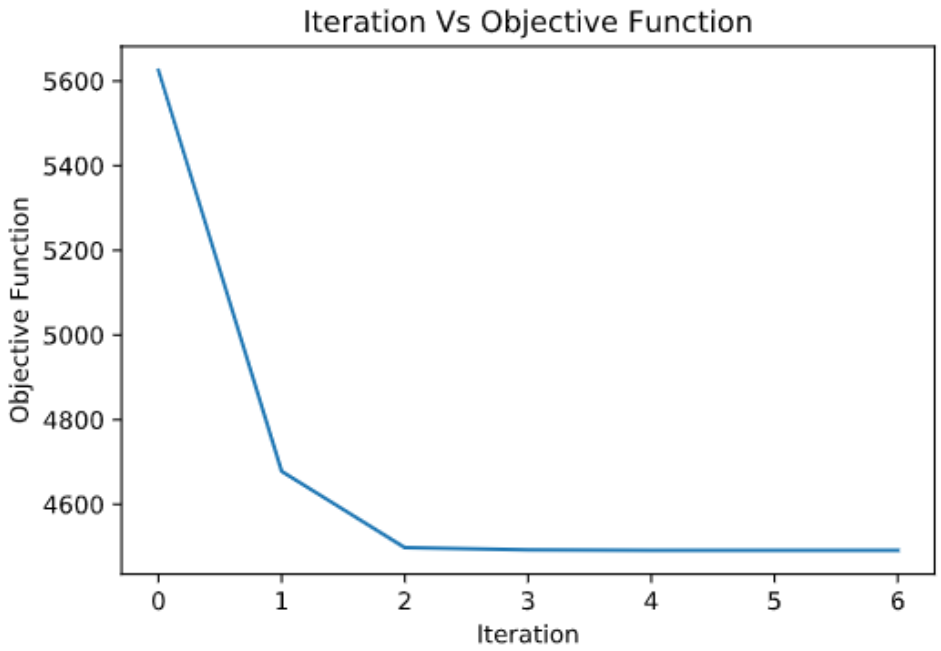

Recall Vs Lambda



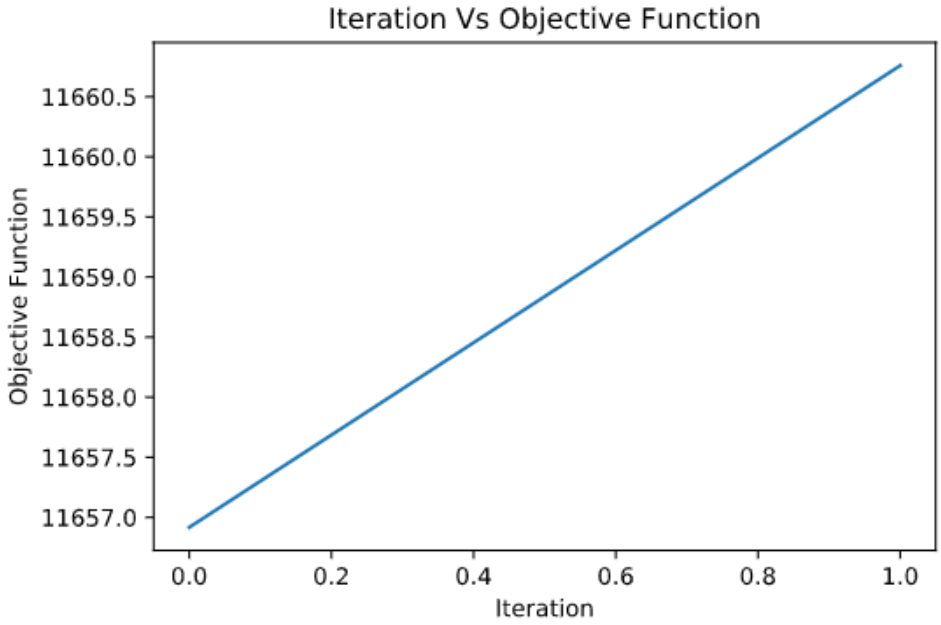

**From the above figure we can see that with the increase of lambda, precision increases and when the lambda is above 50 the precision becomes 1 and remains 1 until 500. In the case of recall, it is always 1 until approximately 400. After that it begins to drop. So, in order to get a good model, we should use the lambda between 50-400. Because after 400, it over shrinks some of the useful coefficients making them zero.**

## Playing with lambda and new discovery

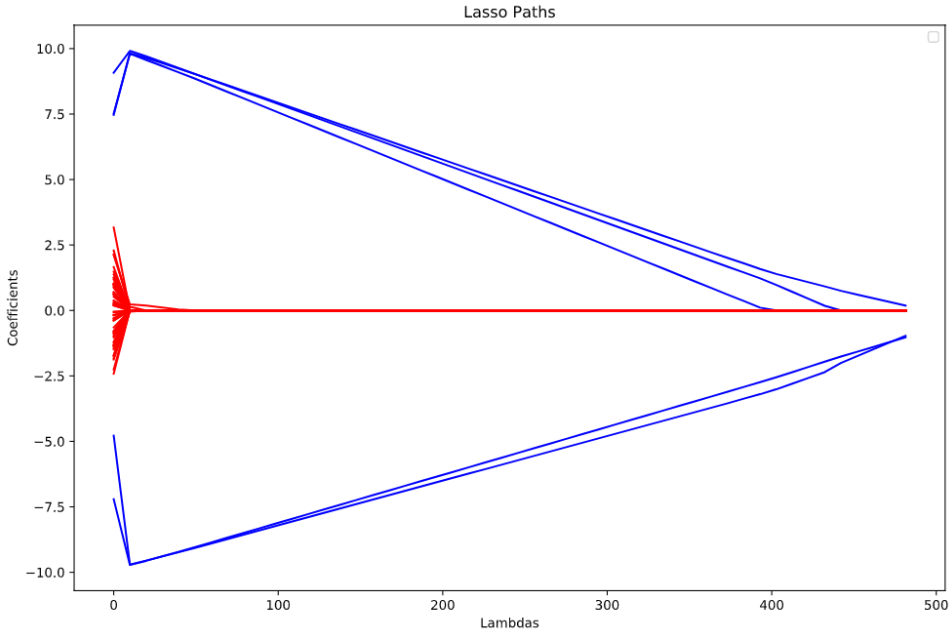
When lambda = 100, the model is taking less iterations to converge. So, the bigger the lambda the less iteration model takes to converge.



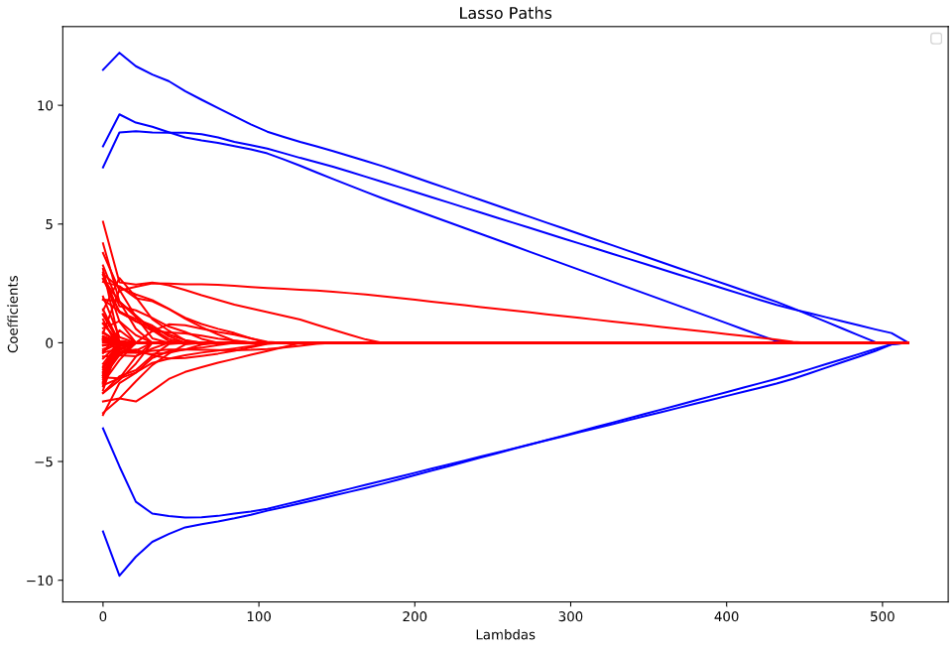

When lambda = 500, it is not converging anymore, it is linearly increasing.




Lasso Solution Path when sigma = 1.00



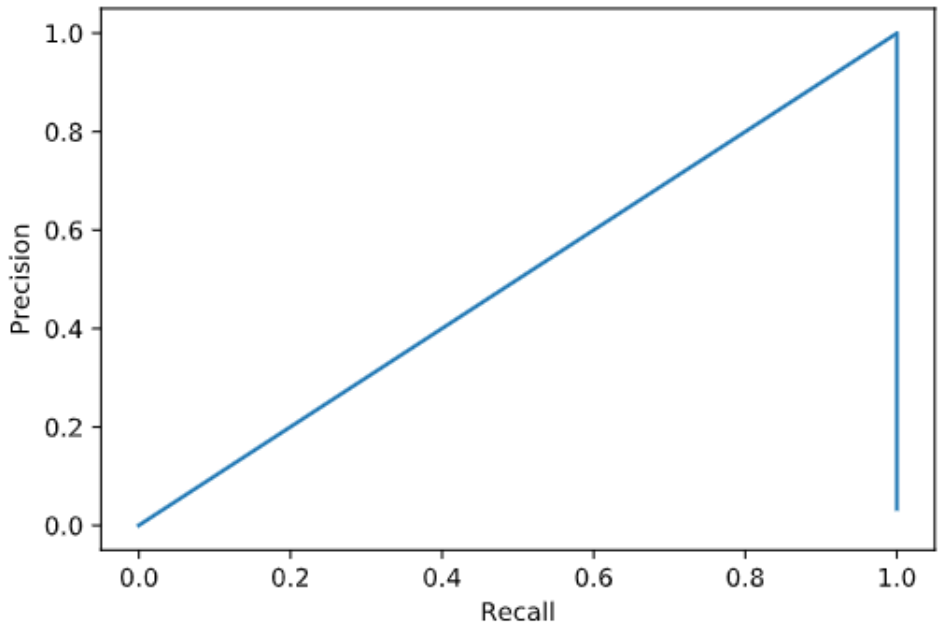

Lasso Solution Path when sigma= 10.00

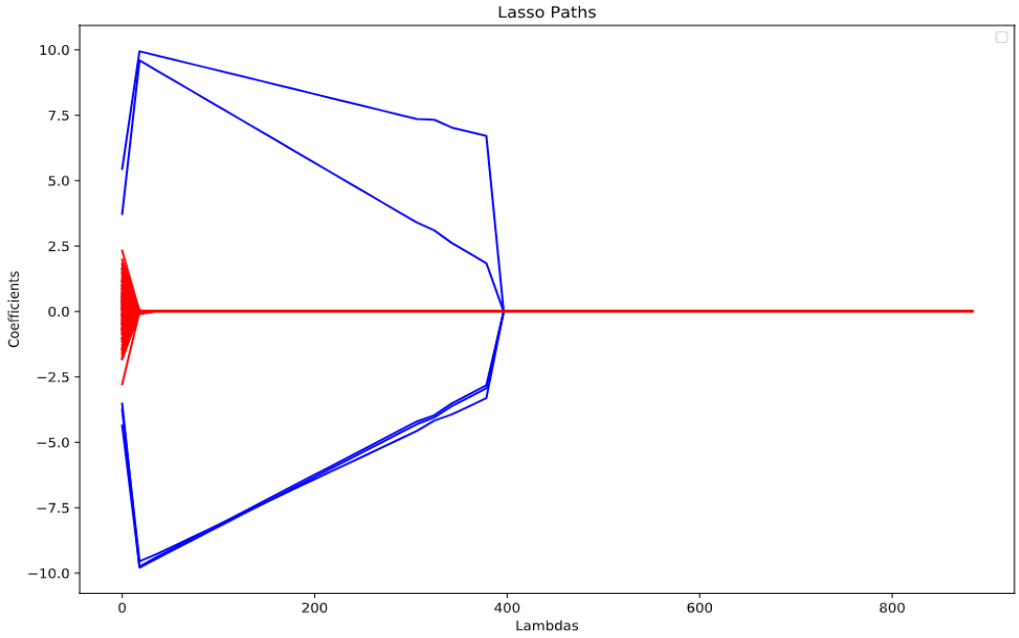



When sigma = 10, more specifically, the standard deviation of noise greater than before then the features other than the 5 features generated in DataGenerator collapses slowly to zero. That means the critical point is higher than sigma = 1.00 for 70 features. But for the rest of the 5 features it is inverse. Now the rest of the features quickly converge to zero than before. That meant the critical point is now lower than before.

**Problem 3(d):**

**n = 50; m = 150:**

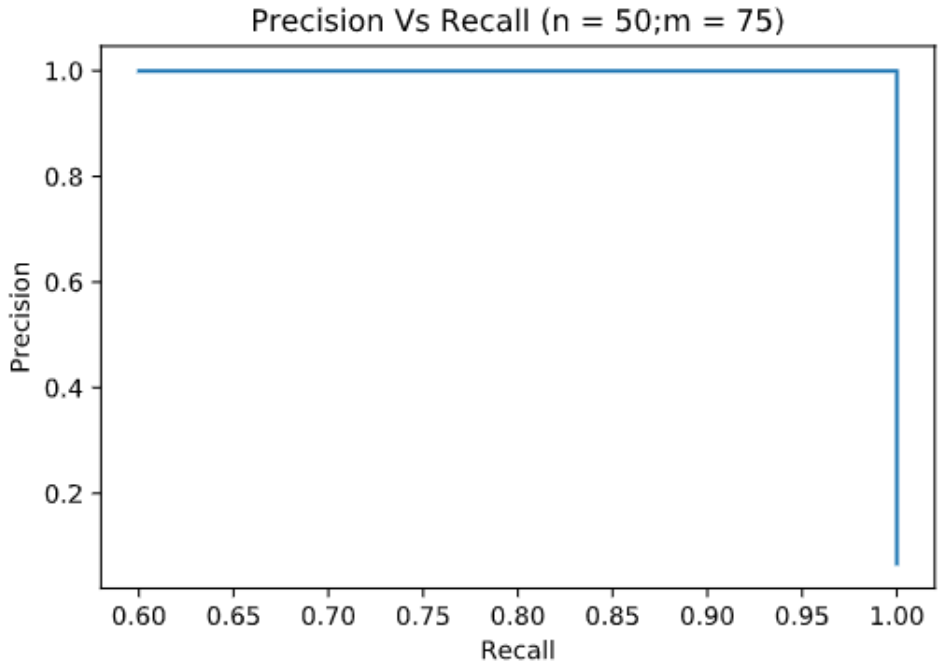


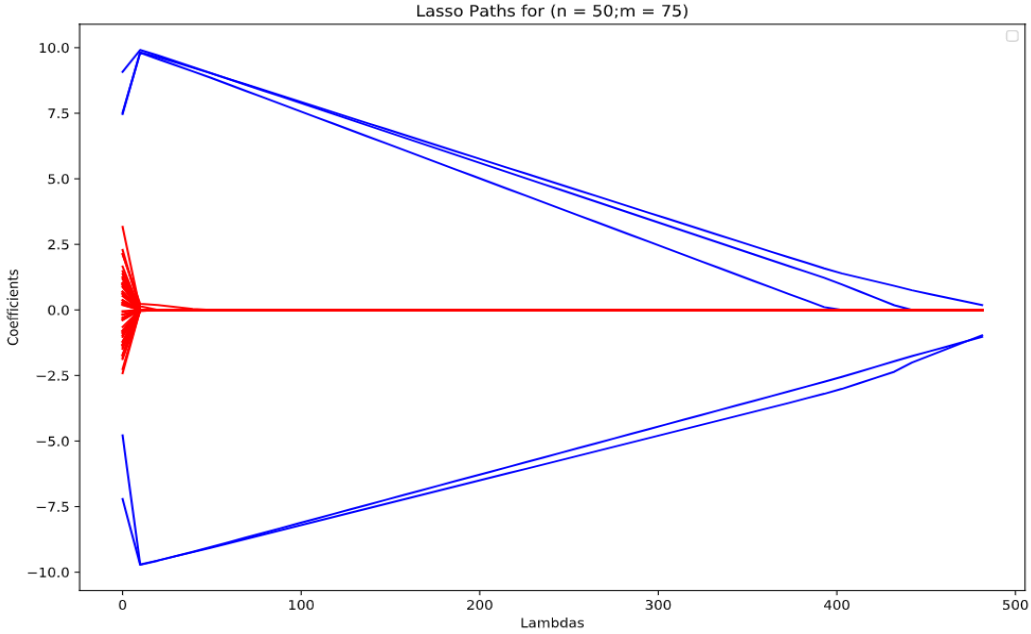




The lambda values that can generate good precision and recall (precision and recall both equals to 1):

36.03149416132465, 54.04724124198698, 72.0629883226493, 90.07873540331163, 108.09448248397396, 126.11022956463628, 144.1259766452986, 162.14172372596093, 180.15747080662325, 198.17321788728557, 216.18896496794792, 234.20471204861025, 252.22045912927257, 270.2362062099349, 288.2519532905972, 306.26770037125954, 324.28344745192186, 342.2991945325842, 360.3149416132465, 378.3306886939088

**n = 50; m = 75:**

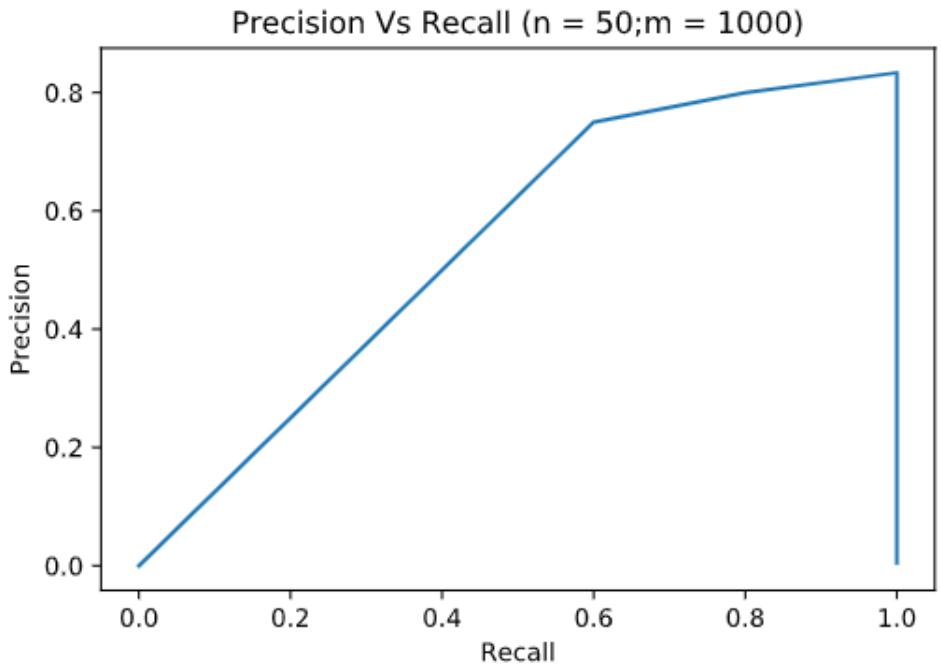


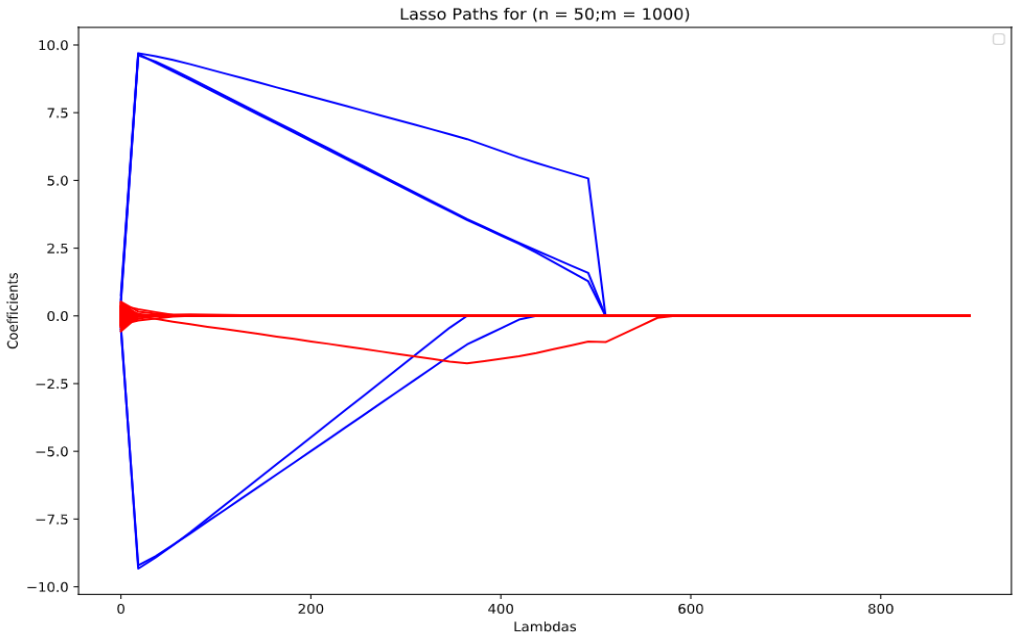



The lambda values that can generate good precision and recall (precision and recall both equals to 1):

49.1158315167013, 58.938997820041564, 68.76216412338182, 78.58533042672208, 88.40849673006234, 98.2316630334026, 108.05482933674286, 117.87799564008313, 127.70116194342339, 137.52432824676364, 147.3474945501039, 157.17066085344416, 166.99382715678442, 176.81699346012468, 186.64015976346494, 196.4633260668052, 206.28649237014545, 216.1096586734857, 225.93282497682597, 235.75599128016626, 245.57915758350651, 255.40232388684677, 265.22549019018703, 275.0486564935273, 284.87182279686755, 294.6949891002078, 304.51815540354806, 314.3413217068883, 324.1644880102286, 333.98765431356884, 343.8108206169091, 353.63398692024936, 363.4571532235896, 373.2803195269299, 383.10348583027013, 392.9266521336104

**n = 50; m = 1000:**

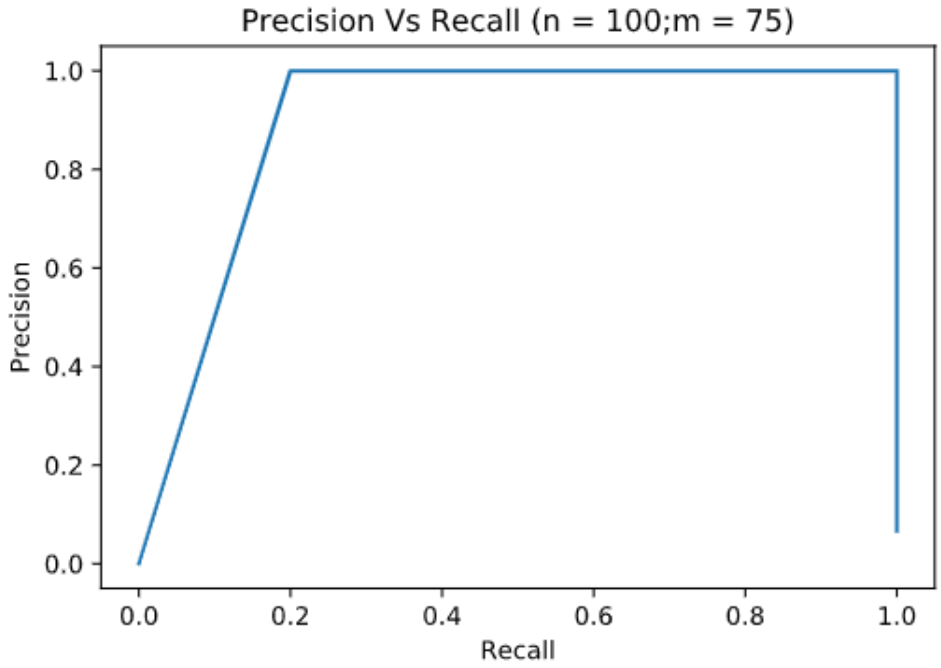


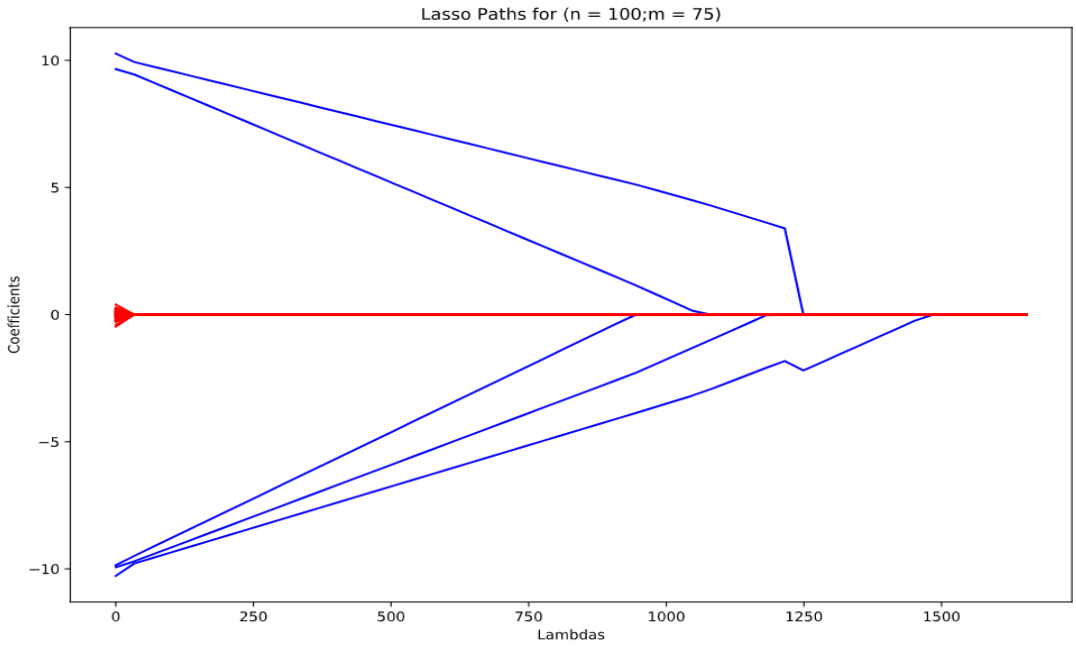



The lambda values that can generate good precision and recall (precision 0.8333333333333334 and recall 1):

218.6744295205359, 236.89729864724725, 255.12016777395857, 273.3430369006699, 291.56590602738123, 309.78877515409255, 328.01164428080386, 346.23451340751524

## n = 100; m = 75

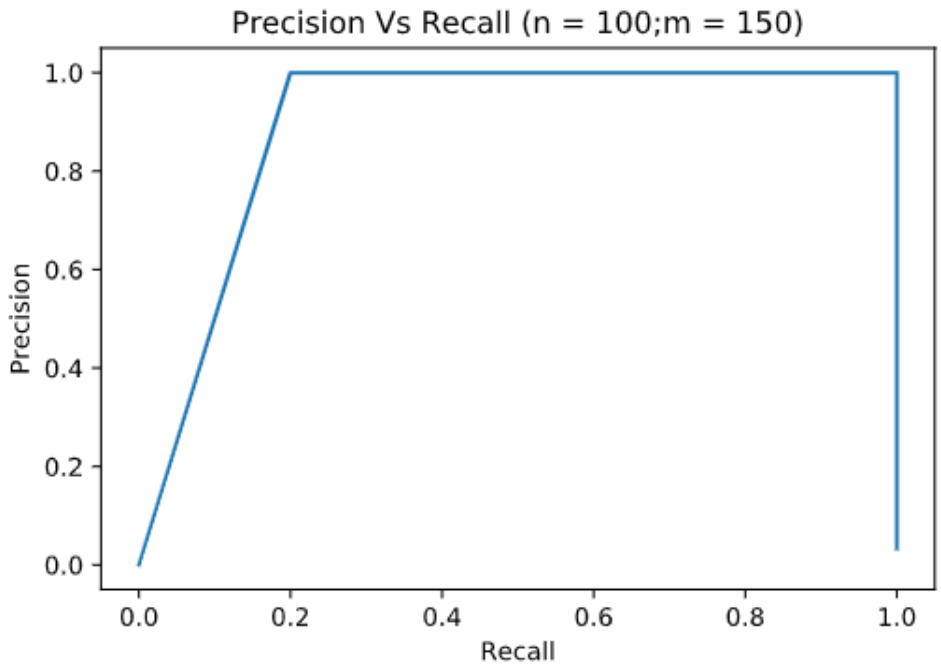


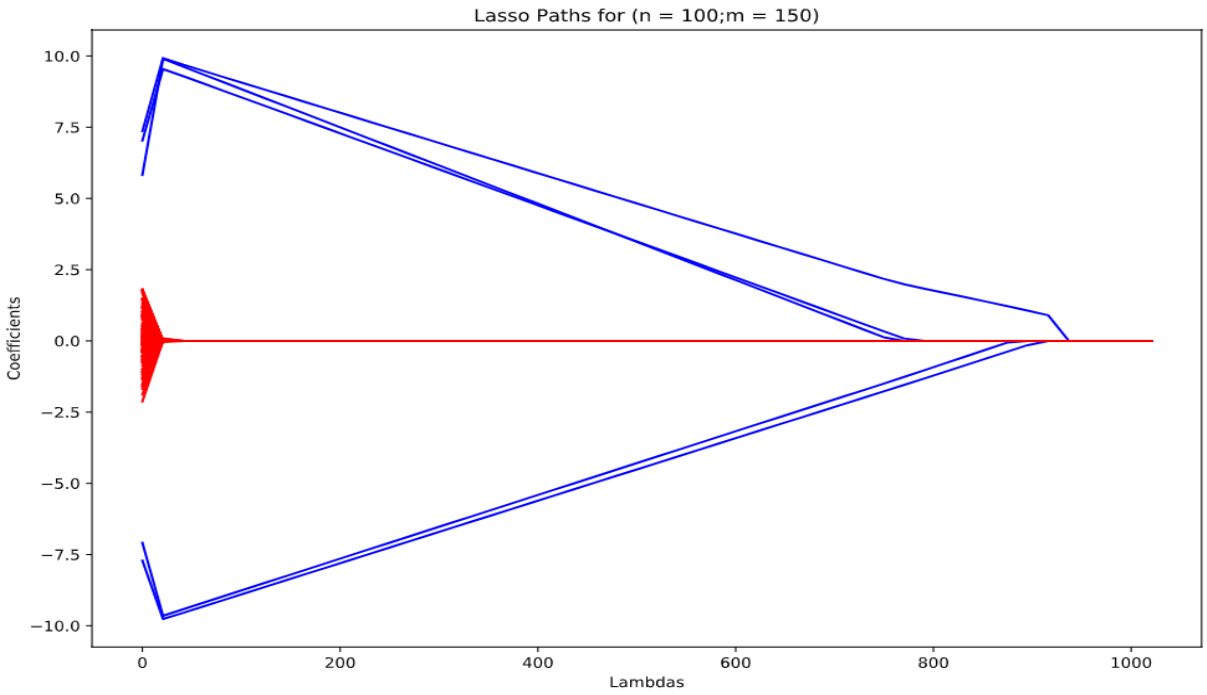



The lambda values that can generate good precision and recall (precision and recall both equals to 1):

33.75566209595288, 67.51132419190576, 101.26698628785863, 135.02264838381151, 168.7783104797644, 202.53397257571726, 236.28963467167014, 270.04529676762303, 303.8009588635759, 337.5566209595288, 371.31228305548166, 405.0679451514345, 438.82360724738743, 472.5792693433403, 506.3349314392932, 540.0905935352461, 573.8462556311989, 607.6019177271518, 641.3575798231047, 675.1132419190576, 708.8689040150105, 742.6245661109633, 776.3802282069162, 810.135890302869, 843.891552398822, 877.6472144947749, 911.4028765907277

# n = 100; m = 150



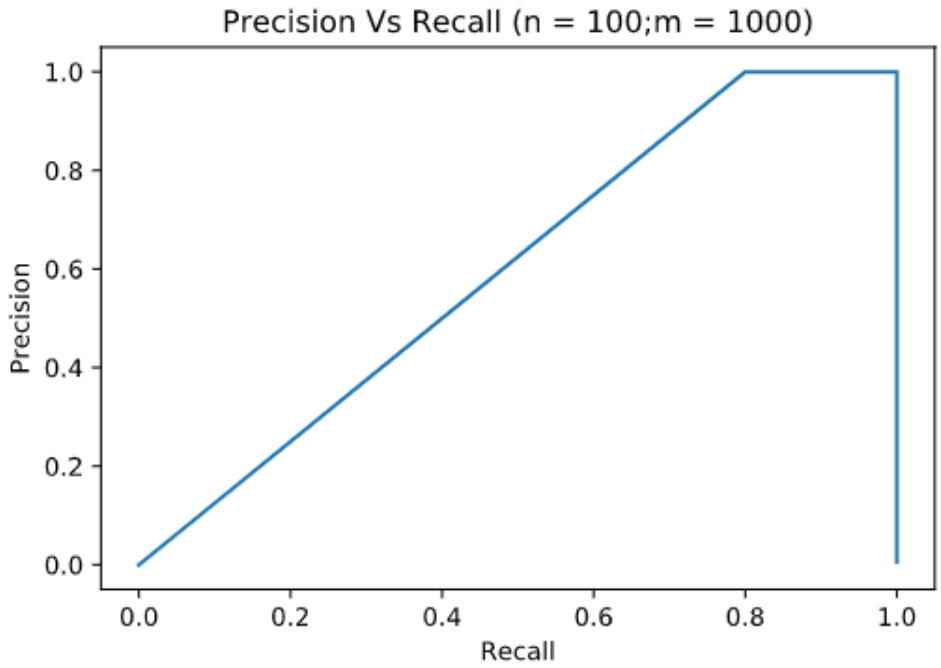



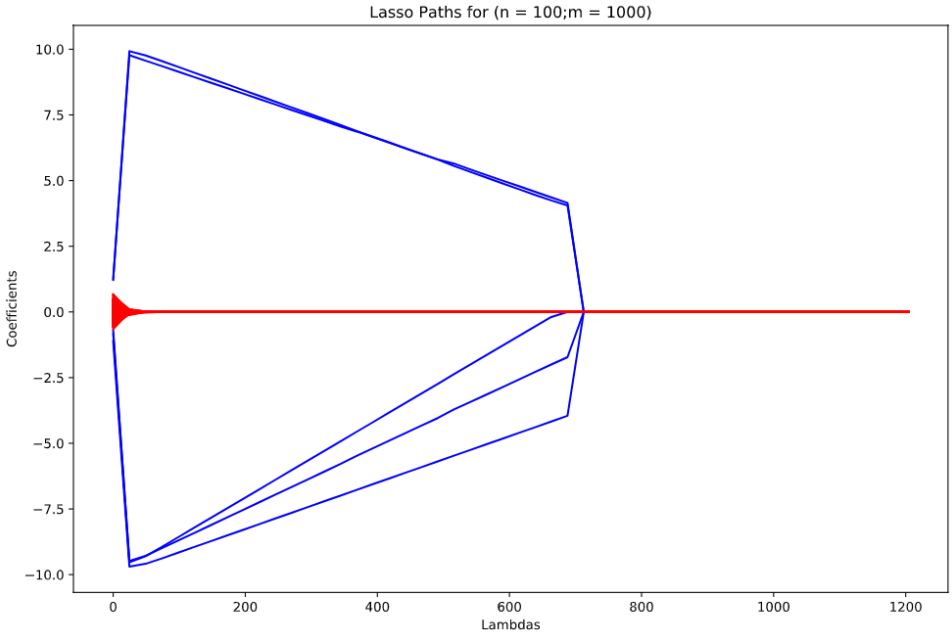

The lambda values that can generate good precision and recall (precision and recall both equals to 1):

41.64043199786918, 62.46064799680377, 83.28086399573836, 104.10107999467294, 124.92129599360754, 145.74151199254212, 166.56172799147672, 187.38194399041132,

208.2021599893459, 229.0223759882805, 249.8425919872151, 270.6628079861497, 291.48302398508423, 312.30323998401883, 333.12345598295343, 353.94367198188803, 374.76388798082263, 395.5841039797572, 416.4043199786918, 437.2245359776264, 458.044751976561, 478.8649679754956, 499.6851839744302, 520.5053999733648, 541.3256159722994, 562.1458319712339, 582.9660479701685, 603.7862639691031, 624.6064799680377, 645.4266959669723, 666.2469119659069, 687.0671279648415, 707.8873439637761, 728.7075599627107, 749.5277759616453

## n = 100; m = 1000





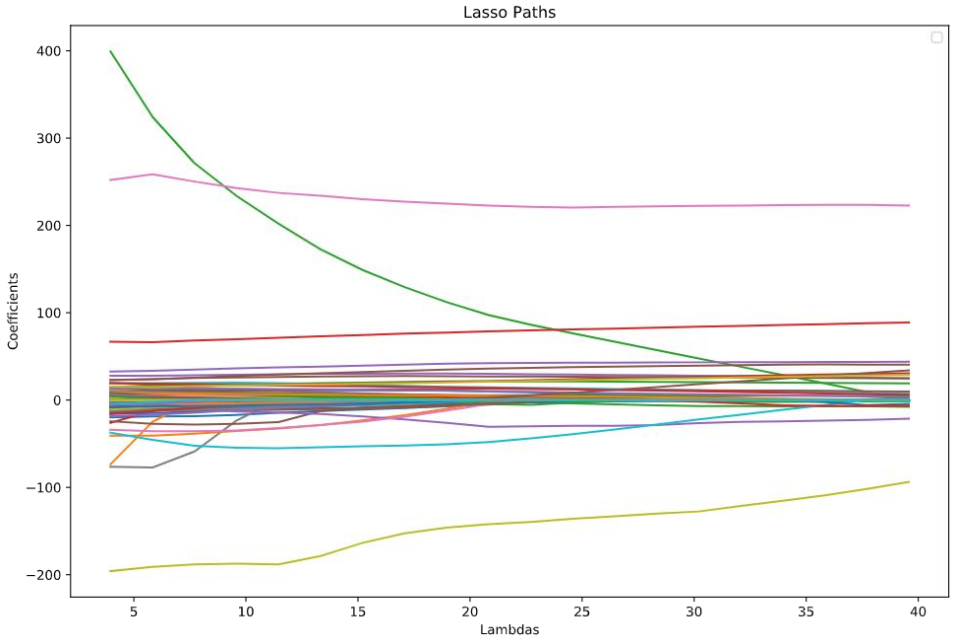

The lambda values that can generate good precision and recall (precision and recall both equals to 1):

73.70493900078691, 98.2732520010492, 122.8415650013115, 147.40987800157382, 171.9781910018361, 196.5465040020984, 221.1148170023607, 245.683130002623, 270.2514430028853, 294.81975600314763, 319.3880690034099, 343.9563820036722, 368.5246950039345, 393.0930080041968, 417.6613210044591, 442.2296340047214, 466.7979470049837, 491.366260005246, 515.9345730055084, 540.5028860057706, 565.0711990060329, 589.6395120062953, 614.2078250065575, 638.7761380068198, 663.3444510070821

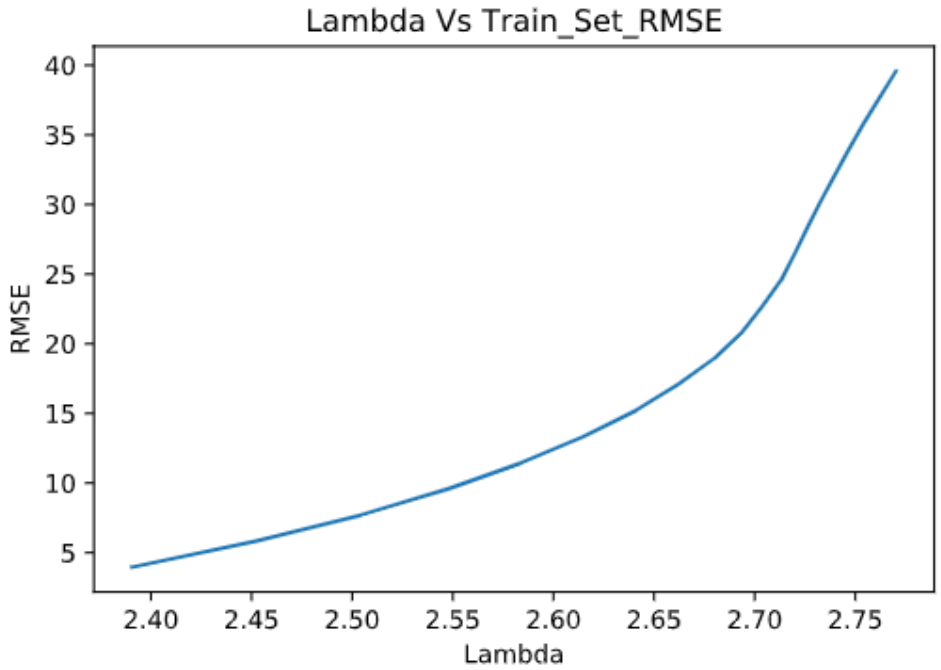
To get good precision and recall values, the most effective relationship between n and m is n = O(m^2), because the number of samples should be double of dimensions in an ideal machine learning model.

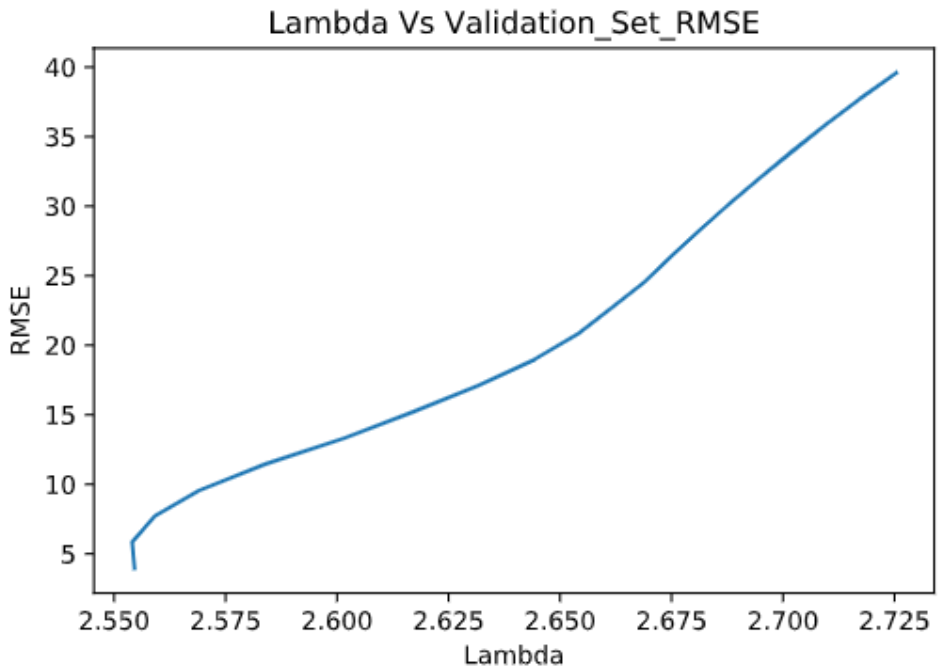
Problem 3(e):

Lasso solution Path




Lambda Vs RMSE






the best lambda value achieving the smallest validation RMSE found on the validation set, reporting

the corresponding test RMSE: 2.6

Report the top-10 features with the largest magnitude in the lasso solution w

when using the best lambda value,

Lasso select features:

the 324.3495809547862
and 258.43971011321804
were soaked in -190.9096607569402
sometime -77.28224385390585
great 66.3497495065244
set -45.47049422842977
were -40.82739778646367
of a -35.8037388362437
best 33.37412766738183
amazing 27.907588327908446

I don't think the selected features are so meaningful. While researching on the fact what's going on, my intuition said that we could do better while choosing the lambda values. The best selected lambda is very small around 5, choosing better regularizer range would make the model more effective.