ELEN 4720: Machine learning for Signals, Information & Data

Homework-2

Date - 31st October 2021

Name- Rakshith Kamath UNI- rk3165

Problem-1

$(y_1, x_1), \ldots (y_n, x_n).$ $y \in \{0,1\}$, $x \in R^d$

$$y_0 = \arg\max_y p(y_0 = y|\pi) \cdot \prod_{d=1}^{D} p(x_{0,d} | \lambda_{y,d})$$

$y_i \overset{iid}{\sim} \text{Bern}(\pi)$, $x_{i,d}|y_i \sim \text{Pois}(\lambda_{y_i,d})$, $d = 1, \ldots D$, Prior: $\lambda_{y,d} \overset{iid}{\sim} \text{Gamma}(2,1)$

$$\hat{\pi}, \hat{\lambda}_{0,1:D}, \hat{\lambda}_{1,1:D} = \arg\max_{\hat{\pi}, \hat{\lambda}_{0,1:D}, \hat{\lambda}_{1,1:D}} \sum_{i=1}^{n} \ln p(y_i|\pi) + \sum_{d=1}^{D} \left(\ln p(\lambda_0, d) + \ln p(\lambda_1, d) + \sum_{i=1}^{n} \ln p(x_{i,d} | \lambda_{y_i,d}) \right)$$

$= L$

a) For $\hat{\pi}$, w.r.t $p(y_i|\pi) = \pi^{y_i}(1-\pi)^{(1-y_i)}$

to get the max value of $\hat{\pi}$, we take the derivative of L w.r.t $\hat{\pi}$ & equate it to zero.

$$\frac{\partial L}{\partial \hat{\pi}} = \frac{\partial}{\partial \hat{\pi}}\left[\sum_{i=1}^{n} \ln p(y_i|\pi) + C\right]$$

C → terms which are constant w.r.t to $\hat{\pi}$, i.e they don't have a $\hat{\pi}$ term in it.

$$\frac{\partial L}{\partial \hat{\pi}} = \frac{\partial}{\partial \hat{\pi}}\left[\sum_{i=1}^{n}\left(y_i \log \pi + (1-y_i)\log(1-\pi)\right) + C\right]$$

$$= \frac{\sum_{i=1}^{n} y_i}{\pi} - \frac{\sum_{i=1}^{n}(1-y_i)}{(1-\pi)} = 0$$

$$\frac{\sum_{i=1}^{n} y_i}{\pi} + \frac{\sum_{i=1}^{n} y_i}{(1-\pi)} - \frac{n}{(1-\pi)} = 0$$

$$\sum_{i=1}^{n} y_i \cdot \frac{1}{\pi\cancel{(1-\pi)}} = \frac{n}{\cancel{(1-\pi)}}$$

$$\therefore \quad \boxed{\hat{\pi} = \frac{1}{n} \sum_{i=1}^{n} y_i}$$

b) $\lambda_{y,d} \sim \text{gamma}(2,1)$

$$P(\lambda_{y,d}) = \frac{1^2}{\Gamma[2]} \cdot (\lambda_{y,d})^1 \cdot e^{-\lambda_{y,d}}$$

$$\underline{\underline{\Gamma[2] = 1}}$$

$$P(\lambda_{y,d}) = \lambda_{y,d} \cdot e^{-\lambda_{y,d}}$$

Also, $P(x_{i,d} \mid y_i) = \text{Pois}(\lambda_{y_i,d})$

$$\text{Pois}(x \mid \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

$$\underline{\underline{= \frac{(\lambda_{y_i,d})^{x_{i,d}} \cdot e^{-\lambda_{y_i,d}}}{(x_{i,d})!}}}$$

Let's rewrite L as follows -

$$L = C_2 + \sum_{d=1}^{D} \Big(\ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \sum_{i=1}^{n} y_i \cdot \ln p(x_{i,d} \mid \lambda_{y,d}) + (1-y_i) \ln p(x_{i,d} \mid \lambda_{y_0,d}) \Big).$$

C_2 is the term which is constant w.r.t $\lambda_{0,1:D}$ & $\lambda_{1,1:D}$.

The last term is modified such that $y_i=1$ can be used to indicate $\lambda_{y_1,d}$ & $y_i=0$ to indicate $\lambda_{y_0,d}$.

Now, differentiating L w.r.t $\lambda_{0,d}$, we get

$$\frac{dL}{d\hat{\lambda}_{0,d}} = \frac{d}{d\hat{\lambda}_{0,d}}\Big[C' + \sum_{d=1}^{D} \log(\lambda_{0,d}) - \lambda_{0,d} + \log(\lambda_{1,d}) - \lambda_{1,d} +$$

$$\sum_{i=1}^{n} y_i \left(x_{i,d}\log(\lambda_{1,d}) - \lambda_{1,d} - \log(x_{i,d})! \right) +$$

$$(1-y_i)\left(x_{i,d}\log(\lambda_{0,d}) - \lambda_{0,d} - \log(x_{i,d})! \right) \Big]$$

$$0 = \frac{1}{\lambda_{0,d}} - 1 + \sum_{i=1}^{n} (1-y_i) \cdot \left[\frac{x_{i,d}}{\lambda_{0,d}} - 1 \right]$$

$$1 + \sum_{i=1}^{n}(1-y_i) = \frac{1}{\lambda_{0,d}}\left[1 + \sum_{i=1}^{n}(1-y_i) \cdot x_{i,d} \right]$$

$$\boxed{\hat{\lambda}_{0,d} = \frac{\left[1 + \sum_{i=1}^{n} (1-y_i) \cdot x_{i,d}\right]}{1 + \sum_{i=1}^{n} (1-y_i)}}$$

$|||^{ly}$ for $\lambda_{1,d}$, we get

$$0 = \frac{1}{\lambda_{1,d}} - 1 + \sum_{i=1}^{n} y_i \left[\frac{x_{i,d}}{\lambda_{1,d}} - 1\right]$$

$$\boxed{\hat{\lambda}_{1,d} = \frac{1 + \sum_{i=1}^{n} y_i \cdot x_{i,d}}{1 + \sum_{i=1}^{n} y_i}}$$

$$\boxed{\lambda_{y,d} = y\left[\frac{1 + \sum_{i=1}^{n} y_i \cdot x_{i,d}}{1 + \sum_{i=1}^{n} y_i}\right] + (1-y)\left[\frac{1 + \sum_{i=1}^{n} (1-y_i) \cdot x_{i,d}}{1 + \sum_{i=1}^{n} (1-y_i)}\right]}$$

This gives us the combined output $\lambda_{y,d}$. combining $\lambda_{y_0,d}$ & $\lambda_{y_1,d}$.

Problem-2

a)

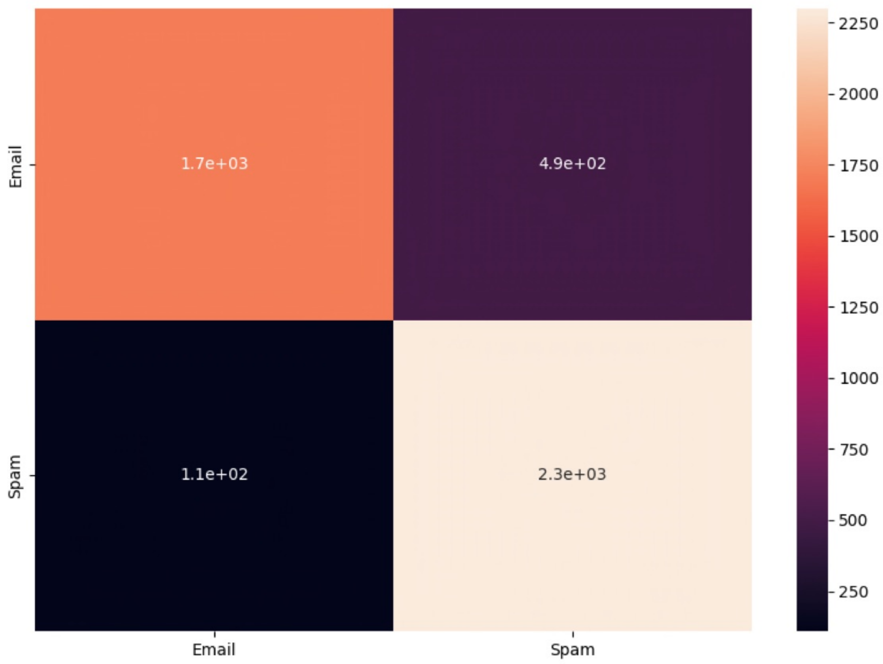
	Predicted $y=0$	Predicted $y=1$
Actual $y=1$	1703	488
Actual $y=0$	110	2299

Predicted accuracy =

$$\left(\frac{1703+2299}{4600}\right) \times 100$$

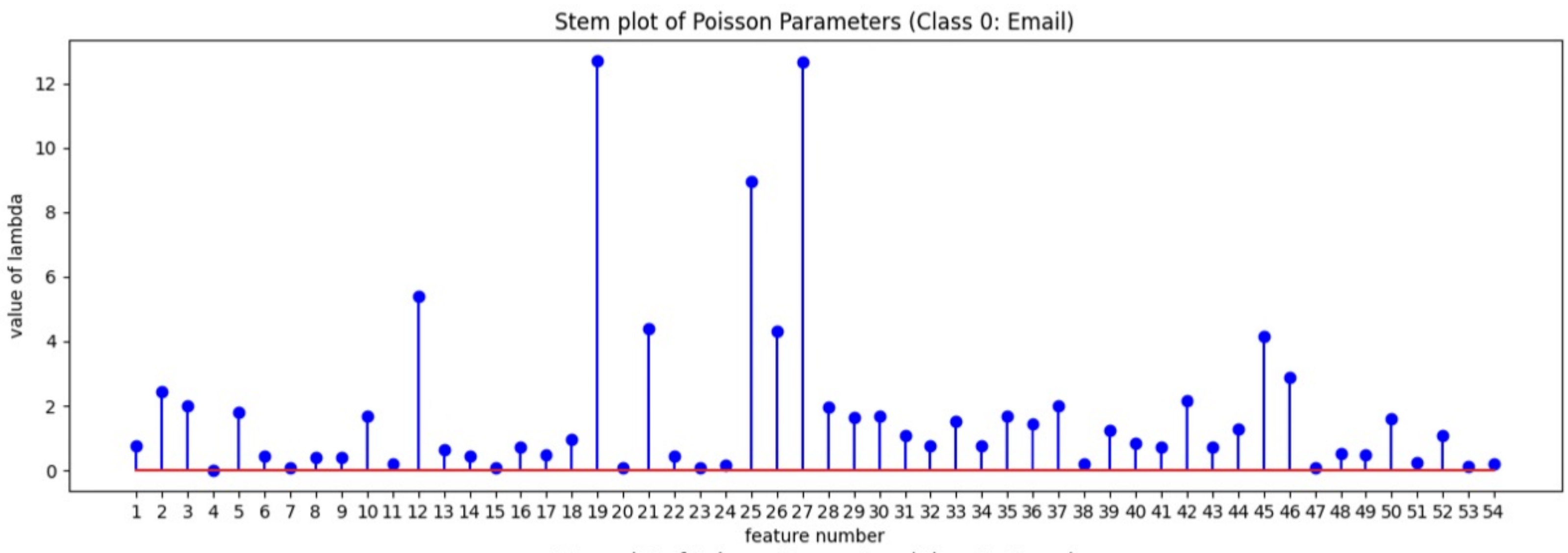
$$= 0.87 \times 100$$

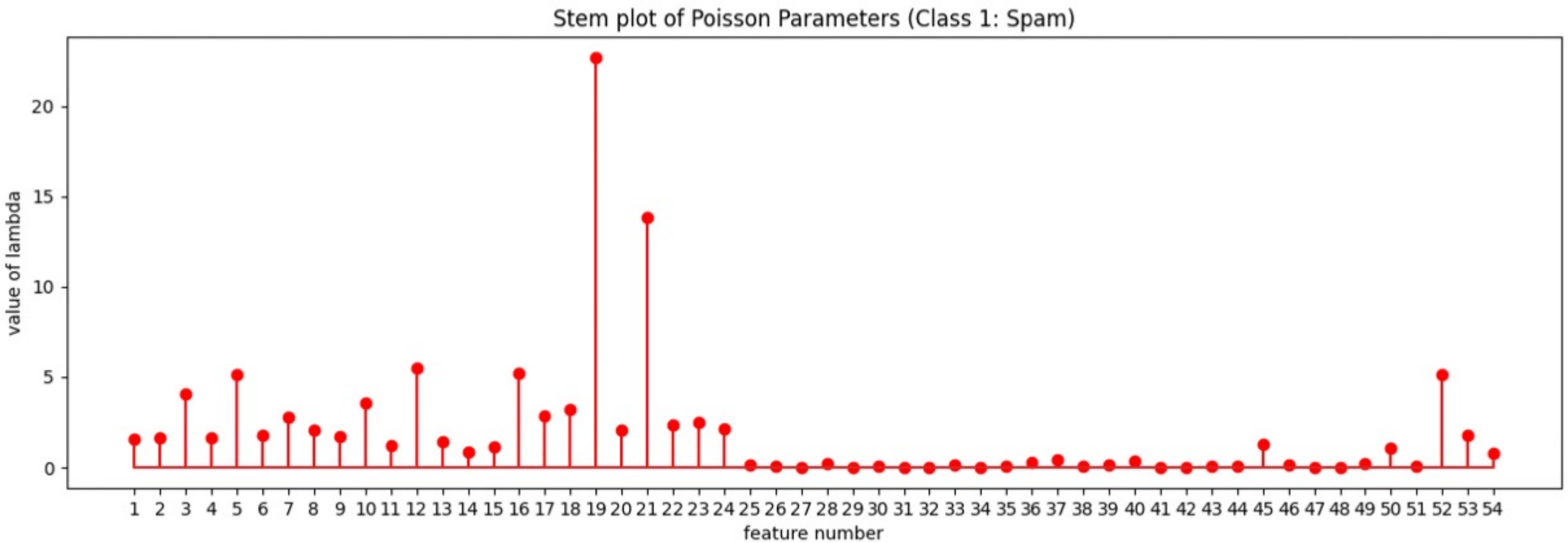
$$= 87\%$$



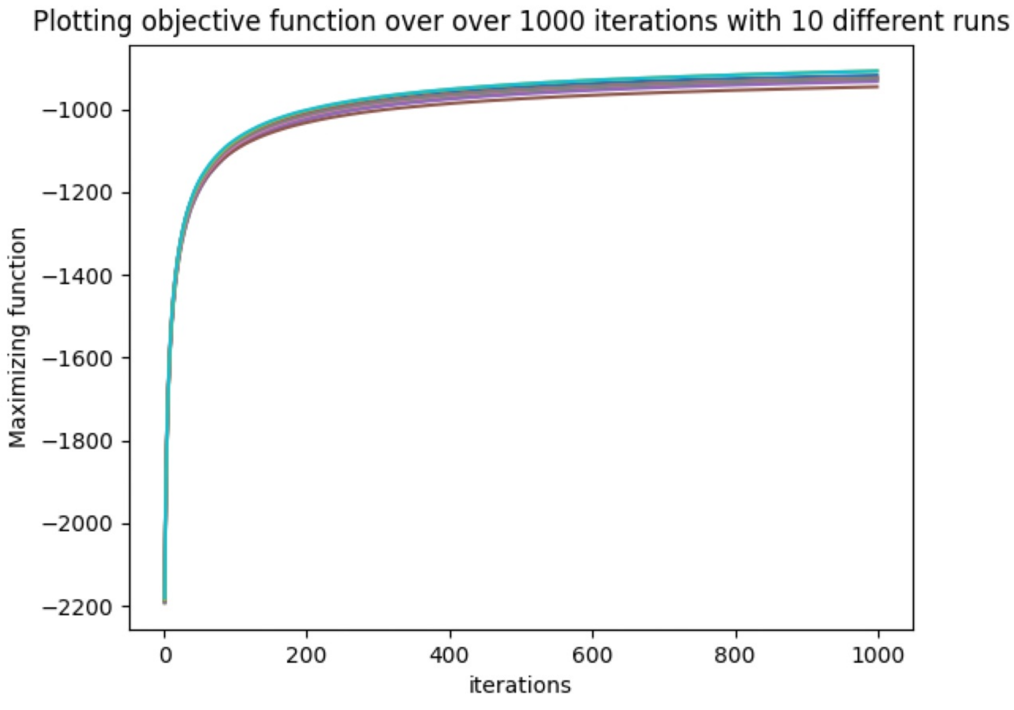
b) The dimension 16 represents the word "free" & dimension 52 represents "!".

From the plot below, we can see that both the dimensions the value of λ is higher in spam mails than the non-spam ones. This tells us that the words "free" & "!" are frequently seen in spam mails which usually is the case, when the companies try to market something they try to give in offers they usually tend to exaggerate & use the above words. Hence it's frequency is rightly so, seen in spam mails.





c) After running the algorithm 10 times, we get the following graph.



d) $L'(\omega) = L(\omega_t) + (\omega-\omega_t)^T \nabla L(\omega_t) + \frac{1}{2}(\omega-\omega_t)^T \nabla^2 L(\omega_t) \cdot (\omega-\omega_t)$

$$\frac{\partial L'(\omega)}{\partial \omega} = 0 + \nabla L(\omega_t) \cdot 1 + \frac{1}{2} \times 2 (\omega-\omega_t) \cdot \nabla^2 L(\omega_t).$$

equating this to zero, we get

$$0 = \nabla L'(\omega_t) + (\omega-\omega_t) \cdot \nabla^2 L(\omega_t).$$

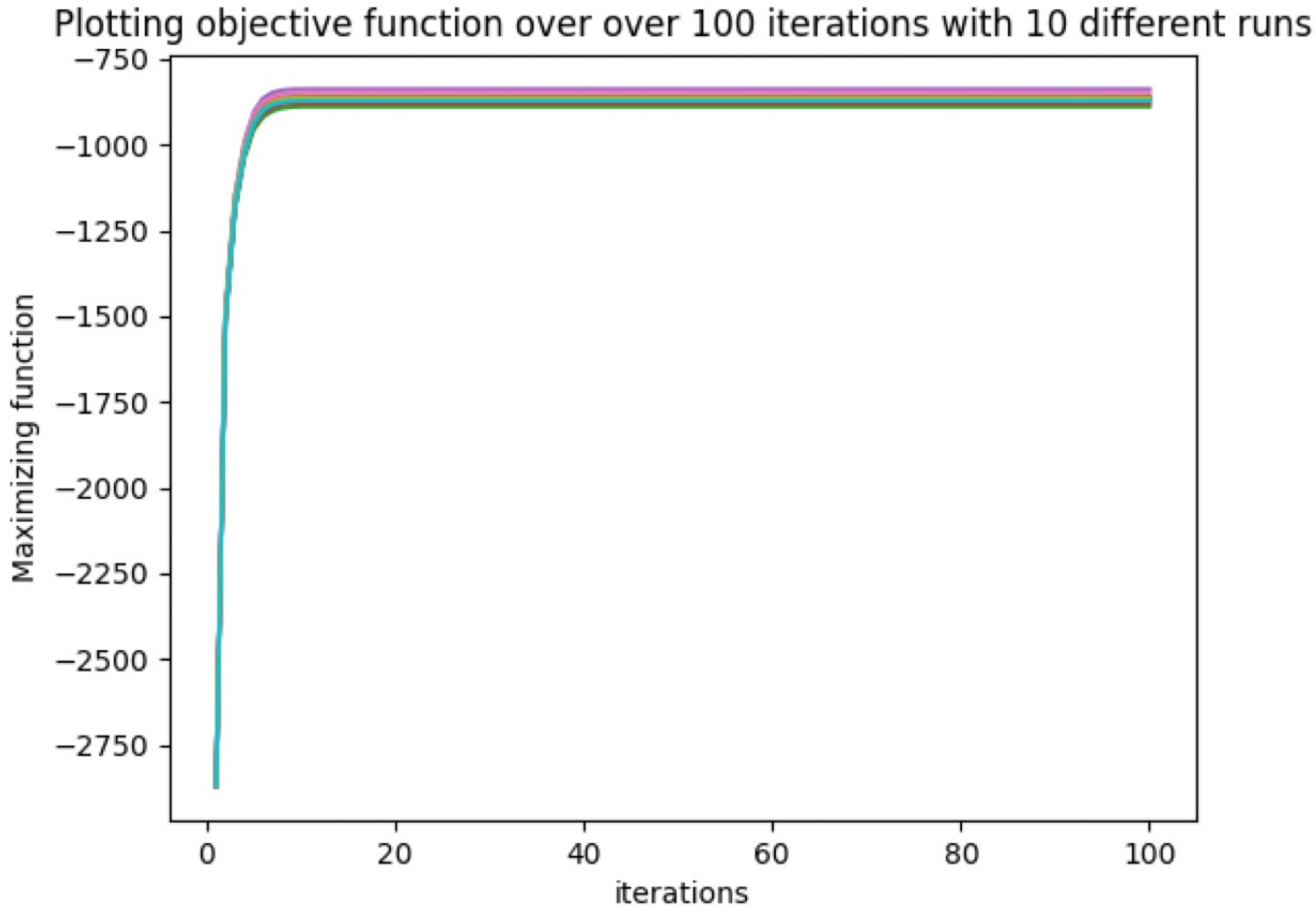
Rearranging we get

$$\omega = \omega_t - \frac{\nabla L(\omega_t)}{\nabla^2 L(\omega_t)},$$

i.e

$$\omega_{t+1} = \underline{\underline{\omega_t - \left[\nabla^2 L(\omega_t)\right]^{-1}}} \nabla L(\omega_t).$$

The graph after running it for 10 runs is as follows-



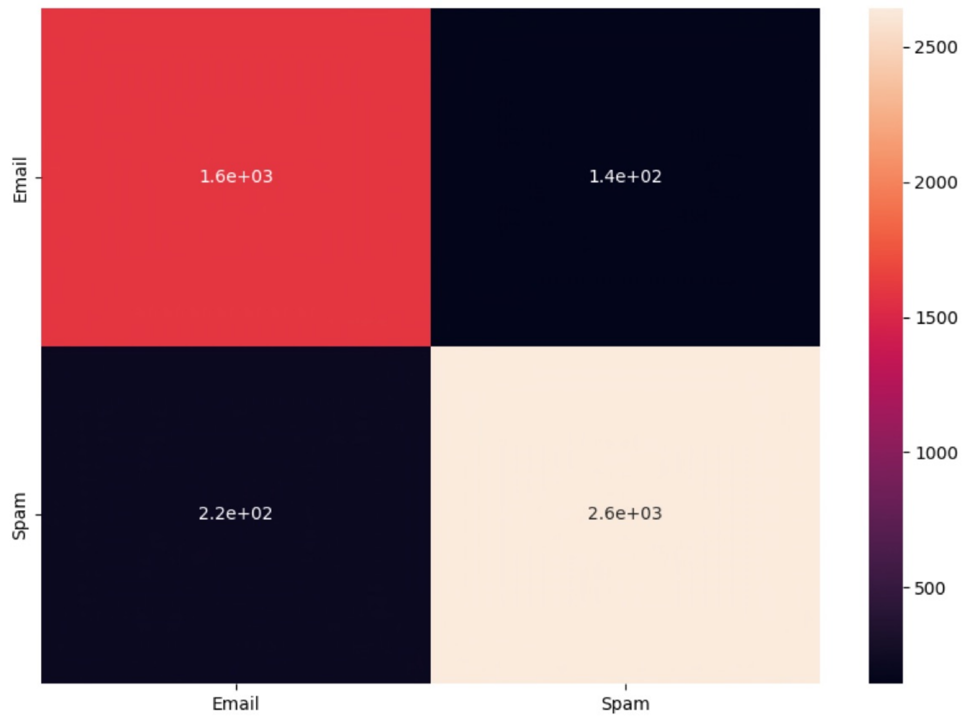
e)

	Predicted $y=-1$	Predicted $y=1$
Actual $y=-1$	1597	144
Actual $y=1$	216	2643

$$\text{Accuracy} = \left[\frac{(1597+2643)}{4600}\right] \times 100$$

$$\text{Accuracy} = \underline{\underline{92.17\%}}$$

The confusion matrix of the same is plotted below-



Problem-3

a) After running the Gaussian Process, Following is table of hyperparameters.

RMSE Table for various values of σ^2 & b.

RMSE		σ^2									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
	5	1.966278	1.933137	1.923422	1.922200	1.924771	1.929215	1.934636	1.940585	1.946822	1.953215
	7	1.920165	1.904878	1.908082	1.915904	1.924806	1.933704	1.942256	1.950382	1.958095	1.965440
b	9	1.897650	1.902521	1.917650	1.932517	1.945702	1.957237	1.967406	1.976494	1.984743	1.992344
	11	1.890509	1.914983	1.938851	1.957938	1.973218	1.985766	1.996377	2.005605	2.013838	2.021347
	13	1.895850	1.935588	1.964600	1.985504	2.001316	2.013881	2.024313	2.033309	2.041320	2.048644
	15	1.909605	1.959551	1.990806	2.011918	2.027372	2.039467	2.049465	2.058107	2.065847	2.072978

b) The best value to use is at b=11 & $\sigma^2 = 0.1$ as highlighted in the table as shown above. The RMSE value here is 1.890509.
In homework 1, the smallest RMSE we could achieve using polynomial regression is 2.08 using 3rd order polynomial.

The drawbacks of this method are that we need to select range of b & σ^2 then get the best one. Each calculation, the program needs to calculate the inverse of the matrix for both mean & variance calculation. Inverse calculation is a very time consuming process relatively.

However in ridge regression, we calculate only 1 parameter λ & it also doesn't have inverse calculation, hence the consumption of time is less.

c) The graph shown below shows both test data points & the predicted value of training data.