Assignment 01 : CS771A
Introduction to Machine Learning

Group name : The_CS771_Group
Jayant Soni (210468), Digvijay (210344), Akhil Naik (210087),
Hridhan Patel (210687), Aman Sarwa (210112), Venkaat Balaje (211159)

1 Calculations

$t_1^u = (1 - c_1) \cdot (t_0^u + P_1) + c_1 \left(t_0^l + s_1\right)$
$t_1^l = (1 - c_1) \cdot \left(t_0^l + q_1\right) + c_1 \cdot (t_0^u + r_1)$
$\Delta_i = t_i^u - t_i^l$
$\Delta_1 = (1 - 2c_i) \cdot \Delta_0 + (q_1 - p_1 + s_1 - r_1) \cdot c_1 + (p_1 - q_1)$
$d_i \stackrel{\text{def}}{=} (1 - 2c_i)$
$\Delta_1 = \Delta_0 \cdot d_1 + \alpha_1 \cdot d_1 + \beta_1$
$\alpha_1 = (p_1 - q_1 + r_1 - s_1)/2$
$\beta_1 = (p_1 - q_1 - r_1 + s_1)/2$

Assume $\Delta_{-1} = 0$
$\Delta_i = d_i \cdot \Delta_{i-1} + \alpha_i \cdot d_i + \beta_i$
$\alpha_i = (p_i - q_i + r_i - s_i)/2, \quad \beta_i = (p_i - q_i - r_i + s_i)/2$
$\Delta_0 = \alpha_0 \cdot d_0 + \beta_0$ (since $\Delta_{-1} = 0$)
$\vdots$
$\Delta_2 = \alpha_0 \cdot d_2 \cdot d_1 \cdot d_0 + (\alpha_1 + \beta_0) \cdot d_2 \cdot d_1 + (\alpha_2 + \beta_1) \cdot d_2 + \beta_2$
$\vdots$
$\Delta_{31} = w_0 x_0 + w_1 \cdot x_1 + \ldots + w_{31} \cdot x_{31} + \beta_{31} = w^\top x + b$

$x_i = d_i \cdot d_{i+1} \cdots d_{31}$
$\omega_0 = \alpha_0$
$w_i = \alpha_i + \beta_{i-1} \quad$ (for $i > 0$)

So we know the expression for Δ_i, now
$\Delta_i = w_0 \cdot x_0 + \ldots. + w_i \cdot x_i + \beta_i$ where $x_k = d_k \cdot d_{k-1} \ldots d_i$
Now I need to find expression for $t^u(c) = \omega^\top \phi(c) + b$

$$t_i^u = (1 - c_i) \cdot \left(t_{i-1}^u + p_i\right) + c_i \cdot (t_{i-1}^l + s_i)$$
$$t_i^l = (1 - c_i) \cdot (t_{i-1}^l + q_i) + c_i \cdot (t_{i-1}^u + r_i)$$

Now we know the expression for Δ_i derived earlier

$$\Delta_i = t_i^u - t_i^l$$

Now we will replace $t_i^l = t_i^u - \Delta_i$

$$t_i^u = (1 - c_i) \cdot \left(t_{i-1}^u + p_i\right) + c_i \cdot \left(t_{i-1}^u - \Delta_{i-1} + s_0\right)$$

Assume $t^u_{-1} = 0$

$$\begin{aligned}
t^u_0 &= (1-c_0)(p_0-s_0)+s_0\\
t^u_1 &= (1-c_0)(p_0-s_0)+(1-c_1)(p_1-c_1)-c_1\Delta_0+\beta_1\\
t^u_2 &= (1-c_0)(p_0-s_0)+(1-c_1)(p_1-s_1)+(1-c_2)(p_2-s_2)-c_1\Delta_0-c_2\Delta_1+\beta_2\\
&\vdots\\
t^u_i &= \sum_{k=0}^{k=i}(1-c_k)(p_k-s_k)-\sum_{k=1}^{k=i}c_k\Delta_{k-1}+\beta_i\\
t^u_{31} &= \sum_{k=0}^{31}(1-c_k)(P_k-s_k)-\sum_{k=1}^{31}c_k\Delta_{k-1}+\beta_{31}
\end{aligned}$$

We see that

$$\Delta_i \Rightarrow d_i = (1-2c_i)$$
$$\text{as } c_i = \{0,1\} \Rightarrow (1-2(i)) \Rightarrow \{-1,1\}$$

So we will try to convert t^u_i expression such that it will also have $1-2c_i$ terms

$$\begin{aligned}
t^u_{31} &= \sum_{k=0}^{31}\frac{(1-2c_k+1)}{2}(p_k-s_k)+\sum_{k=1}^{31}\frac{(1-2c_k-1)}{2}\Delta_{k-1}\\
&= \sum_{k=0}^{31}(1-2c_k)\frac{(p_k-s_k)}{2}+\sum_{k=0}^{31}\frac{(p_k-s_k)}{2}+\sum_{k=1}^{31}(1-2c_k)\frac{\Delta_{k-1}}{2}\\
&-\sum_{k=1}^{31}\frac{\Delta_{k-1}}{2} \quad (1)
\end{aligned}$$

We will try to simplify the terms $\sum_{k=1}^{31}(1-2c_k)\frac{\Delta_{k-1}}{2}$ &

$$-\sum_{k=1}^{31}\frac{\Delta_{k-1}}{2}$$

$$d_k = (1-2c_k)$$

$$-\sum_{k=1}^{31}\frac{\Delta_{k-1}}{2} = -\frac{1}{2}(\Delta_0+\Delta_1+\ldots+\Delta_{30}) \quad (2)$$

$\Delta_0 = \omega_0 d_0 + \beta_0$
$\Delta_1 = \omega_0 d_0 d_1 + \omega_1 d_1 + \beta_1$
$\Delta_2 = \omega_0 d_0 d_1 d_2 + \omega_1 d_1 d_2 + \omega_2 d_2 + \beta_2$
$\Delta_3 = \omega_0 d_0 d_1 d_2 d_3 + \omega_1 d_1 d_2 d_3 + \omega_2 d_2 d_3 + \omega_3 d_3 + \beta_3$
$\vdots$
$\Delta_{30} = w_0 d_0 \ldots d_{30} + w_1 d_1 \ldots d_{30} + \cdots + w_{30} d_{30} + \beta_{30}$

Now,

$\frac{1}{2}\sum_{k=1}^{31}(1-2c_k)\Delta_{k-1} = \frac{1}{2}\sum_{k=1}^{31}d_k\Delta_{k-1} \quad (3)$

$$\begin{aligned}
d_1\Delta_0 &= \omega_0 d_0 d_1 + \beta_0 d_1\\
d_2\Delta_1 &= \omega_0 d_0 d_1 d_2 + \omega_1 d_1 d_2 + \beta_1 d_2\\
d_3\Delta_2 &= \omega_0 d_0 d_1 d_2 d_3 + \omega_1 d_1 d_2 d_3 + \omega_2 d_2 d_3 + \beta_2 d_3\\
&\vdots\\
d_{31}\Delta_{30} &= w_0 d_0 \ldots d_{31} + w_1 d_1 \ldots d_{31} + \ldots + w_{30} d_{30} d_{31} + \beta_{30} d_{31}
\end{aligned}$$

Now terms will cancel out and we will have

$$t_{31}^{u}=\frac{1}{2}\sum_{k=0}^{31}\left(1-2c_k\right)u_k+\frac{1}{2}\left(w_0d_0\ldots d_{31}+w_1d_1\ldots d_{31}+\ldots+\omega_{30}d_{30}d_{31}\right)+b$$

so there will be around $32+31=63$ terms and one bias term i.e. b

$$t_{31}^{u}=w^{\top}\phi(c)+b$$

$$\phi(c)=\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{31} \\ d_0,d_1,\ldots..d_{31} \\ d_1,\ldots..d_{31} \\ d_2\ldots..d_{31} \\ \vdots \\ d_{30}d_{31} \end{bmatrix}$$

dimensionality will be 63.

3. In this part we need to convert this into a linear model for prediction

model for prediction

$$t_{31}^{4}=\sum_{k=0}^{31}\left(1-2C_k\right)\frac{\left(P_k-S_k\right)}{2}+\frac{1}{2}\sum_{k=1}^{31}\left(1-2C_k\right)\Delta_{k-1}-\frac{1}{2}\sum_{k=1}^{31}\Delta_{k-1}+b$$

For d_0 we will have two terms from (1), (2) and (3)

$$d_0*\left(\frac{\left(p_0-s_0\right)}{2}\right)\&\frac{\omega_0d_0}{2}$$

$$\frac{d_0}{2}\left(p_0-s_0-\omega_0\right)$$

$$\omega_0=\alpha_0=\left(p_0-q_0+r_0-s_0\right)/2$$

$$p_0-s_0-\omega_0=\left(p_0+q_0r_0-s_0\right)/2$$

Weigh corresponding to $d_0=\left(p_0+q_0-r_0-s_0\right)/4$ For d_1 to d_{30} we will have three terms from (1), (2) and (3)

$$\begin{aligned}\frac{di}{2}&=\left(p_i-s_i+\beta_{i-1}-w_i\right)\\ \omega_i&=\alpha_i+\beta_{i-1}\\ p_i-s_i+\beta_{i-1}-\alpha_i-\beta_{i-1}&=p_i-s_i-\alpha_i\\ &=p_i-s_i-\left(p_i-q_i+r_i-si\right)/2\\ &=p_i+q_i-r_i-s_i\end{aligned}$$

For d_i to d_{30} weights will be $\frac{(pi+qi-ri-6i)}{4}$ For d_{31} we will have form two terms from 1 & 3

$$\frac{d_{31}}{2}=\left(p_{31}-s_{31}+\beta_{30}\right)$$

$$\beta_i=p_i-q_i-r_i+s_i$$

$$p_{31}-s_{31}+p_{30}-q_{30}-r_{30}+s_{30}$$

This will be the weight for d_{31} and weights for the remaining terms those are d_0. $d_1 ... d_{31}$, $d_1 ... d_{31}$, $d_2 ... d_{31}$,..$d_{30} d_{31}$ are $\frac{W_0}{2}$ to $\frac{W_{30}}{2}$ respectively where $W_0 = \alpha_0$ and for i = 1 to 30 $W_i = \alpha_i + \beta_{i-1}$

$$\omega = \frac{1}{4} \begin{bmatrix} p_0 + q_0 - r_0 - s_0 \\ p_1 + q_1 - r_1 - s_1 \\ \vdots \\ p_{30} + q_{30} - r_{30} - s_{30} \\ p_{31} - s_{31} + p_{30} - q_{30} - \gamma_{30} + s_{30} \\ \alpha_0 \\ \alpha_1 + \beta_0 \\ \vdots \\ \alpha_{30} + \beta_{29} \end{bmatrix}$$

So our linear model will be $\omega^\top \phi(c) + b$
Now we are looking at top signals of both pufs and checking their difference for prediction if signal from puf 1 reaches 1^{st} output will be 1 and if it reaches from puf 0 first output will be 0.
So we will look $\left(tu^{31}\right)_1 - \left(tu^{31}\right)_0$ so our linear model will become where we will have two weight matrices ω_0 and ω_1 corresponding to these pufs. So our linear model becomes (ω_1 -$\omega_0)^\top \phi(c)$+ b and prediction $\tilde{\omega}^\top$ will become $\frac{1+\text{sign}\left(\tilde{\omega}^\top \tilde{\phi}(c)+\tilde{b}\right)}{2} = r'(c)$ similar approach we can use for response o which will compare lower times of both the pufs if will have slightly different weight matrix. but the dimensionality will be the same for both i.e 63.

Experimentation Outcomes

LinearSVC

C	Tolerance	Training Time (seconds)	Accuracy (Response 0)	Accuracy (Response 1)	Max Iter
10^{-2}	0.0001	7.40	0.9804	0.9948	5000
10^{-1}	0.0001	7.21	0.9840	0.9966	5000
10^{0}	0.0001	10.68	0.9841	0.9979	5000
10^{1}	0.0001	16.75	0.9842	0.9989	5000
10^{2}	0.0001	49.68	0.9839	0.9989	5000
10^{3}	0.0001	70.12	0.9505	0.9986	5000

Table 1: For constant Tolerance = 0.0001, Maximum Iteration = 5000

Tolerance	C	Training Time (seconds)	Accuracy (Response 0)	Accuracy (Response 1)	Max Iter
10^{-6}	10	14.67	0.9846	0.9989	5000
10^{-5}	10	15.01	0.9845	0.9989	5000
10^{-4}	10	16.59	0.9843	0.9989	5000
10^{-3}	10	15.60	0.9845	0.9988	5000
10^{-2}	10	15.27	0.9847	0.9988	5000
10^{-1}	10	14.90	0.9850	0.9985	5000
10^{0}	10	16.16	0.9764	0.9939	5000
10^{1}	10	6.58	0.9504	0.9504	5000

Table 2: For constant C = 10, Maximum Iteration = 5000

Number of Iterations	Tolerance	Training Time (seconds)	Accuracy (Response 0)	Accuracy (Response 1)	Max Iter
0	0.0001	20.21	0.9590	0.9919	10
2000	0.0001	11.91	0.9816	0.9981	2000
4000	0.0001	13.31	0.9841	0.9988	4000
6000	0.0001	14.00	0.9844	0.9989	6000
10000	0.0001	18.85	0.9843	0.9989	10000

Table 3: For constant Tolerance = 0.0001

Logistic Regression

C	Tolerance	Training Time (seconds)	Accuracy (Response 0)	Accuracy (Response 1)	Max Iter
10^{-2}	0.0001	9.15	0.9789	0.9907	5000
10^{-1}	0.0001	6.33	0.9800	0.9932	5000
10^{0}	0.0001	8.10	0.9805	0.9956	5000
10^{1}	0.0001	8.42	0.9806	0.9970	5000
10^{2}	0.0001	9.18	0.9808	0.9981	5000
10^{3}	0.0001	9.14	0.9808	0.9990	5000
10^{4}	0.0001	8.61	0.9808	0.9994	5000
10^{5}	0.0001	10.29	0.9808	0.9995	5000

Table 4: For constant Tolerance = 0.0001, Maximum Iteration = 5000

Tolerance	C	Training Time (seconds)	Accuracy (Response 0)	Accuracy (Response 1)	Max Iter
10^{-6}	10	8.32	0.9970	0.9989	5000
10^{-5}	10	7.10	0.9970	0.9989	5000
10^{-4}	10	9.00	0.9969	0.9989	5000
10^{-3}	10	8.63	0.9970	0.9989	5000
10^{-2}	10	7.44	0.9973	0.9989	5000
10^{-1}	10	7.27	0.9971	0.9988	5000
10^{0}	10	8.15	0.9937	0.9957	5000
10^{1}	10	8.15	0.9800	0.9993	5000

Table 5: For constant C = 10, Maximum Iteration = 5000

Number of Iterations	Tolerance	Training Time (seconds)	Accuracy (Response 0)	Accuracy (Response 1)	Max Iter
0	0.0001	9.35	0.9808	0.9980	10
2000	0.0001	8.91	0.9808	0.9981	2000
4000	0.0001	9.13	0.9808	0.9981	4000
6000	0.0001	9.86	0.9808	0.9981	6000
10000	0.0001	9.72	0.9808	0.9981	10000

Table 6: For constant Tolerance = 0.0001

Appendix

Fig 1: LinearSVC: Training Time vs C

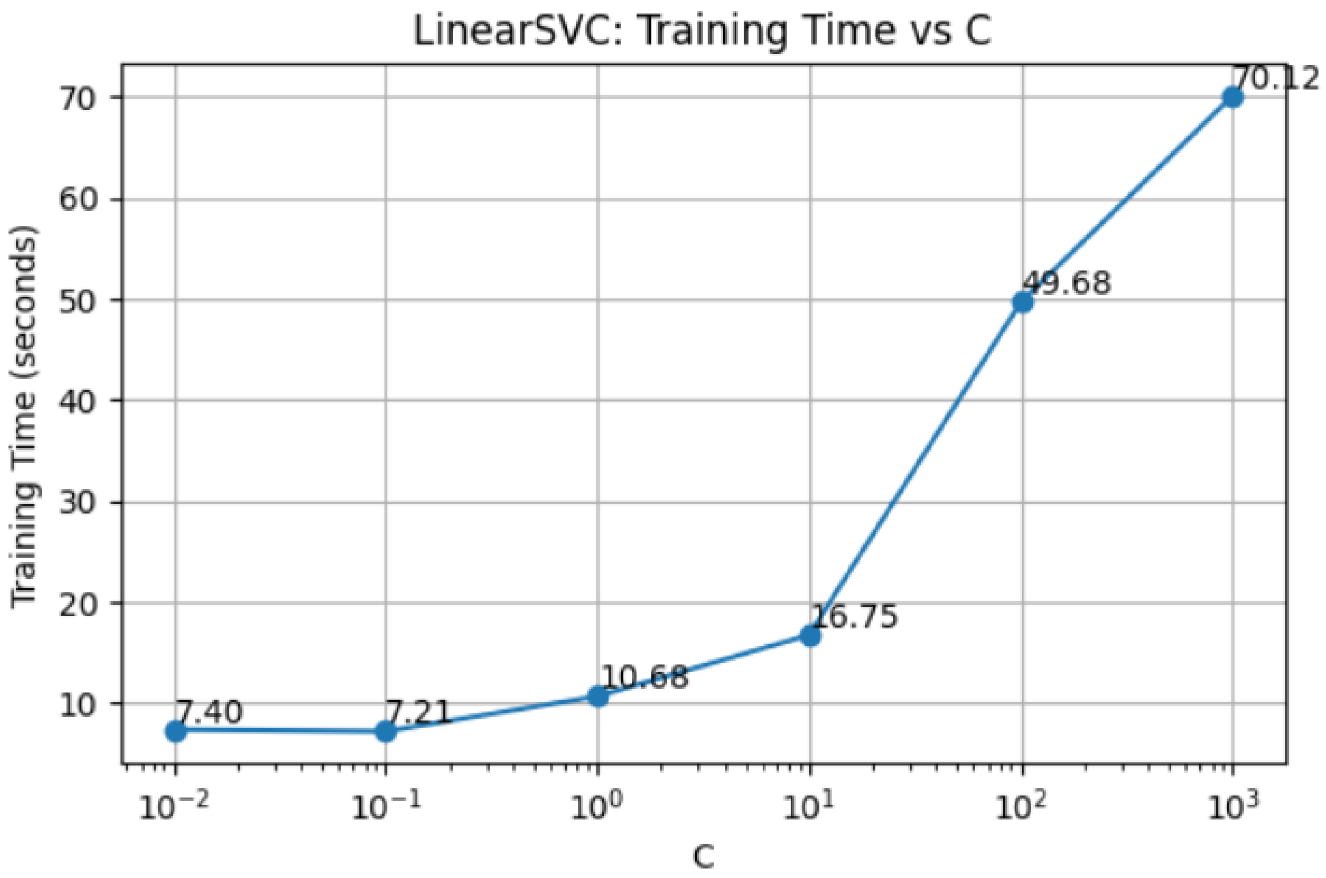

Fig 2: LinearSVC: Accuracy vs C

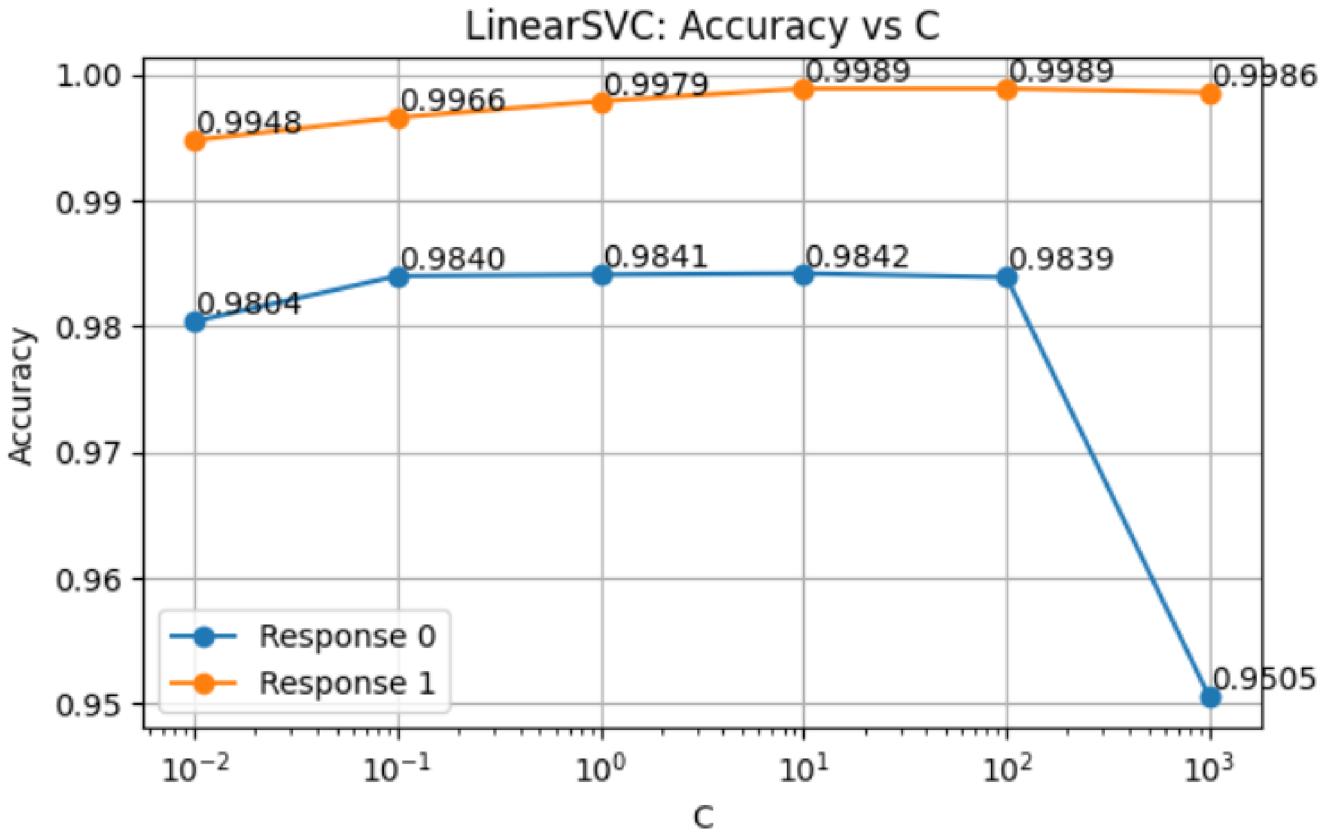

Fig 3: LinearSVC: Training Time vs Tolerance

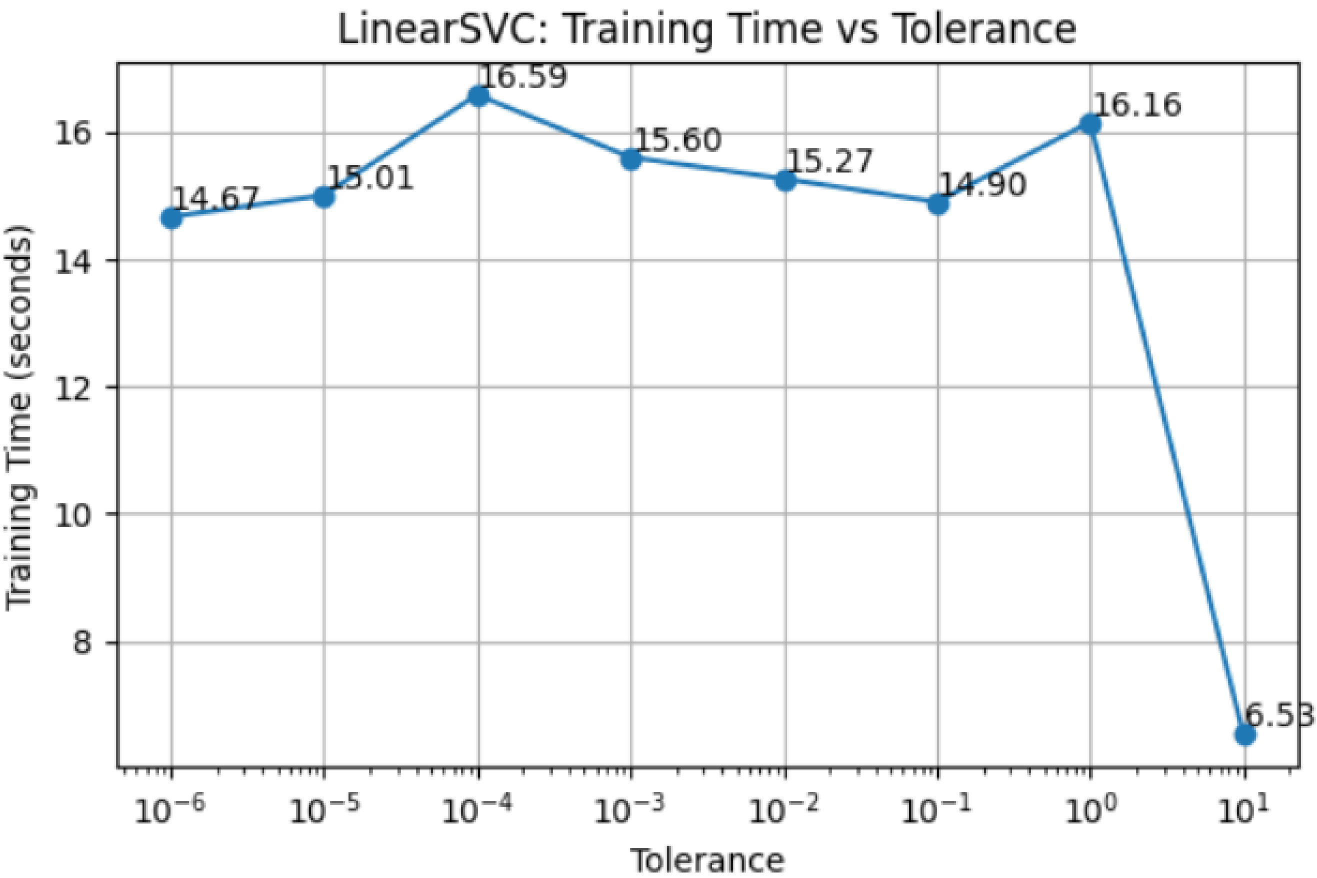

Fig 4: LinearSVC: Accuracy vs Tolerance

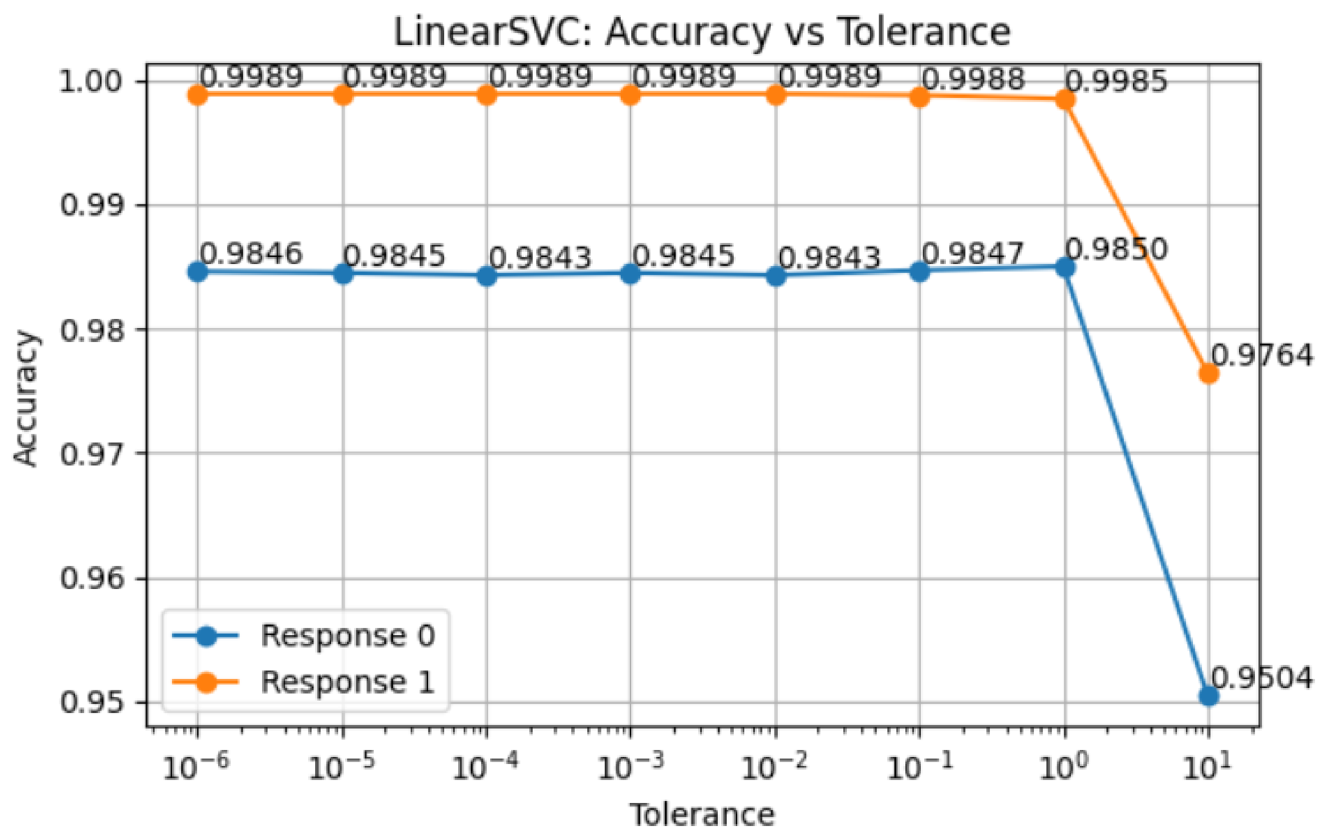

Fig 5: Logistic Regression: Training Time vs C

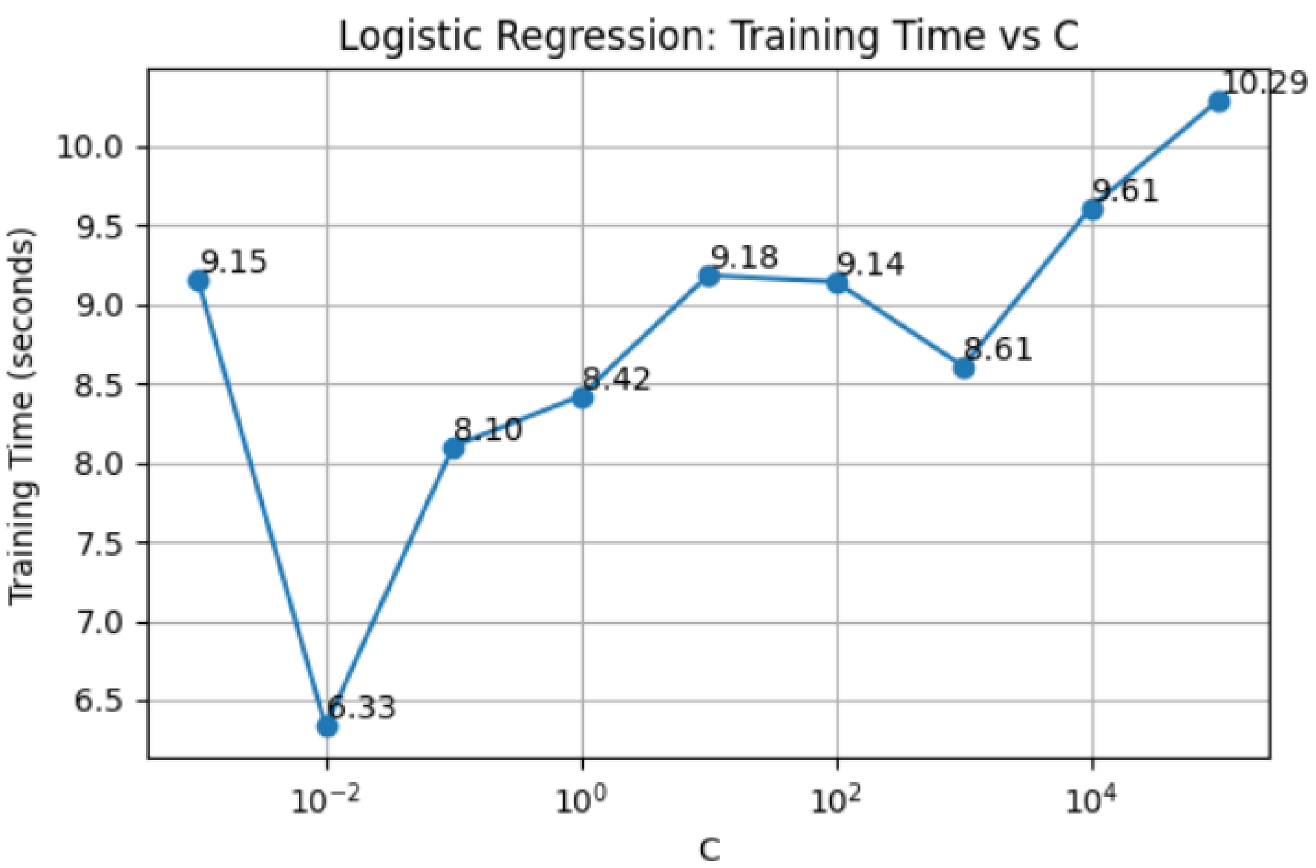

Fig 6: Logistic Regression: Accuracy vs C

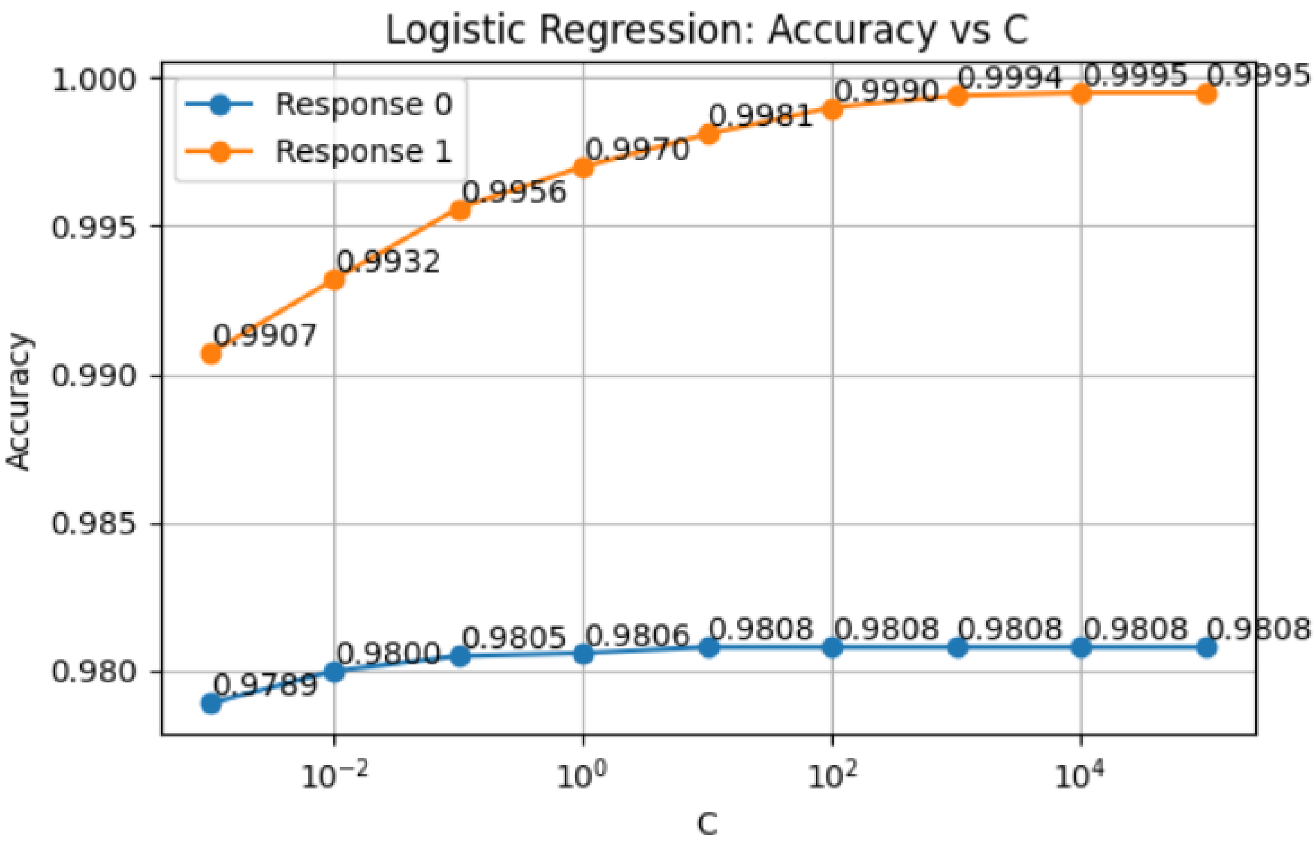

Fig 7: Logistic Regression: Training Time vs Tolerance

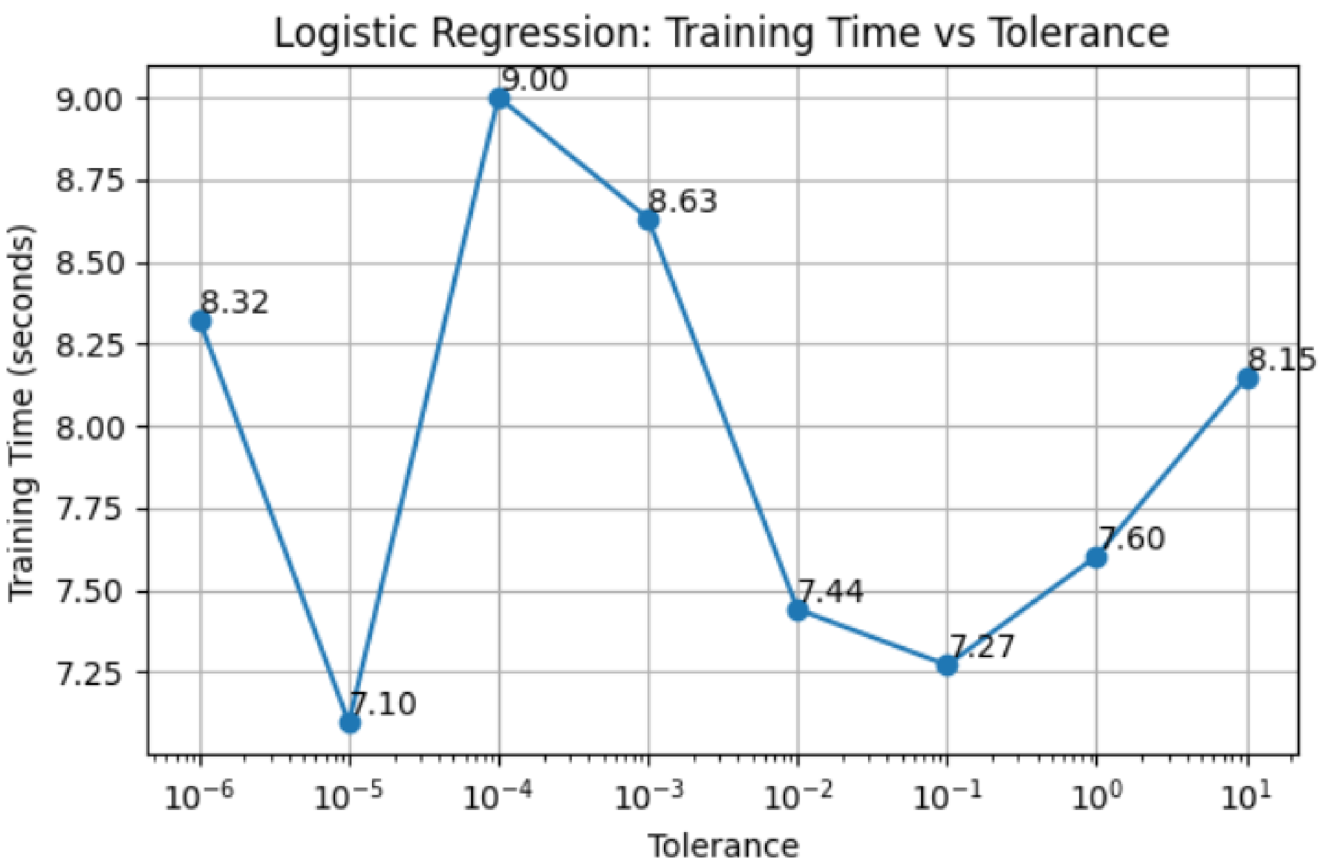

Fig 8: Logistic Regression: Accuracy vs Tolerance

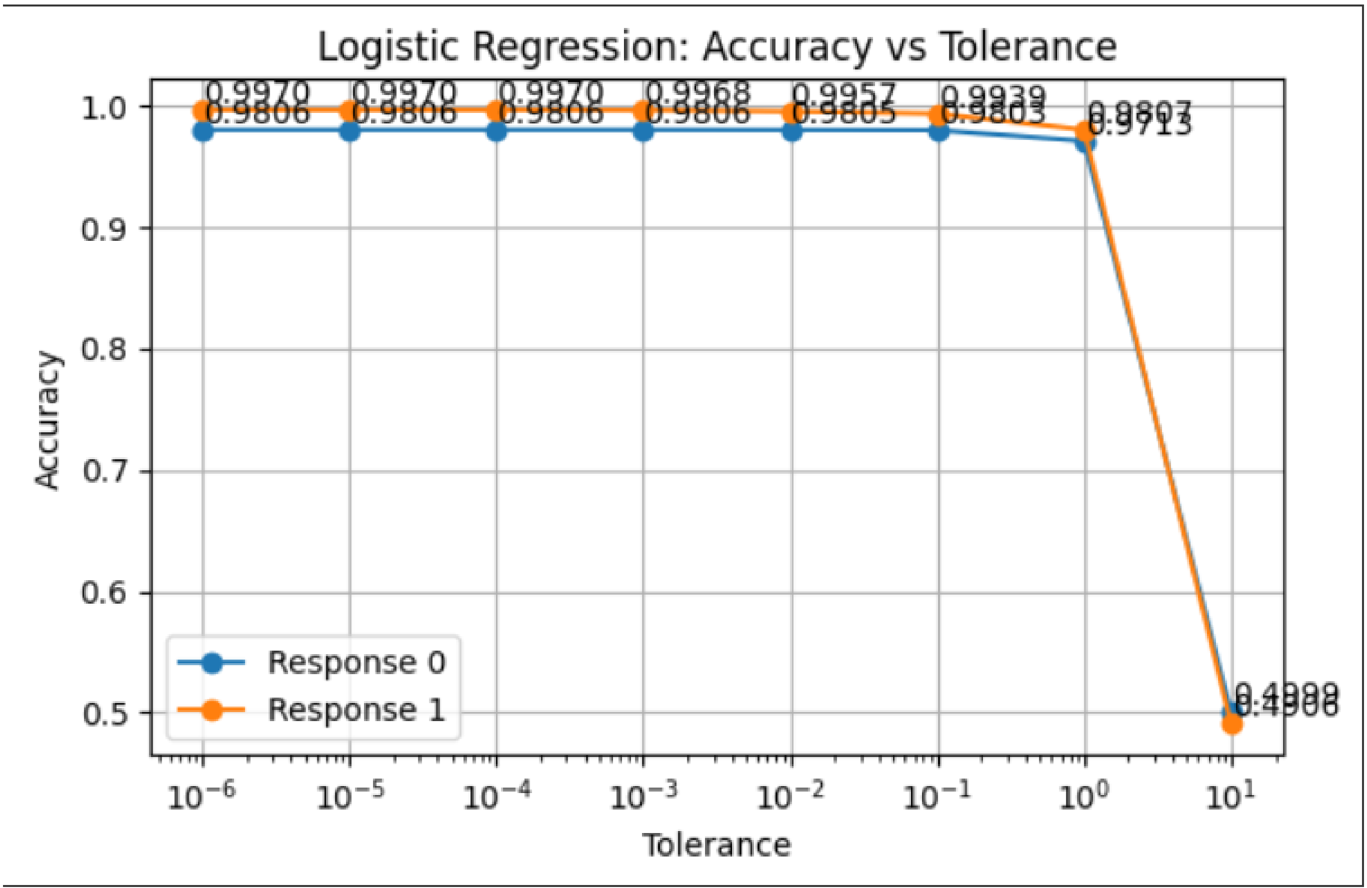

Fig 9: LinearSVC: Training Time vs Number of Iterations

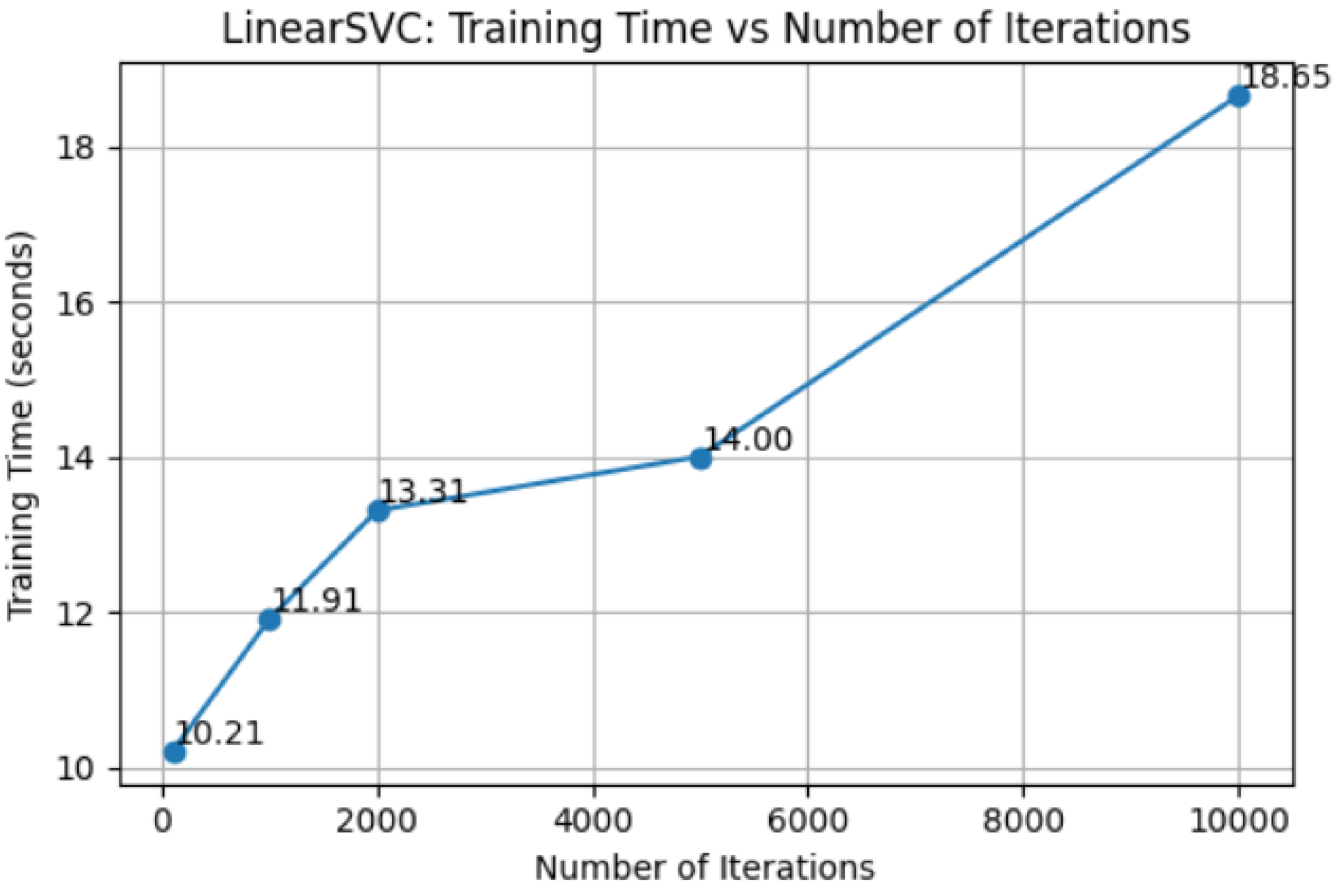

Fig 10: LinearSVC: Accuracy vs Number of Iterations

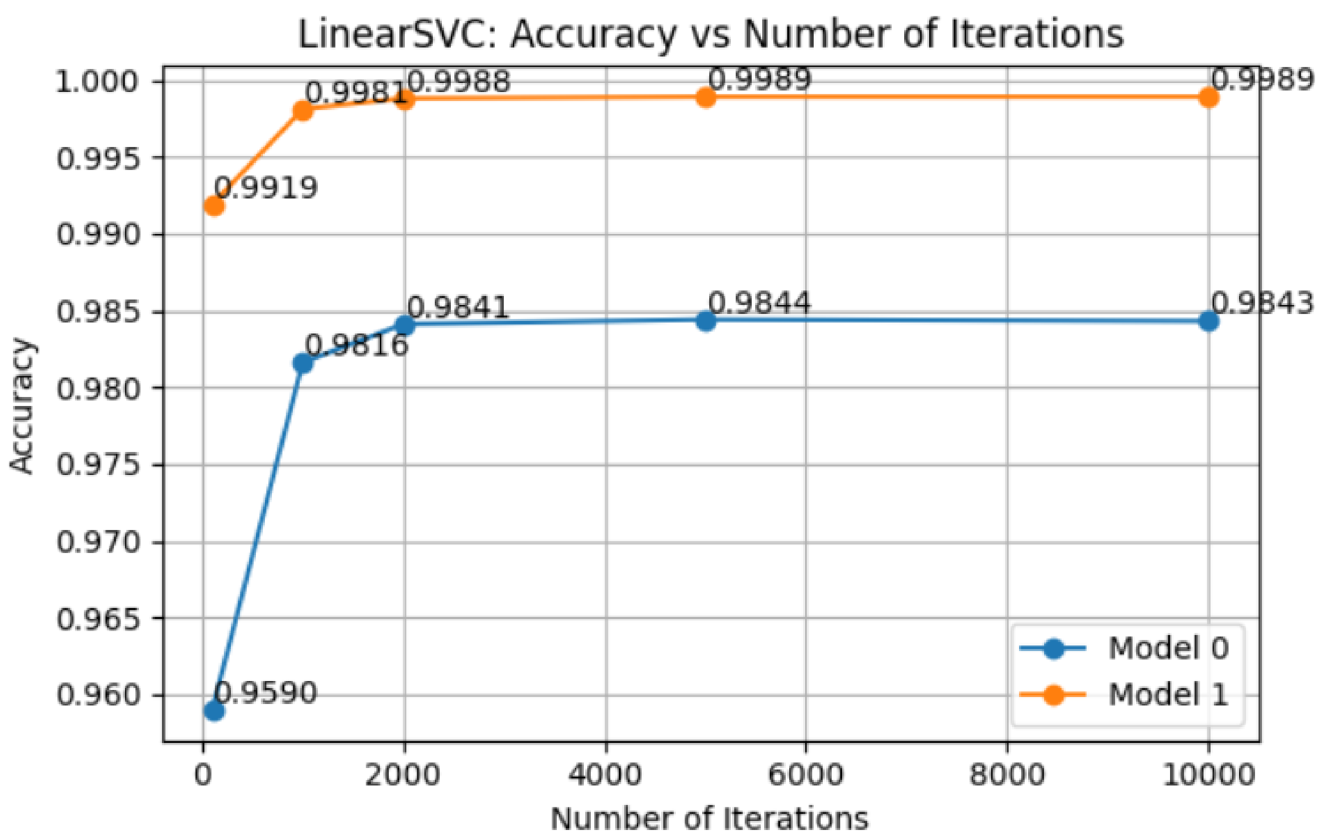

Fig 11: Logistic Regression: Training Time vs Number of Iterations

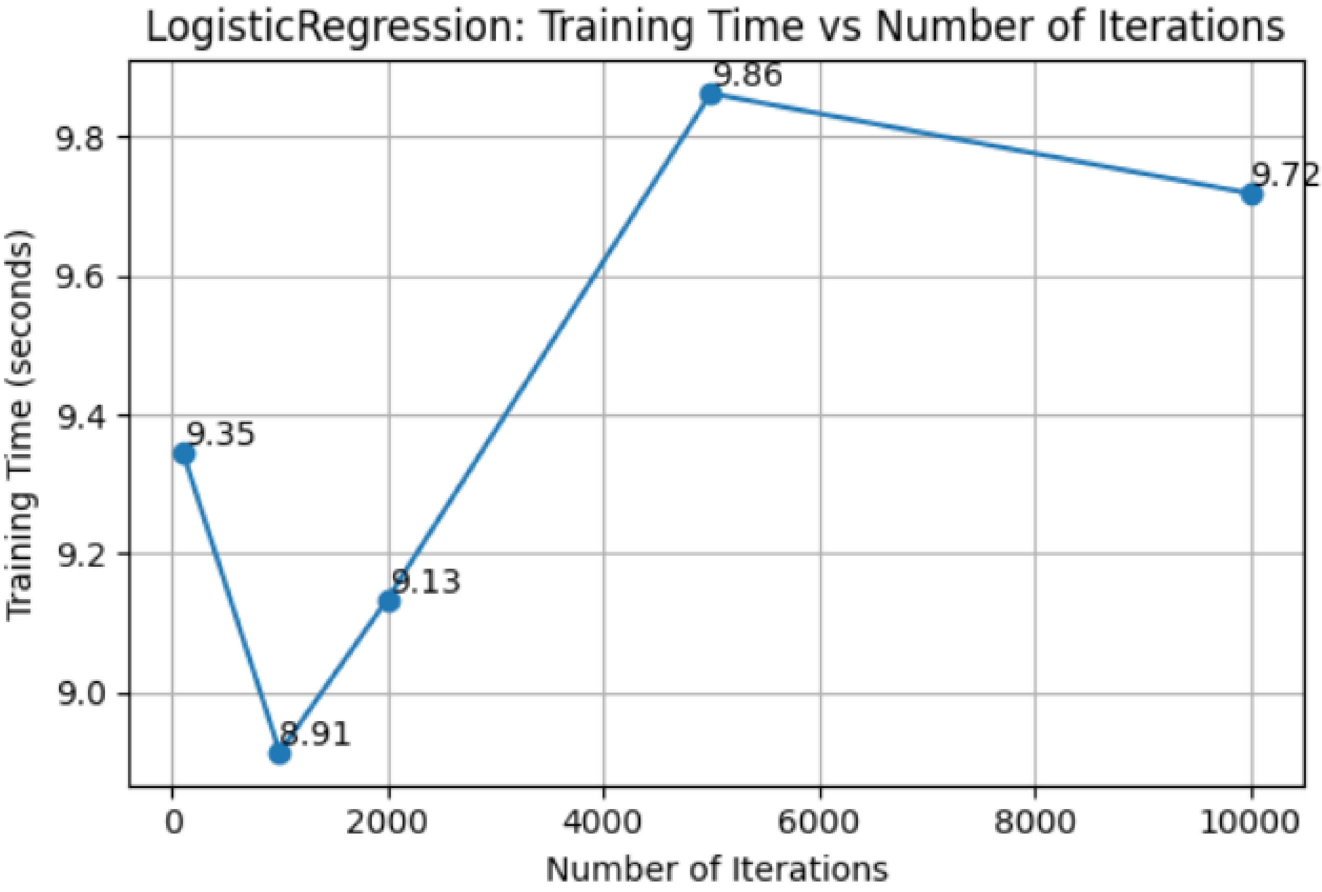

Fig 12: Logistic Regression: Accuracy vs Number of Iterations