Part C – Logistic Regression

Learning Task 1												
		Batch Gradient Descent				Mini-batch Gradient Descent				Stochastic Gradient Descent		
		Learning Rates				Learning Rates				Learning Rates		
		0.01	**0.001**	**0.0001**		**0.01**	**0.001**	**0.0001**		**0.01**	**0.001**	**0.0001**
Decision Probability Threshold	**0.3**	0.9	0.9	0.89		0.72	0.7	0.67		0.82	0.8	0.89
	0.4	0.9	0.9	0.89		0.72	0.7	0.67		0.82	0.8	0.89
	0.5	0.9	0.9	0.91		0.72	0.7	0.67		0.82	0.8	0.89
	0.6	0.9	0.9	0.91		0.72	0.7	0.67		0.82	0.8	0.89
	0.7	0.9	0.9	0.91		0.72	0.7	0.67		0.82	0.8	0.89

Learning Task 2												
		Batch Gradient Descent				Mini-batch Gradient Descent				Stochastic Gradient Descent		
		Learning Rates				Learning Rates				Learning Rates		
		0.01	**0.001**	**0.0001**		**0.01**	**0.001**	**0.0001**		**0.01**	**0.001**	**0.0001**
Decision Probability Threshold	**0.3**	0.94	0.93	0.64		0.93	0.91	0.67		0.97	0.97	0.98
	0.4	0.96	0.97	0.8		0.93	0.91	0.84		0.97	0.97	0.98
	0.5	0.96	0.97	0.9		0.93	0.92	0.92		0.97	0.97	0.98
	0.6	0.97	0.92	0.56		0.93	0.93	0.66		0.97	0.97	0.97
	0.7	0.96	0.85	0.41		0.93	0.93	0.4		0.97	0.97	0.97

What happens to testing accuracy when you vary the decision probability threshold from 0.5 to 0.3, 0.4, 0.6 and 0.7?

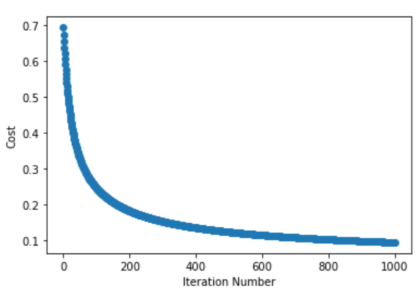
The accuracy is not affected by the threshold as this is the case of overfitting where the probability of positive points is very close to one and probability of negative points is very close to zero.

Learning rate:

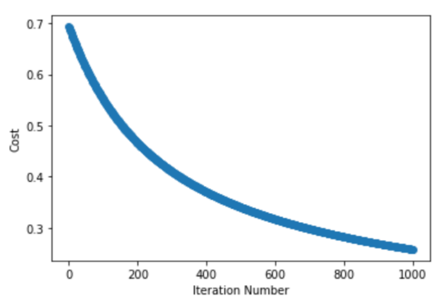
When the learning rate is high the weight vector converges to the global minima faster hence we observed a higher accuracy at a higher learning rate.

But this need not be the case all the time, if the learning rate is too high the weight vector moves ahead of the global minimum and may result in oscillation about the minimum, giving a low accuracy.

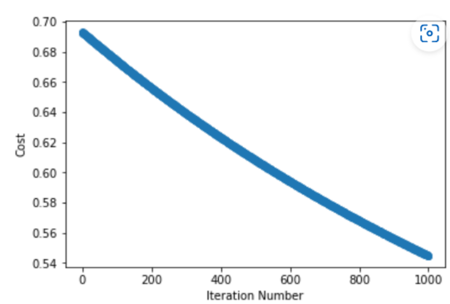
When the learning rate is reduced the cost function vs number of iteration curve will tend to be linear



Learning rate = 0.01

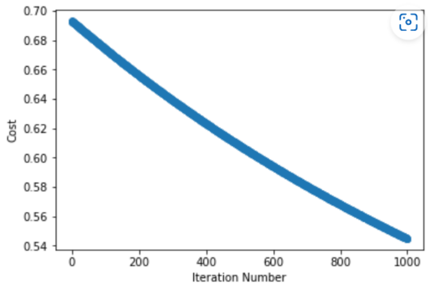


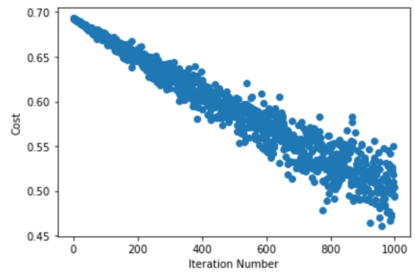
Learning rate = 0.001

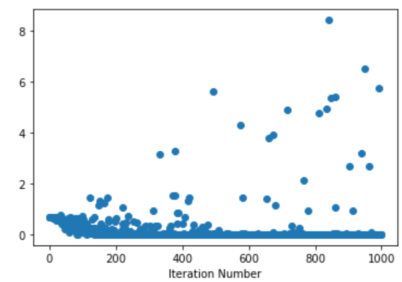


Learning rate = 0.0001

Batch Gradient Descent vs Mini-batch Gradient Descent vs Stochastic Gradient Descent







Learning Task 1

		Batch Gradient Descent				Mini-batch Gradient Descent				Stochastic Gradient Descent		
		0.01	0.001	0.0001		0.01	0.001	0.0001		0.01	0.001	0.0001
Average performance metrics of 10 random splits	Mean Accuracy	90.957%	91.489%	89.893%		75.372%	77.851%	71.148%		70.106%	72.659%	71.755%
	Mean Recall	98.275%	97.175%	93.888%		77.521%	78.686%	71.281%		72.482%	82.399%	78.268%
	Mean Precision	92.432%	93.989%	95.480%		96.463%	98.748%	99.724%		97.253%	89.937%	93.198%
	Mean F1 Score	95.264%	95.550%	94.677%		85.929%	87.522%	83.068%		80.651%	82.393%	82.576%

Learning Task 2

		Batch Gradient Descent				Mini-batch Gradient Descent				Stochastic Gradient Descent		
		0.01	0.001	0.0001		0.01	0.001	0.0001		0.01	0.001	0.0001
Average performance metrics of 10 random splits	Mean Accuracy	95.851%	92.553%	66.489%		90.234%	91.234%	90.224%		95.946%	96.936%	94.946%
	Mean Recall	97.849%	98.913%	94.475%		93.474%	95.474%	92.474%		96.019%	98.009%	95.219%
	Mean Precision	98.913%	97.849%	96.067%		91.997%	94.997%	91.887%		96.903%	98.913%	96.823%
	Mean F1 Score	98.378%	98.378%	95.264%		92.235%	95.235%	92.212%		97.429%	98.459%	97.119%

The cost curve for batch gradient descent is typically smoother than that of mini-batch or stochastic gradient descent in logistic regression because of the way that each method updates the model parameters.

Batch gradient descent updates the model parameters by calculating the gradient of the cost function with respect to all training examples in the batch, and then taking a step in the direction of that gradient. Since the gradient is calculated over all training examples in the batch, it provides a more accurate estimate of the direction of steepest descent, leading to smoother updates and a smoother cost curve.

On the other hand, mini-batch and stochastic gradient descent update the model parameters using only a subset of the training examples. Mini-batch gradient descent updates the parameters using a small batch of examples, while stochastic gradient descent updates the parameters using a single example at a time. This can result in more abrupt updates to the model parameters and a more erratic cost curve because the gradient is estimated based on a smaller sample of the data.

Additionally, mini-batch and stochastic gradient descent are more prone to noise in the gradient estimate due to the smaller sample sizes, which can lead to fluctuations in the cost curve. However, these methods can often converge faster than batch gradient descent, as they update the model parameters more frequently, allowing for quicker adjustments to the gradient descent path