1. Machine learning fundamentals
    a. Using all 100 data points for just training the model is a bad idea because it will cause overlap between other data sets such as testing data points and validation data points. When the model is trained, it will be tailored very specifically for those 100 data points and will ultimately pass any testing and validation conducted using those 100 data points.
    b. The number of iterations that gradient descent is run during training is an example of hyperparameter tuning. Hyperparameter tuning involves configuring the training process to optimize results. Using the test set to determine hyperparameters is bad practice because it will again cause the model to be trained specifically to the data it is to be ultimately tested with. An appropriate way to partition the data points would be to randomly sample the data points without replacement to generate a validation set.
    c. I agree with this fear because the database used to generate the training, validation, and testing sets would most likely be coming from the same distribution. This may lead to overfitting because the model would become too specific towards the probabilities and noise of that distribution. An example leading to bad real world effects is with cancer detection. A model that is trained, validated, and tested based on a database of cancer patients in a country with a single large ethnic majority will perform poorly when utilized for a group of people of a different ethnicity or geographic location. Hypothetically, speaking, drawing data samples from areas with diverse populations can help overcome this instance of overfitting.

2. Cross Validation

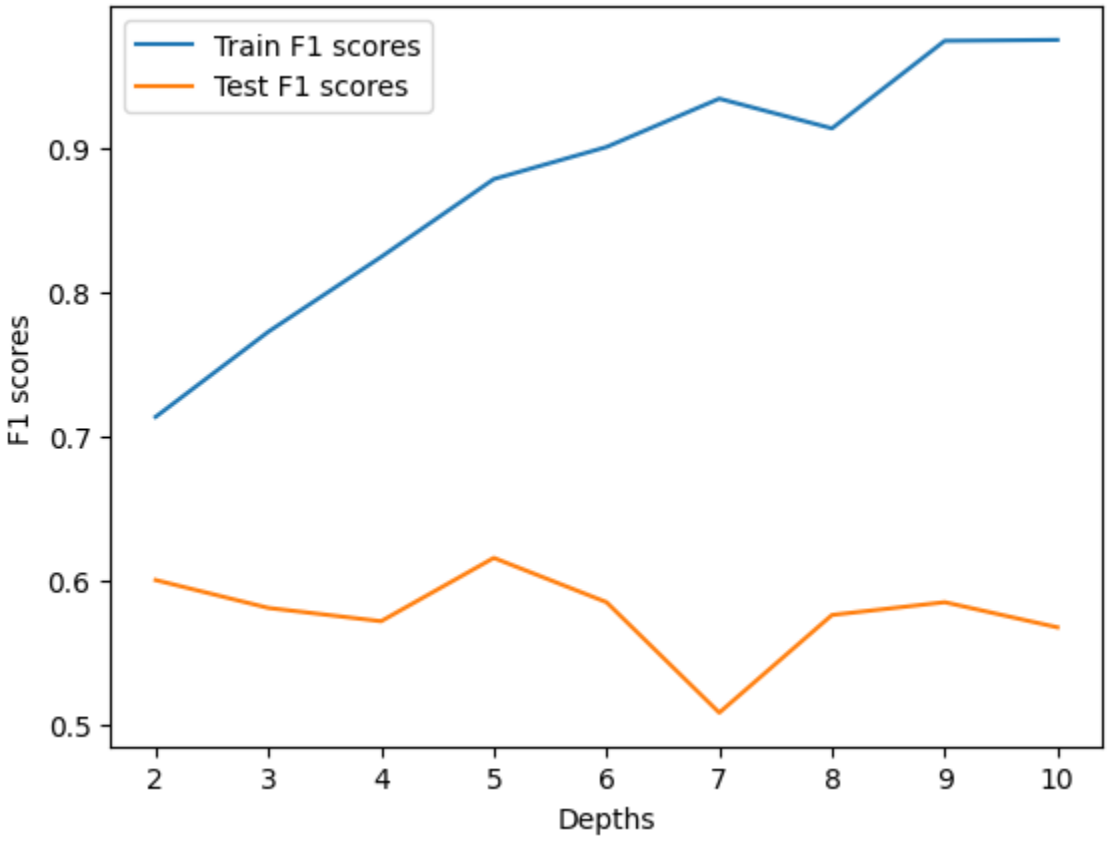
    a. Based on the class balance, this task may be different from previous binary classification tasks we have done because the labels are heavily skewed as negatives. There is an inherent lack of balance in the labels.

```
print( get_misclass(ytrain, clf.predict(Xtrain) ), get_f1(ytrain, clf.predict(Xtrain) )) #training
print (get_misclass(ytest, clf.predict(Xtest) ), get_f1(ytest, clf.predict(Xtest) ))     #testing
```

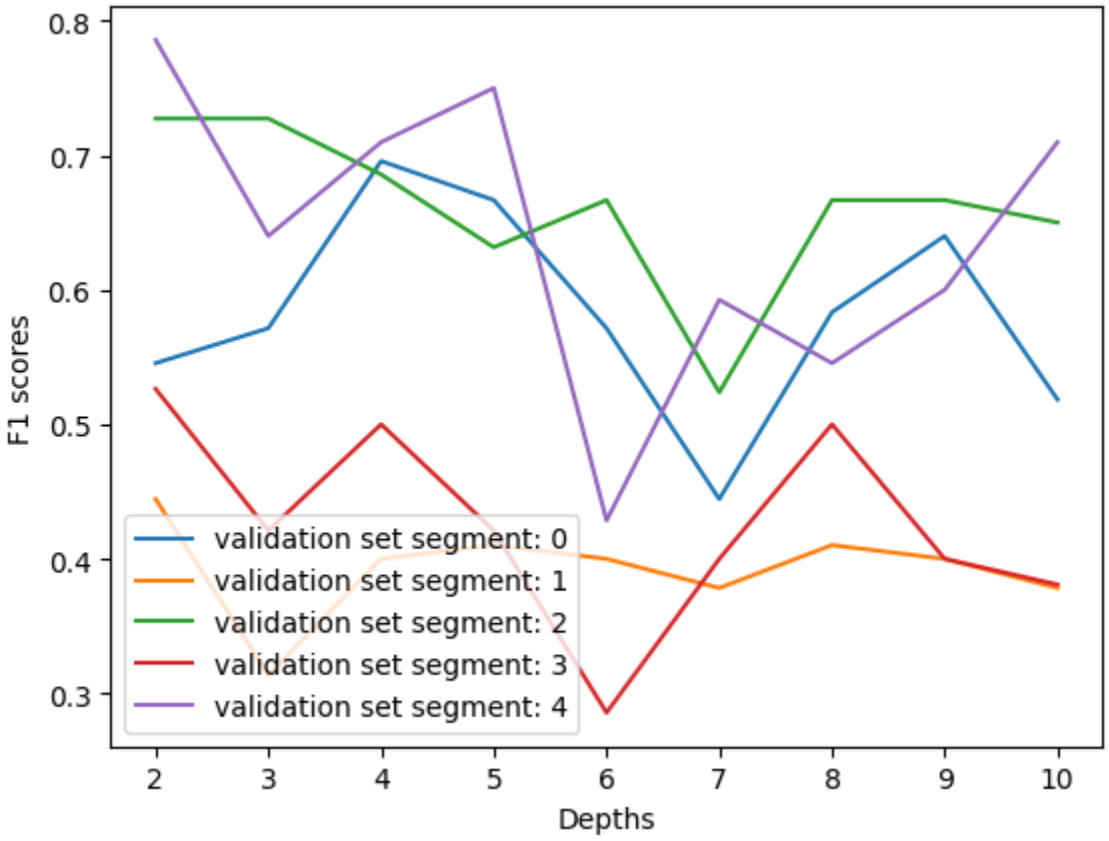
```
0.04125 0.7724137931034484
0.07142857142857142 0.6031746031746031
```

    b. Using misclassification rate over the test and train set, the performance was poorer by only 3%. Based on this alone, the classifier does a good job.

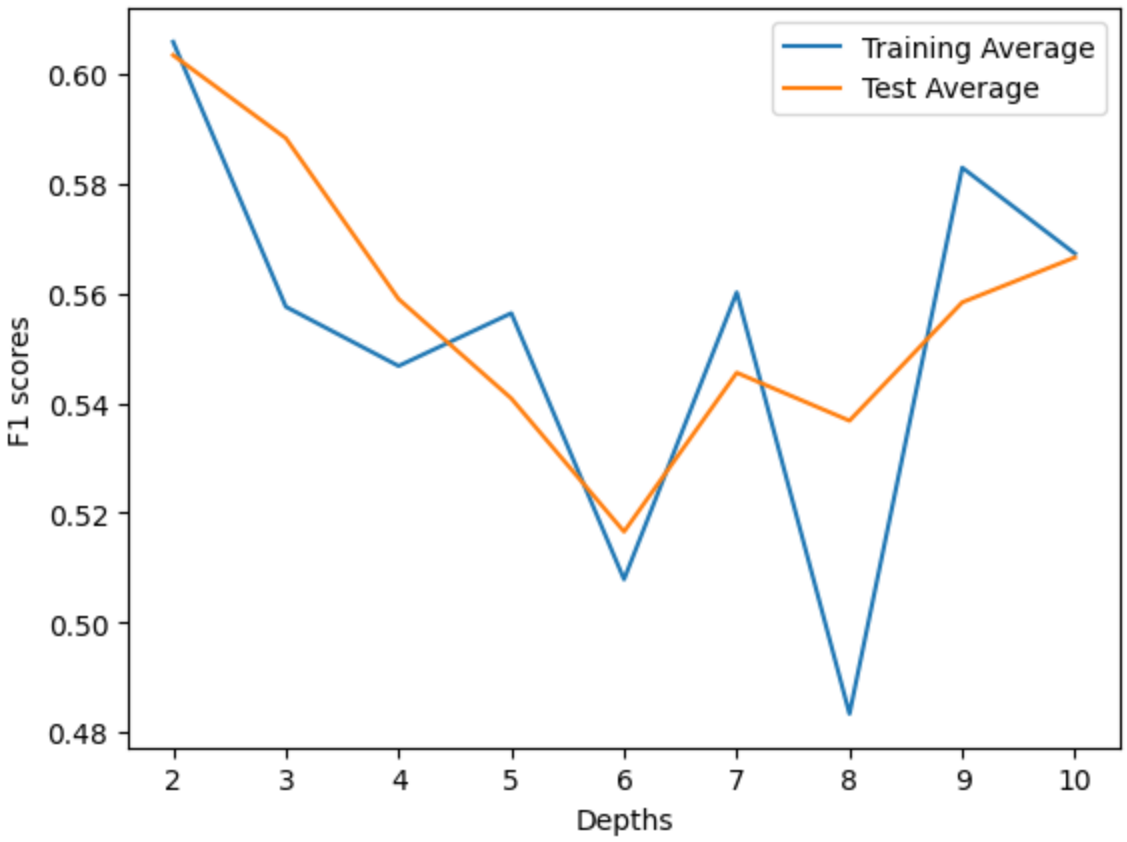
    c. The F1 score of training data was 77.7% and the F1 score of the test data was 60%. The F1 score does look like a more reasonable performance metric because it takes into consideration the imbalance of the data set.



d. Without doing any cross validation, the test F1 scores are much worse than the train F1 scores. This shows that in testing, the classifier performs poorly compared to training.



e. With validation, there is a downwards trend in the F1 score. It is difficult to discern a proper correlation between which segment of data was used for the validation set and improved performance.

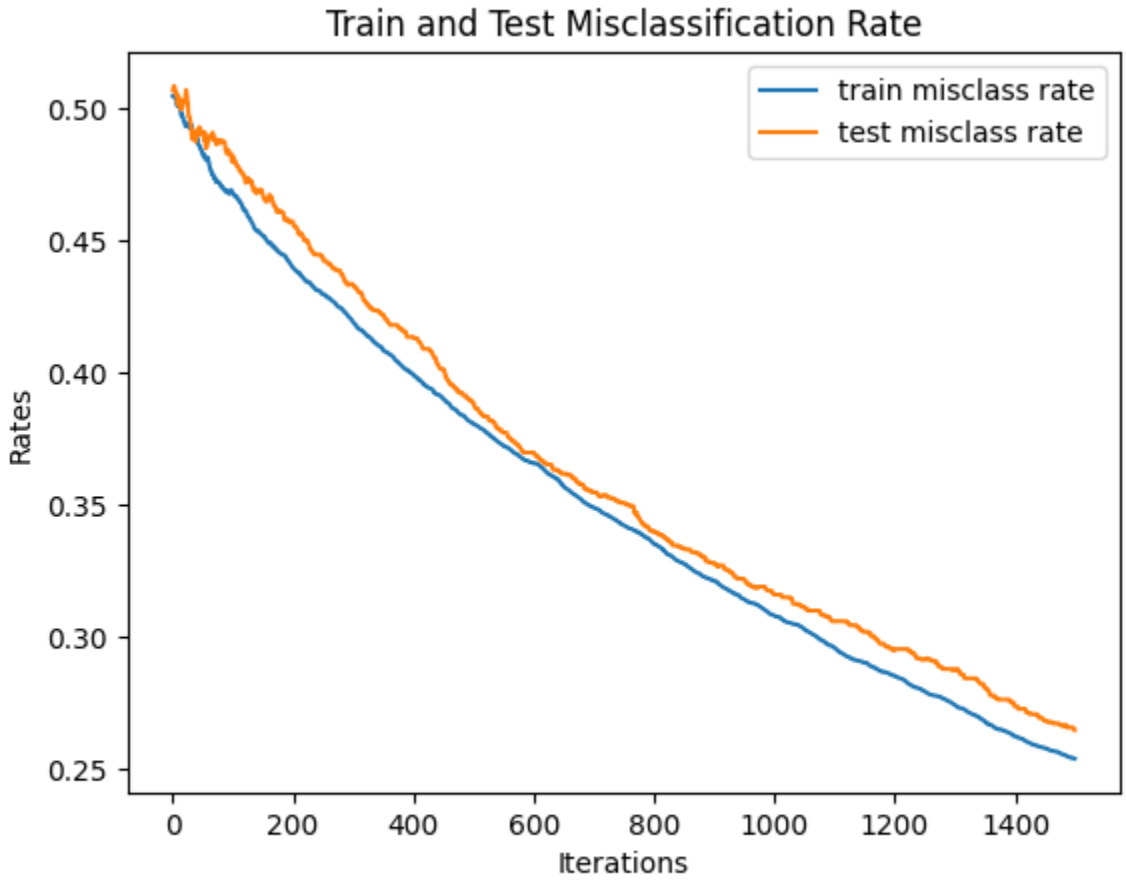


f. The best average F1 score of the validation set does correspond to a good F1 score over the test because the plotted lines visually show proximity between the training and test average F1 scores. Yes, this was a successful venture.

3. Logistic regression for Binary MNIST

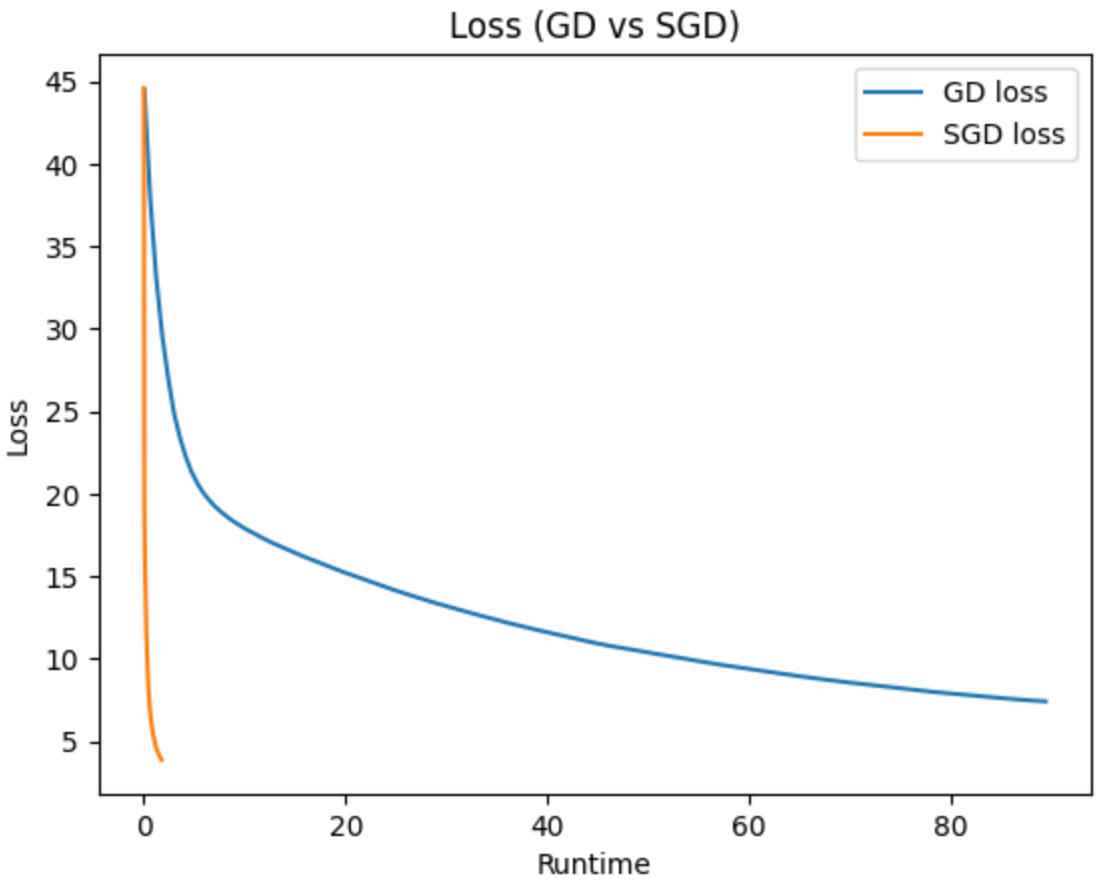
   a. The gradient of the logistic loss function is:

$$\nabla f_i(\theta) = \sigma(y_i x_i^T \theta) - 1) \cdot y_i x_i,$$

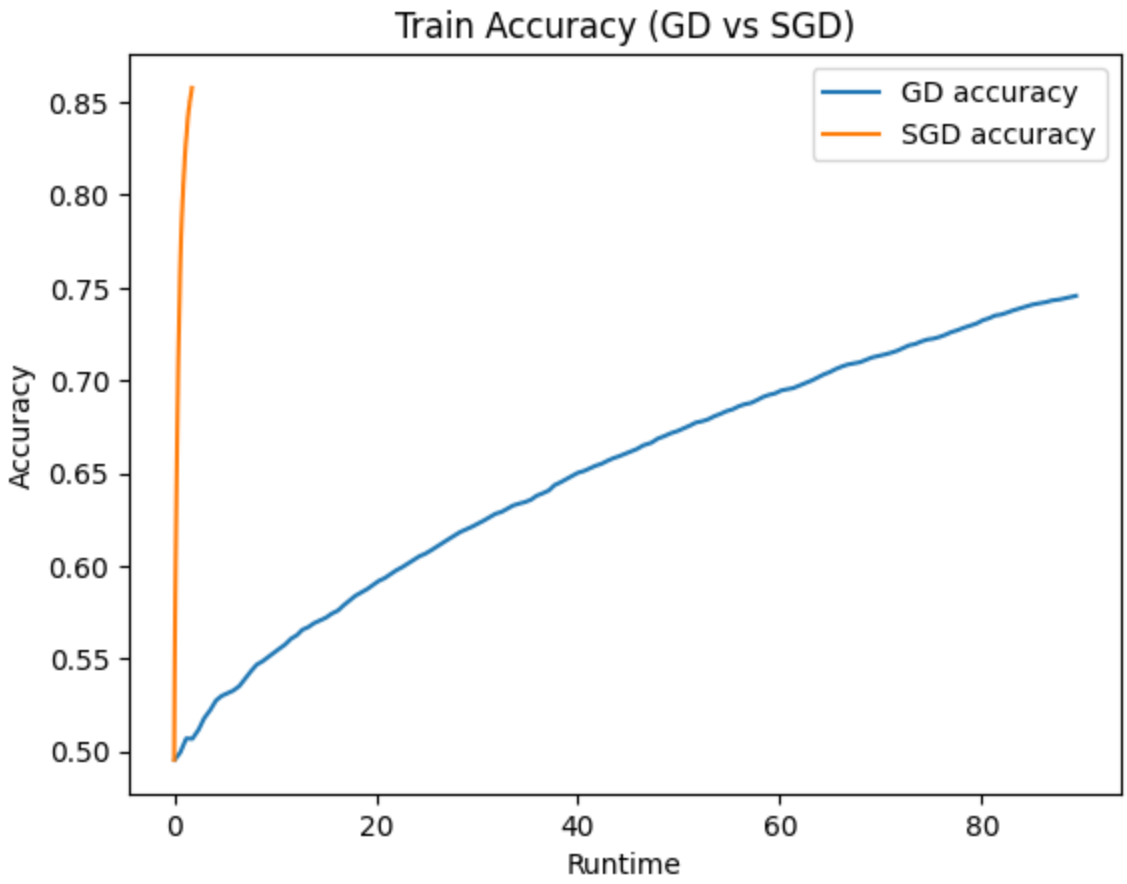


   b.
```
final train accuracy:  0.7460775167500636
final test accuracy:  0.7353088900050226
```

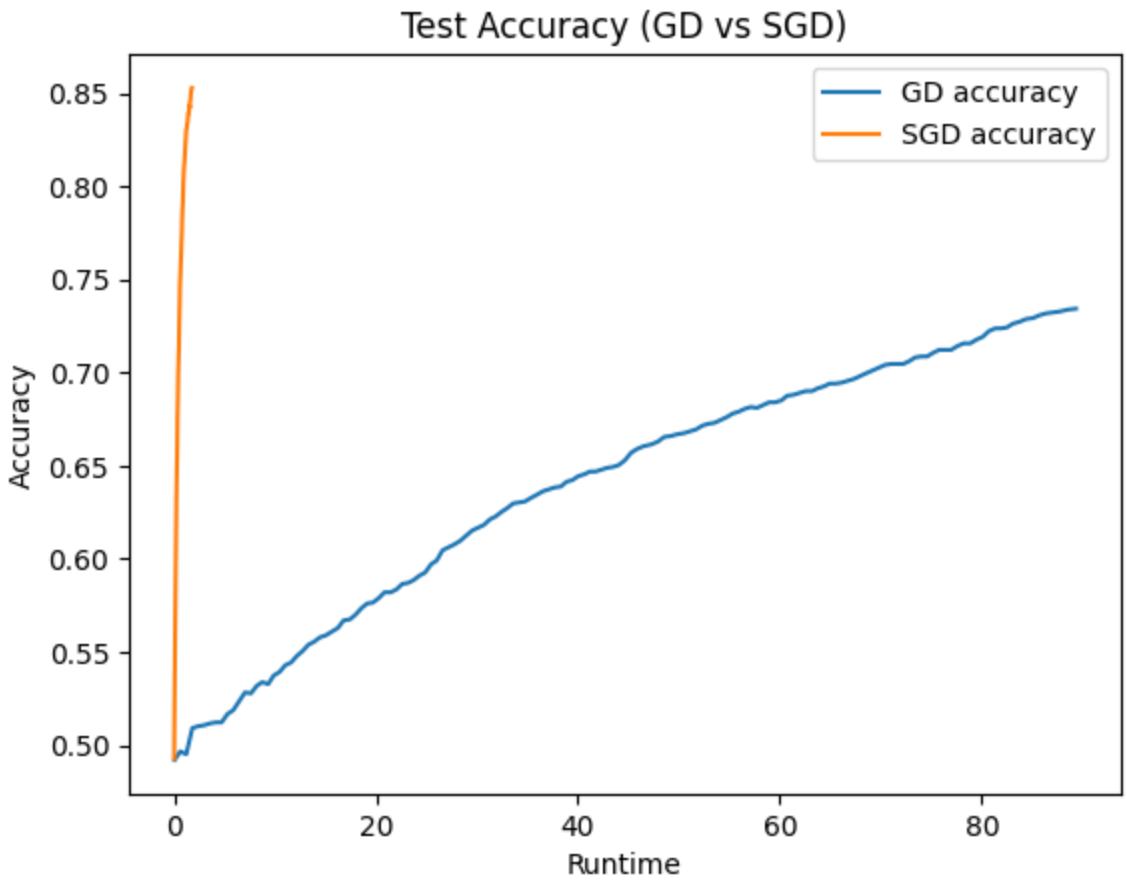
c.


Loss (GD vs SGD)
GD loss
SGD loss
Loss
Runtime
45
40
35
30
25
20
15
10
5
0
20
40
60
80

d.


Train Accuracy (GD vs SGD)
GD accuracy
SGD accuracy
Accuracy
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0
20
40
60
80
Runtime

e.



f. The pros of Stochastic Gradient Descent are that it is very accurate and quick. The cons are that it requires a large data set to properly sample its batches from and that it has high measurable loss. The pros of gradient descent are that it can be reliably accurate with a smaller data sample. The cons are that it is not as reliable as Stochastic Gradient Descent and takes longer.