Q1

1. $f(w) = \alpha w^2 - \beta w + \gamma \; => f'(w) = 2\alpha w - \beta => f''(w) \; = \; 2\alpha$
   Since $f''(w) \; >= \; 0$, the second derivative is non-negative, therefore $f(w)$ is convex

2. $f(w) = -\log(\alpha w) \, subsituting \; \alpha w \; with \; x => f'(x) = -\frac{1}{x} => f''(x) = \frac{1}{x^2} \; . Since \; f'' > \; 0, f(w) \; is \; convex$

3. $f(w)$ is a convex function as sum of convex functions is a convex function, $||w||$ is a convex as it is a norm, λ is always positive and $||Xw - y||$ is a convex with a linear system of $||Xw - y||$

4. $1 + \exp(-y_i w^T x_i)$ is linear and hence is convex hence, $f'(w) = \frac{1}{1+e(-x)}$ and $f''(w)$ is a sigmoid a function which is never negative hence $f(w)$ is a convex function

5. $|w^T x_i - y_i|$ is a convex function and hence $|w^T x_i - y_i| - \in$ is also a convex function. $f(w)$ is a convex function because maximum of a convex function is also a convex function with λ also greater than 0.

Q2

Q2.1

Training error: 0.002
Validation error: 0.074
Number of nonZeros: 101
Number of Iterations: 36

Q2.2

Training error: 0.000
Validation error: 0.052
Number of nonZeros: 71
Number of Iterations: 78

Q2.3

Training error: 0.000
Validation error: 0.040
Number of nonZeros: 25

Q2.4

L2- Regularization produces the most non-zero values as the weight is never 0 but very close to 0 hence, never discarding the feature

L1- Regularization produces comparatively less non-zero values due to the minimum often found at 0 moreover, the value of lambda being 1 result's in the minimum being past the origin

L0- Regularization produces the least non-zero values as it uses greedy forward selection process

Q2.5

L2 Scikit
Training error: 0.002
Validation error: 0.074
Number of nonZeroes: 101

L1 Scikit
Training error: 0.000
Validation error: 0.052
Number of nonZeroes: 71
Our implementation yields the same result as scikit implementation

Q2.6

1. $f(w) = \frac{1}{2}((w-2)^2 + 1) + \lambda\sqrt{|w|}$
2. $\arg min = 2 \ and \ minimum \ value \ is \ 1$
3. $\arg min = 0$
4. $\arg min = 1.605 \ and \ minimum \ value \ is \ 1.844$
5. $\arg min = 0 \ and \ minimum \ value \ is \ 2.5$
6. It behaves more like L1 regularization because the co-efficient goes to 0 thus making the feature irrelevant
7. It is not a convex optimization problem as the regularization function is not a convex function

Q3

Q3.1

$W^T = \begin{matrix} 2 & 2 & 3 \\ -1 & -2 & -1 \end{matrix}$

$class\ label\ is\ the\ maximum\ value$:

$$w_{11}x_1 + w_{12}x_2 = (2)(1) + (-1)(1) = 1$$
$$w_{12}x_1 + w_{22}x_2 = (2)(1) + (-2)(1) = 0$$
$$w_{13}x_1 + w_{23}x_2 = (3)(1) + (-1)(1) = 2$$

Hence, the example belongs to the 3rd class

Q3.2

Training error: 0.084
Validation error: 0.070

Q3.3

$f(W) = -w^T x_i + \log\left(\sum_{c'=1}^{k} \exp\,(w^T x_i)\right)$

$$\frac{df}{dW_{cj}} = -I(y_i = c)(x_i)_j + \frac{\exp(wc'^T x_i)}{\sum_{c'=1}^{k} \exp\left(wc'^T x_i\right)}(x_i)_j$$

$$\frac{df}{dW_{cj}} = \sum_{i=1}^{n} (x_{ij})\left(\frac{\exp(wc'^T x_i)}{\sum_{c'=1}^{k} \exp\left(wc'^T x_i\right)} - I(y_i = c)\right)$$

Q3.4

Training error: 0.000
Validation error: 0.008

Q3.5

One V all
Training error: 0.084
Validation error: 0.070

Softmax
Training error: 0.000
Validation error: 0.008

Our implementation yields the same result as scikit implementation

Q3.6

1. $O(ndkT)$
2. $O(tdk)$

Q4

1. One should not promise to provide relevant factors as there is so particular set of factors that will help prediction as different regularization and models will have different factors

2. Incorrect predictions because each feature is being treated independently and ignoring any dependencies if there exists any, like the sick and not sick example in class

3. L1 loss is robust to outliers, whereas L1 regularization works for feature selection wherein we have a lot of features

4. Sparse solutions are L1 regularization; Convex solutions is L2 regularization and Unique solutions is L0 regularization

5. As lambda increases the sparsity of the solutions increases as more coefficients go to zero. As lambda increases, the model underfits leading to decrease in approximate error but increase in training error

6. Could use an multi class SVM Kernel with support vectors

7. Linearly separable in 3 dimensions means that there exists a plane that separate all the different points from each other

8. We use logistic loss instead of minimizing classification errors because logistic loss returns a set of probabilities for the features it classifies and we can smooth the function using log-sum-exp

9. Support vectors are required in SVM's and these are the vectors which are closest of the line of fit, these are the only factors on which the fit of the line depends

10. The perceptron algorithm requires the data to be linearly separable for us to fit a linear classifier

11. We would use multi-class loss instead of binary SVMs in a one-vs-all framework because we have multiple columns and to better classify the data multiple classes would yield better errors

12. Sigma affects the width of the Gaussian distribution, it has an indirect relationship with overfitting and hence training error