Part I: Loss minimization with gradient descent

1. Given: $L(w_1, w_2) = -10(0.4\cos(w_1) - w_1^2 - 0.2w_2^7 + \sin(w_2))e^{-w_1^2-w_2^2}$

$$\frac{\partial L(w_1,w_2)}{\partial w_1} = -10(-0.4\sin(w_1) - 2w_1)e^{-w_1^2-w_2^2}$$

$$- 10(0.4\cos(w_1) - w_1^2 - 0.2w_2^7 + \sin(w_2))(-2w_1)e^{-w_1^2-w_2^2} =$$

$$= -10e^{-w_1^2-w_2^2}(-0.4\sin(w_1) - 2\sin(w_2)\,w_1 - 0.8\cos(w_1)\,w_1$$

$$- 2w_1 + 2w_1^3 + 0.4w_1w_2^7)$$

$$= e^{-w_1^2-w_2^2}[-20w_1^3 + 4\sin(w_1)$$

$$+ w_1(-4w_2^7 + 20\sin(w_2) + 8\cos(w1) + 20)]$$

$$\frac{\partial L(w_1,w_2)}{\partial w_2} = -10(-1.4w_2^6 + \cos(w_2))e^{-w_1^2-w_2^2}$$

$$- 10(0.4\cos(w_1) - w_1^2 - 0.2w_2^7 + \sin(w_2))(-2w_2)e^{-w_1^2-w_2^2}$$

$$= -10e^{-w_1^2-w_2^2}(\cos(w_2) - 2\sin(w_2)\,w_2 - 0.8\cos(w_1)\,w_2$$

$$+ 0.4w_2^8 - 1.4w_2^6 + 2w_1^2w_2)$$

$$= e^{-w_1^2-w_2^2}[-4w_2^8 + 14w_2^6 - 10\cos(w_2) + w_2(-20w_1^2 + 20\sin(w_2)$$

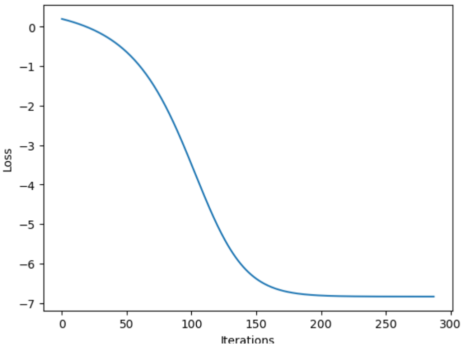
$$+ 8\cos\,(w_1))]$$

And of course:

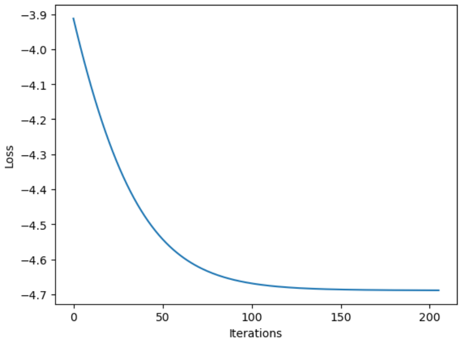
$$\nabla L(w_1, w_2) = \left[\frac{\partial L(w_1,w_2)}{\partial w_1}, \frac{\partial L(w_1,w_2)}{\partial w_2}\right]$$

3.

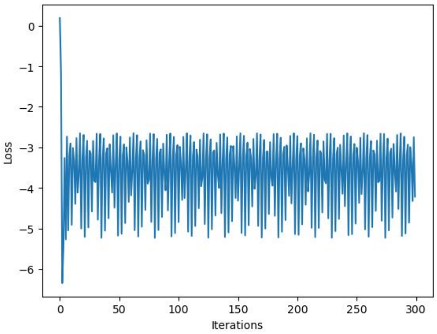
- $(w_1, w_2) = (1,1) \qquad \eta = 0.001$
  Loss is decreasing in steady rate and set on value of around -6.84. low step size keeps the curve smooth on one hand and avoid overshoot on the other hand. it might hit a global minimum (we can't be sure).



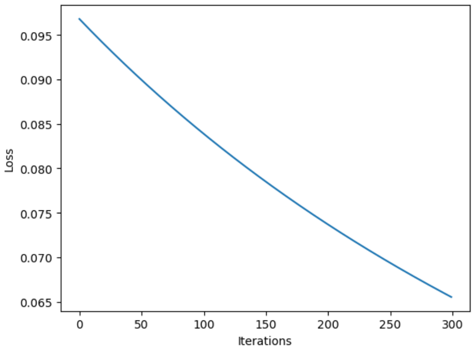
- $(w_1, w_2) = (0, -2.2)$
  $\eta = 0.001$
  Different starting point led us to converge to a different local minimum, or saddle point. We know that is not the global minimum, because the final loss value is higher -4.69.



- $(w_1, w_2) = (1,1) \qquad \eta = 0.1$
  Same starting point as the first run, but step size is 100 time bigger. The big step size is dropping the loss value fast but cause it to move around the global minimum point, and not be able to converge.



- $(w_1, w_2) = (2,2) \qquad \eta = 0.001$
  By the values of the weights in the final iteration, it seems to converge to a different local minimum. This local minimum might be further, and it seems that it need bigger step in order to converge to this minimum point (or more iterations).



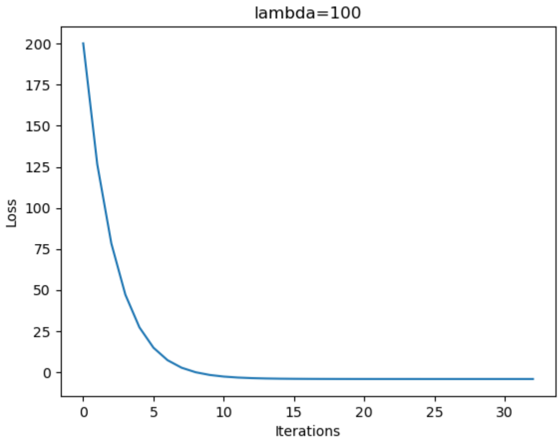
4. $L(w_1, w_2) = -10(0.4\cos(w_1) - w_1^2 - 0.2w_2^7 + \sin(w_2))e^{-w_1^2 - w_2^2} + \lambda\|w\|_2^2 = -10(0.4\cos(w_1) - w_1^2 - 0.2w_2^7 + \sin(w_2))e^{-w_1^2 - w_2^2} + \lambda w_1^2 + \lambda w_2^2$

   The gradient is approximately the same except the addition of the regularization terms derivatives.

$$\frac{\partial L(w_1, w_2)}{\partial w_1} = e^{-w_1^2 - w_2^2}[-20w_1^3 + 4\sin(w_1) + w_1(-4w_2^7 + 20\sin(w_2) + 8\cos(w1) + 20)] + 2\lambda w_1$$

$$\frac{\partial L(w_1, w_2)}{\partial w_2} = e^{-w_1^2 - w_2^2}[-4w_2^8 + 14w_2^6 - 10\cos(w_2) + w_2(-20w_1^2 + 20\sin(w_2) + 8\cos(w_1))] + 2\lambda w_2$$

5. Adding regularization of $\lambda = 0.01$ didn't affect much the result, and the weights settled on the same values as in section 3. On the other hand, adding $\lambda = 100$ restrained the loss function optimization, and kept w_1 *and* w_2 much smaller. Adding the regularization caused to converge to a different local minimum, but with the benefit of much smaller weight vector.



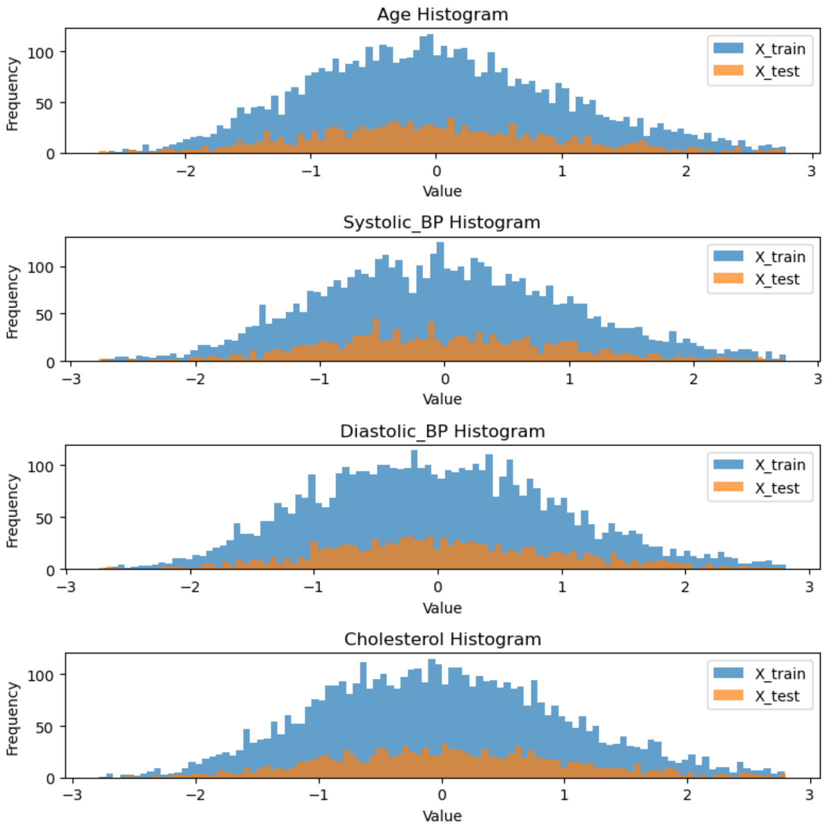
Part 2: Binary classifiers

1. Train-test split and preprocess:
   - We first split the dataset into train and test set, and check for missing values
   - Next, we checked for outliers in the data. After replacing the outliers with Nan, we replaced the Nan values with **K-NN algorithm to find the closest neighbor for each missing value**. This method has proven improving prediction of cervical cancer [1]
   - Last thing, we standardized the data.

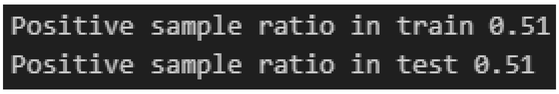
2. Data visualization and exploration

We examined the features distributions - Its look like the features are close to normally distributed, and each feature distribution is the same for train and test.

After that we preformed features standardization:



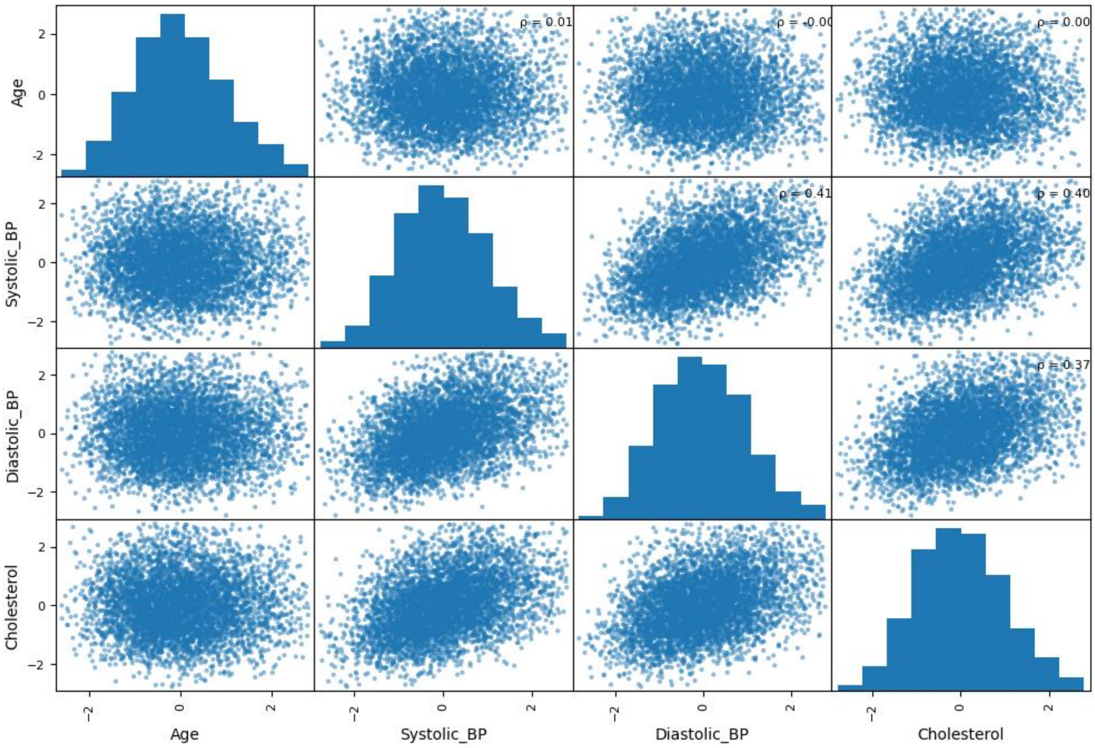
We also verified that label ratio is the same for train and test:



```
Positive sample ratio in train 0.51
Positive sample ratio in test 0.51
```

The importance of balance classes is significant. If we evaluate our performance on model with class imbalance, it can lead to wrong conclusions on how the model is performing (a biased model) – for example if have dataset with 99% healthy patients and 1% ill, if the model will predict $\hat{y} = helathy$ for each sample, it will get accuracy of 99%. Obviously, it is not we search for when evaluating the model. One way to deal with unbalanced dataset is to under sample the class which is larger in size. This way we can have 2 balanced classes. [2]

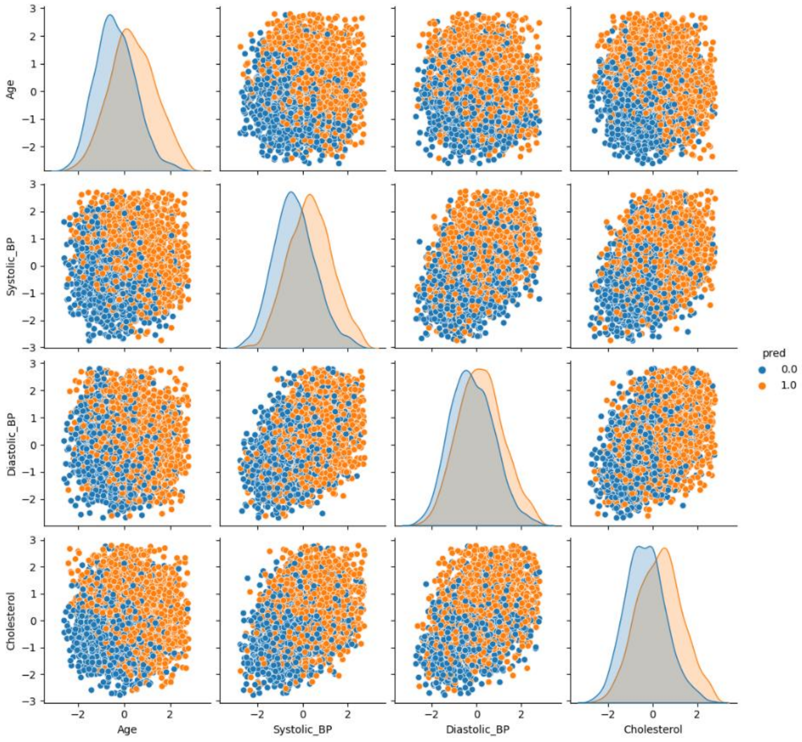
After the features are preprocessed and standardized, we went on and examined their relationship. We used pandas' scatter_matrix() to plot relationships between features:

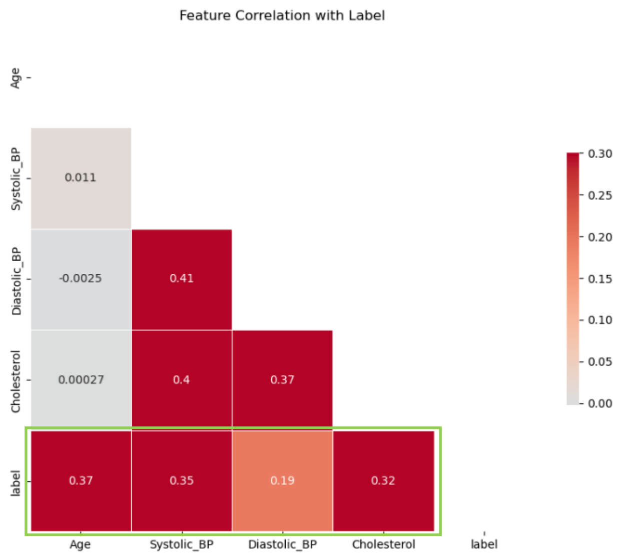


We can tell from the plot, that there is strong correlation ($\rho = 0.41$) between systolic and diastolic BP. This is no surprise as it observed before. [3]

Second thing we can observe is the strong correlation ($\rho = 0.40$) between higher cholesterol level and increased BP. This is also well-known phenomena. [4]

Next thing, we will examine the relationship between features and the label using seaborn pair plot and heatmap:



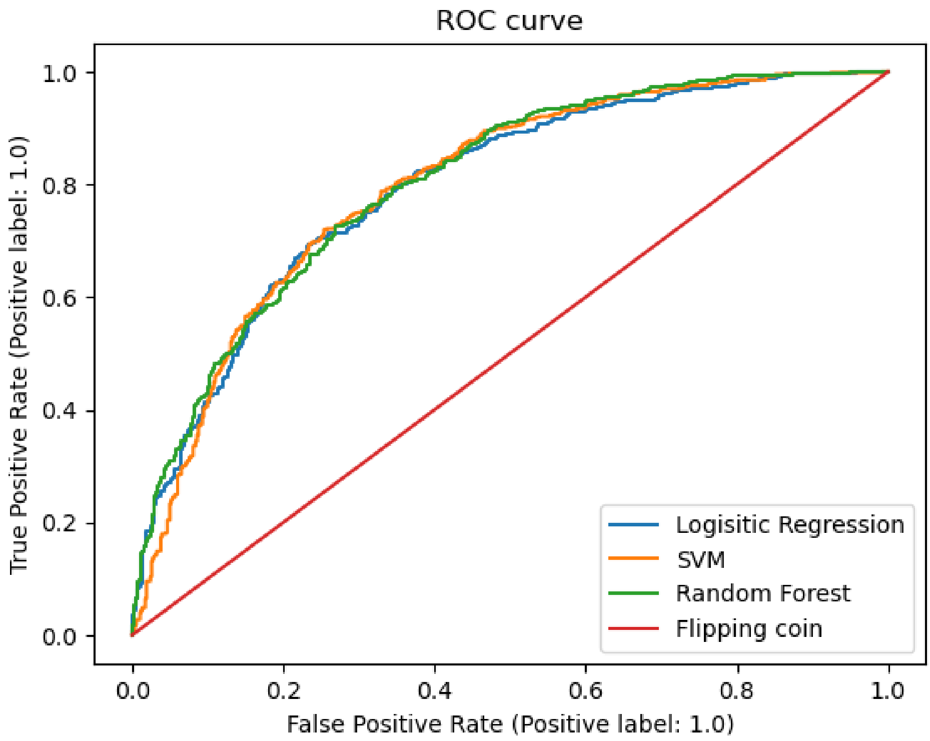


It seems from the plots that increasing age, high BP and high cholesterol are connected to positive prediction of diabetic retinopathy. there is no surprise here too. Age, High blood pressure and high levels of cholesterol are risk factors for diabetic retinopathy (DR) in diabetic patients. [5]. Also, we can see from the heatmap plot that Age and systolic BP are the most important features for predicting DR – their correlation with the label is the most significant.

3. Choose, build and optimize Machine Learning Models
    a. **Logistic regression** – we preformed 5-fold cross validation to choose hyper parameters. We examined different C Values and L_1 and L_2 regularization. Best model in terms of $AUROC = 0.819$ with $C = 1,\ penalty = L_1$.
    b. **SVM** – we examined 'poly' and 'rbf' kernel and changing C values. The best model chosen: $AUROC = 0.815$ with $C = 0.1,\ kernel = rbf$
    c. **Random forest** – We examined different criterions (gini and entropy), changing model depth and number of max features to choose from (sqrt, log2 or none). Model chosen with best $AUROC = 0.82$ with $criterion = entropy,$ $maxdepth = 6, maxfeatures = \sqrt{N}$.
        Note: we didn't check $maxfeatures = \log_2 N$, because $N = 4$ and $\log_2 N = \sqrt{N} = 2$

Next, we evaluated the best model on the test set:

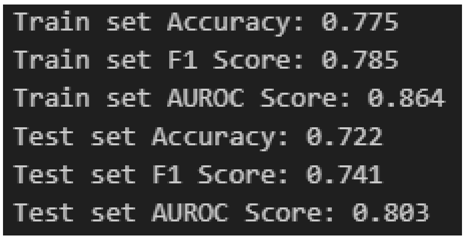
Model	Accuracy	F1	AUROC
Logistic Regression	0.717	0.726	0.795
SVM	**0.728**	**0.748**	0.797
Random Forest	0.722	0.741	**0.803**



It looks like that SVM is the best model in terms of Accuracy and F1 score. In terms of AUROC the best model is random forest. By looking at the ROC curves, it is hard to tell the difference between the models. They all performed better than the naive classifier, but overall, nonlinear models (SVM with rbf kernel and random forest in our case), preformed better on the test set, compared to the linear model. The non-linearity introduced by those models, allows them to better represent the data. On the other hand, they tend to overfit easily. This can be addressed by adding regularization term, same as we added, which reduces the tendency to overfit. And as we can see, indeed the best models is SVM with rbf (nonlinear) with regularization ($C = 10$).

4. Random forest feature selection

i.



```
Train set Accuracy: 0.775
Train set F1 Score: 0.785
Train set AUROC Score: 0.864
Test set Accuracy: 0.722
Test set F1 Score: 0.741
Test set AUROC Score: 0.803
```

ii. We can see that our model is overfitted. Test metrics are lower significantly from the train. Random forest is ensemble model which average different decision trees with Bagging (bootstrap aggregating) on the training samples. It is also use bagging to pick a random subset of features. So, in general we take multiple decision trees, which are low-bias-high-variance models and reduce their variance.

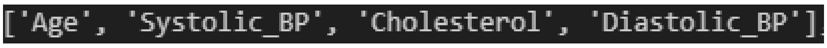
There can be several reasons for overfit: first, the decision tree depth hyper-parameter is too high – the trees are too deep, and overfitted on the train set, and failed to generalize. Another reason is minimum number of samples per leaf. Sklearn default is 1, which may cause to many splits in the tree.

To reduce overfit in random forest (In matter of fact, in each decision tree itself), we can use a method called pruning. We will check on validation set the contribution of each branch (with or without him). If this branch doesn't affect estimator performance, we can remove it. We continue to next branch, and to next one, until there are no more branches to prune.

iii. Few main advantages:

1. Random forests are more interpretable than SVM by their nature, and we can check feature importance. SVM on the other hand is not easily interpretable, especially when dealing with nonlinear kernels.
2. Random forests are less prone to overfit. Their stochastic ensemble nature is more robust than the single decision tree, and they usually generalize well. SVM is easily affected by outliers.

iv. The 2 most important features according to both Random Forest model and Logistic regression are age and systolic BP.

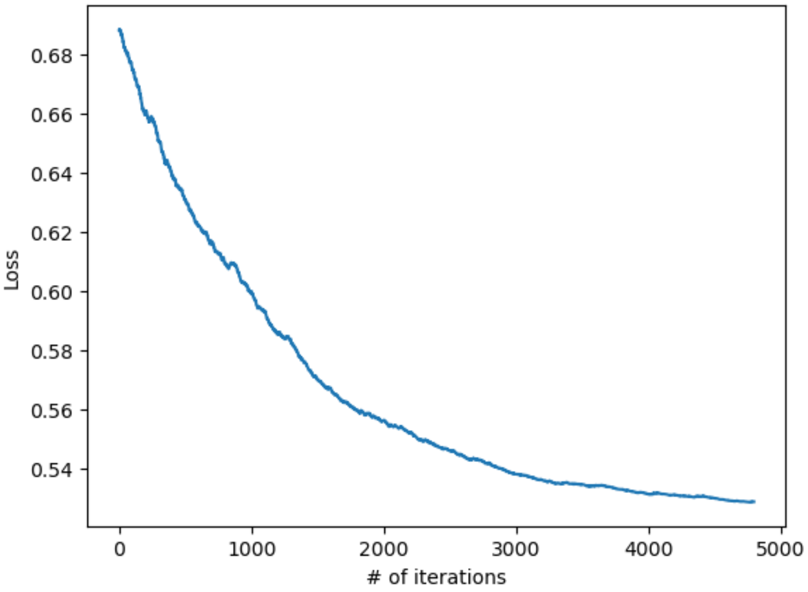


```
['Age', 'Systolic_BP', 'Cholesterol', 'Diastolic_BP']
```

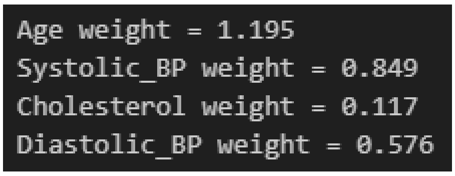
We saw earlier, in the data exploration part, that those are the 2 most important features in terms of correlation with the label. In fact, the order of features importance is match exactly to the order of high-to-low correlation with the label, so it is not surprising that this is the result.

2.3 Logistic regression from scratch

After implementing the class methods, we trained the model on the training set. A minor preprocessing has been applied, to remove outliers and standardized the feature. After model fitting, we got slowly decreasing loss as we expected:



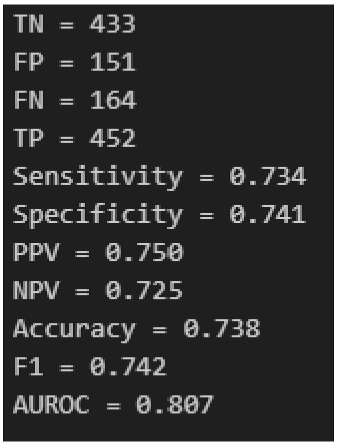
Next, we sorted the trained weights:



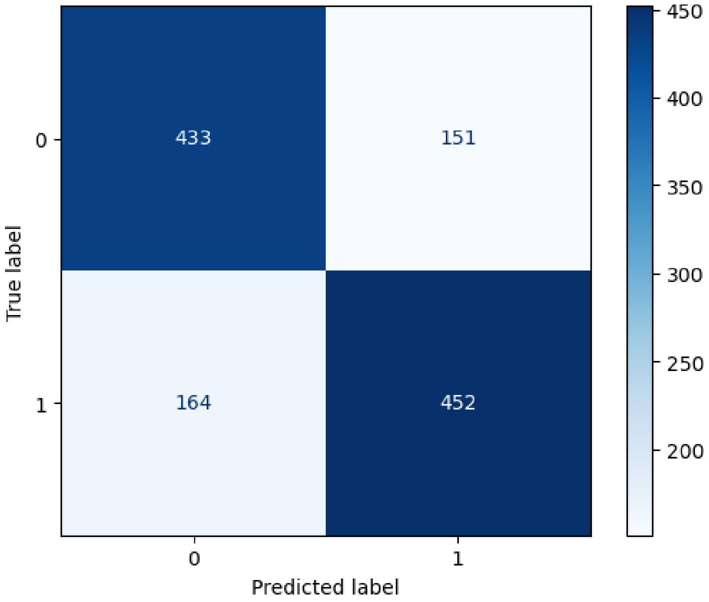
```
Age weight = 1.195
Systolic_BP weight = 0.849
Cholesterol weight = 0.117
Diastolic_BP weight = 0.576
```

We can see that the same weight order is present here, the same as in sklearn's logistic regression classifier.

Classification metrics:



```
TN = 433
FP = 151
FN = 164
TP = 452
Sensitivity = 0.734
Specificity = 0.741
PPV = 0.750
NPV = 0.725
Accuracy = 0.738
F1 = 0.742
AUROC = 0.807
```



The metrics are close to the result we got with sklearn classifier. There are slight differences, because we split the dataset differently.

Sources

[1] https://pubmed.ncbi.nlm.nih.gov/38170713/ " Improving prediction of cervical cancer using KNN imputer and multi-model ensemble learning" Turki Aljrees

[2] https://pubmed.ncbi.nlm.nih.gov/20411285/ " An approach for classification of highly imbalanced data using weighting and undersampling" Ashish Anand 1, Ganesan Pugalenthi, Gary B Fogel, P N Suganthan.

[3] https://pubmed.ncbi.nlm.nih.gov/18192832/ "Linear relationship between systolic and diastolic blood pressure monitored over 24 h: assessment and correlates", Benjamin Gavish, Iddo Z Ben-Dov, Michael Bursztyn

[4] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3075799/ "Relationship of Dietary Cholesterol to Blood Pressure": Masaru Sakurai,a Jeremiah Stamler,b Katsuyuki Miura,c Ian J Brown,d Hideaki Nakagawa,a Paul Elliott,d Hirotsugu Ueshima,c Queenie Chan,d Ioanna Tzoulaki,d Alan R Dyer,b Akira Okayama,e Liancheng Zhao,f and The INTERMAP Research Group

[5] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7478682/ "Prevalence and risk factors of diabetic retinopathy in diabetic patients" Li Yin, MS, Delong Zhang, MD, Qian Ren, MS, Xian Su, MS, and Zhaohui Sun, MD.