Hw2 ML in Healthcare:

Part i:

1.1) $L(w_1, w_2) = -10\left(0.4\cos(w_1) - w_1^2 - 0.2w_2^7 + \sin(w_2)\right) \cdot e^{-w_1^2 - w_2^2}$

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

$$\mathrm{I} \triangleq \frac{\partial L}{\partial w_1} = -10\left[\left(-0.4\sin(w_1) - 2w_1\right) \cdot e^{-w_1^2 - w_2^2} + \left(0.4\cos(w_1) - w_1^2 - 0.2w_2^7 + \sin(w_2)\right) \cdot \left(-2w_1 \cdot e^{-w_1^2 - w_2^2}\right)\right]$$

$$\mathrm{II} \triangleq \frac{\partial L}{\partial w_2} = -10\left[\left(-1.4w_2^6 + \cos(w_2)\right) e^{-w_1^2 - w_2^2} + \left(0.4\cos(w_1) - w_1^2 - 0.2w_2^7 + \sin(w_2)\right) \cdot \left(-2w_2 e^{-w_1^2 - w_2^2}\right)\right]$$

$$\boxed{\nabla L(w_1, w_2) = (\mathrm{I}, \mathrm{II})}$$

כלל מכפלת הנגזרות: $(f(x) \cdot g(x))' = f(x)' g(x) + f(x) g'(x)$

$(e^{x^2})' = 2x e^{x^2}$

$(\cos x)' = -\sin x$

$(\sin x)' = \cos x$

$\frac{\partial f(x)}{\partial y} = 0$

1.4) $L(w_1, w_2) = -10\left(0.4\cos w_1 - w_1^2 - 0.2w_2^7 + \sin w_2\right) e^{-w_1^2 - w_2^2} + \underbrace{\lambda \cdot \left(w_1^2 + w_2^2\right)}_{\text{regularization term}}$

$\frac{\partial L}{\partial w_1} = \mathrm{I} + 2\lambda w_1$

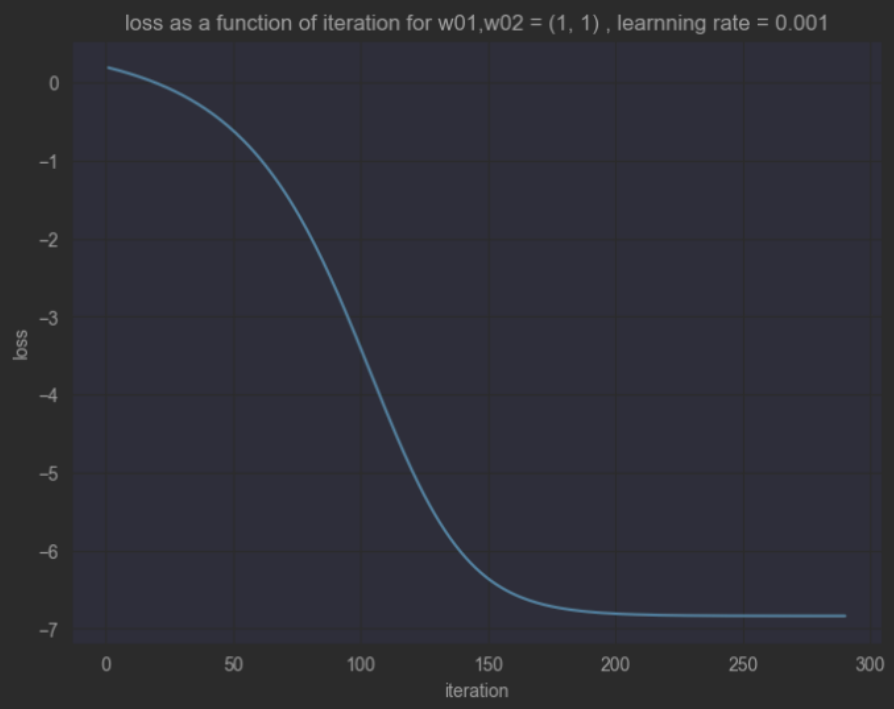
$\frac{\partial L}{\partial w_2} = \mathrm{II} + 2\lambda w_2$

$$\boxed{\nabla L(w_1, w_2) = \left(\mathrm{I} + 2\lambda w_1, \mathrm{II} + 2\lambda w_2\right)}$$

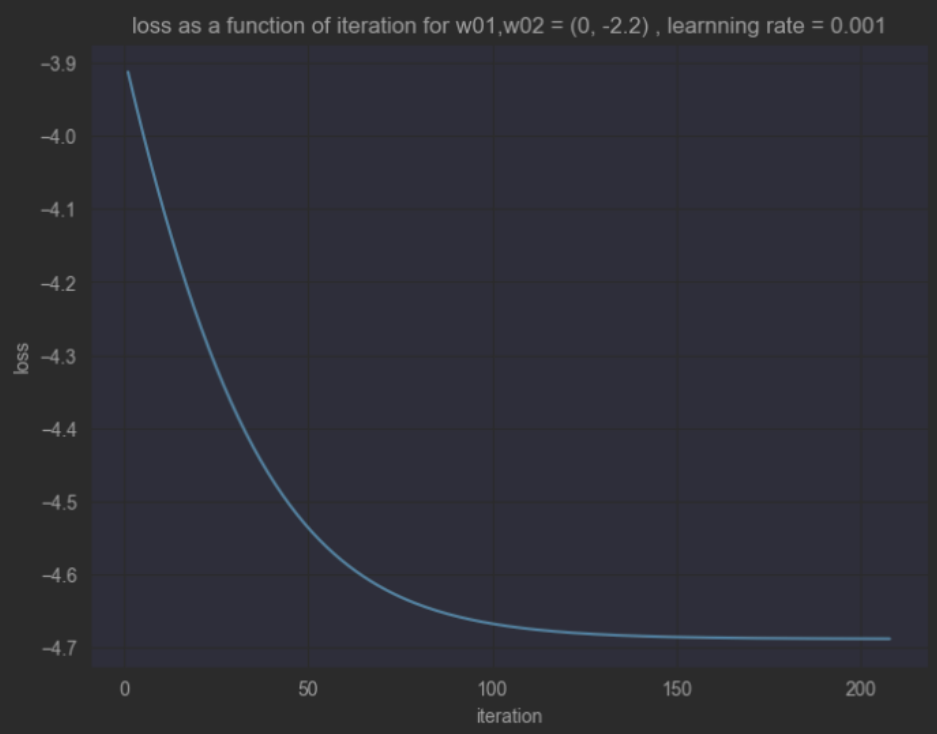
* We used I, II from (1.1), justification for that - linearity of summation - $(f(x) + g(x))' = f'(x) + g'(x)$

- From here and below- all answers are written at the jupyternotebook (might be simpler to red them there)

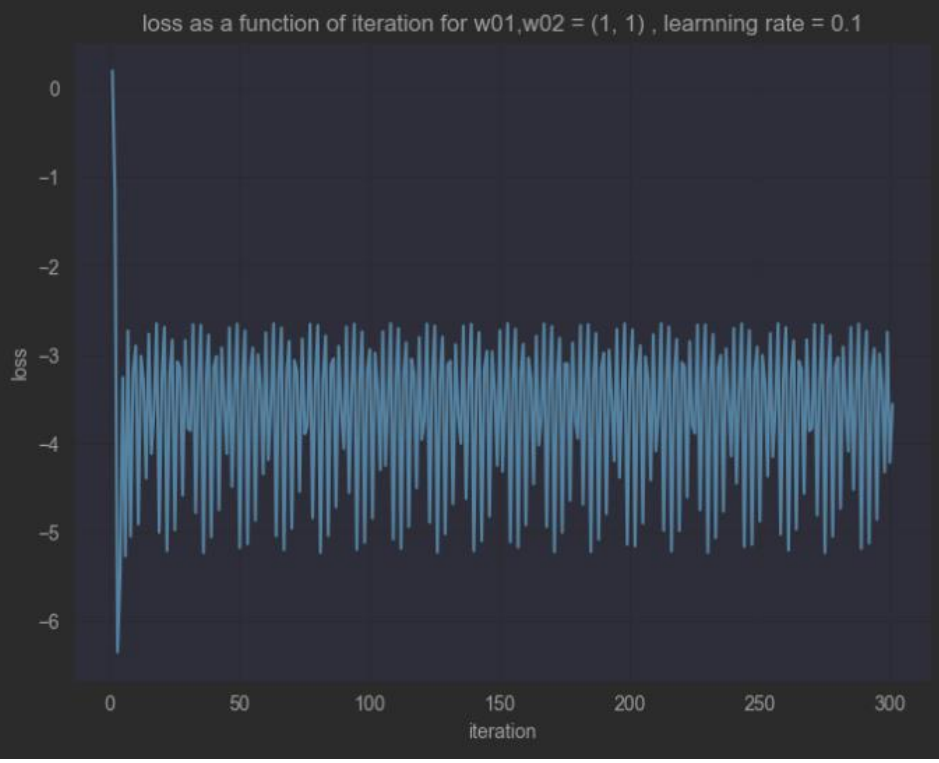
```
Optimal weights (w1,w2)=(0.001581214458306869, 0.4964921926906216)
The optimal loss = -6.837209294852138
```



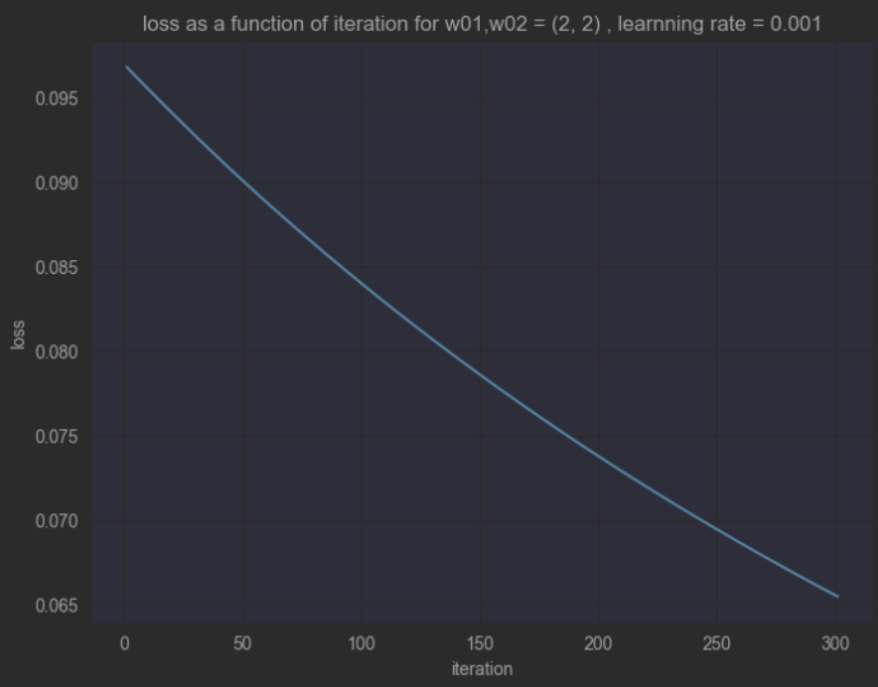

```
Optimal weights (w1,w2)=(0.0, -1.908769390536913)
The optimal loss = -4.688359059966798
```



```
Optimal weights (w1,w2)=(0.49146026667276954, 0.47730772088159834)
The optimal loss = -3.5609643259232966
```




```
Optimal weights (w1,w2)=(2.093983220084856, 2.0221014383514975)
The optimal loss = 0.06545202166601019
```



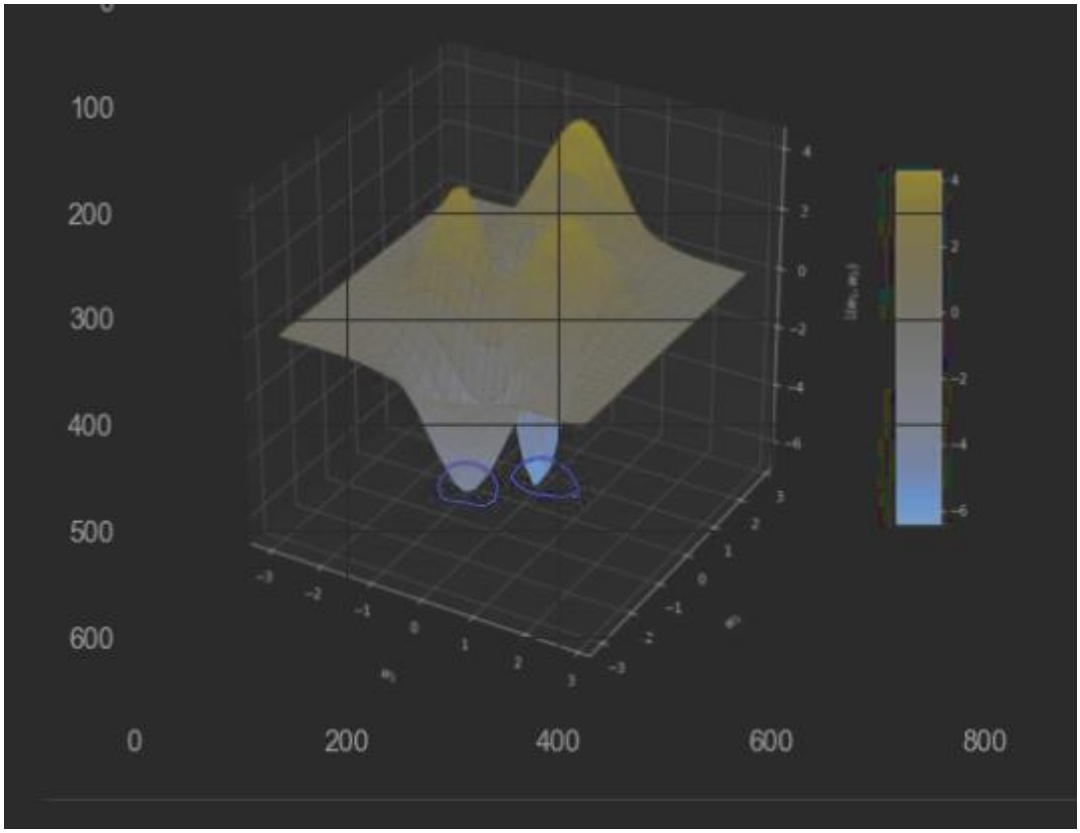
1.3:

The loss function is non convex so the minimal loss we will depend on the initial weights and the step size (learning rate) we have no guarantee of converging to the global optimum,the loss can converge to different local/global minima.

The first graph,grad_desc(1,1,0.001), converge to the global minima (optimal loss around -6) and the second graph-grad_desc(0,-2.2,0.001) converge to the local minima (optimal loss around -4) but with higher convergence rate - converge rapidly and there was no much change at our weight after 250 iteration so our algorithm stopped. those 2 minimum points can be seen in the plot at the next cell.

for grad_desc(2,2,0.001) - the last graph , we can see that the optimal weights are very similar to the initial weights and the loss is very small - so there was'nt a significant learning.

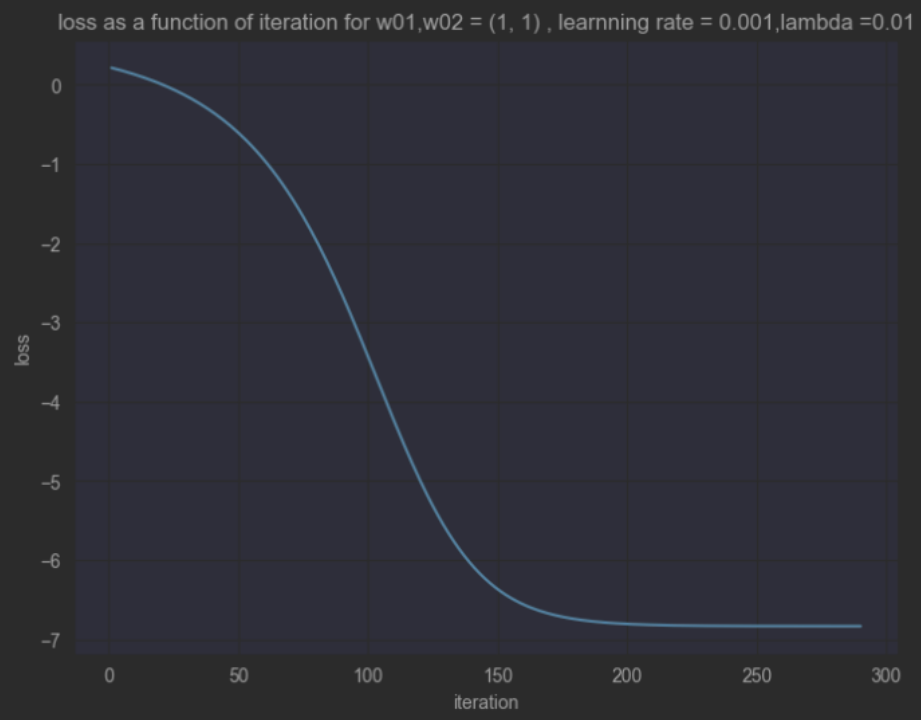
for grad_desc(1,1,0.1) the step size is larger than at the other setups so we can see more drastic changes at each iteration and we can see that our model does no converge to a certain minima - we can see oscillations and unsteady loss.



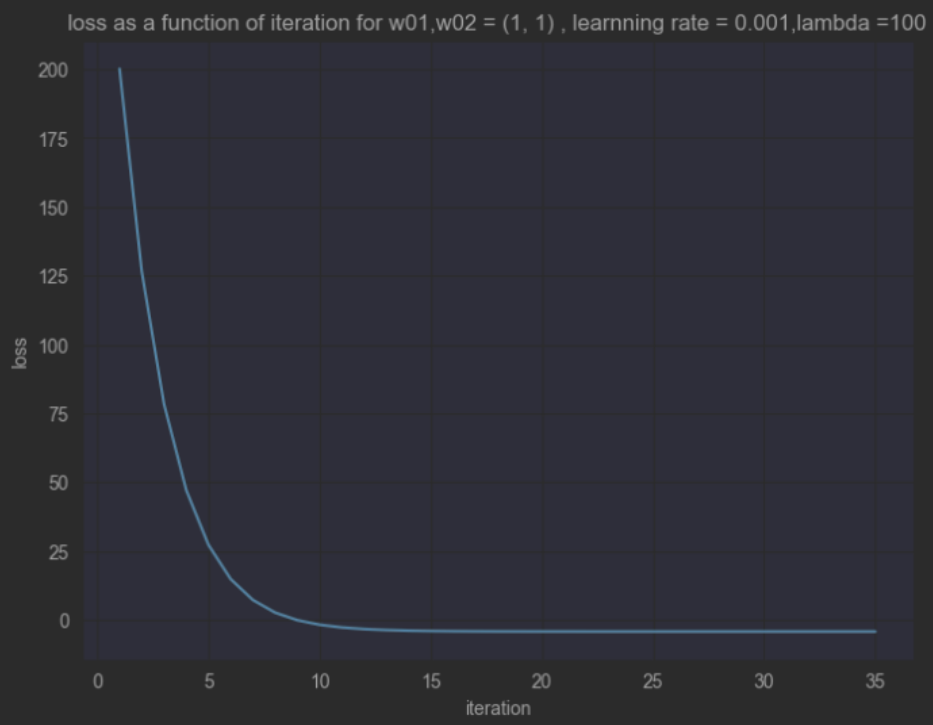
##1.4: We solved it above under 1.1.

##1.5:

```
Optimal weights (w1,w2)=(0.0015489296378002953, 0.49605952602776476)
The optimal loss = -6.8347829893499155
```

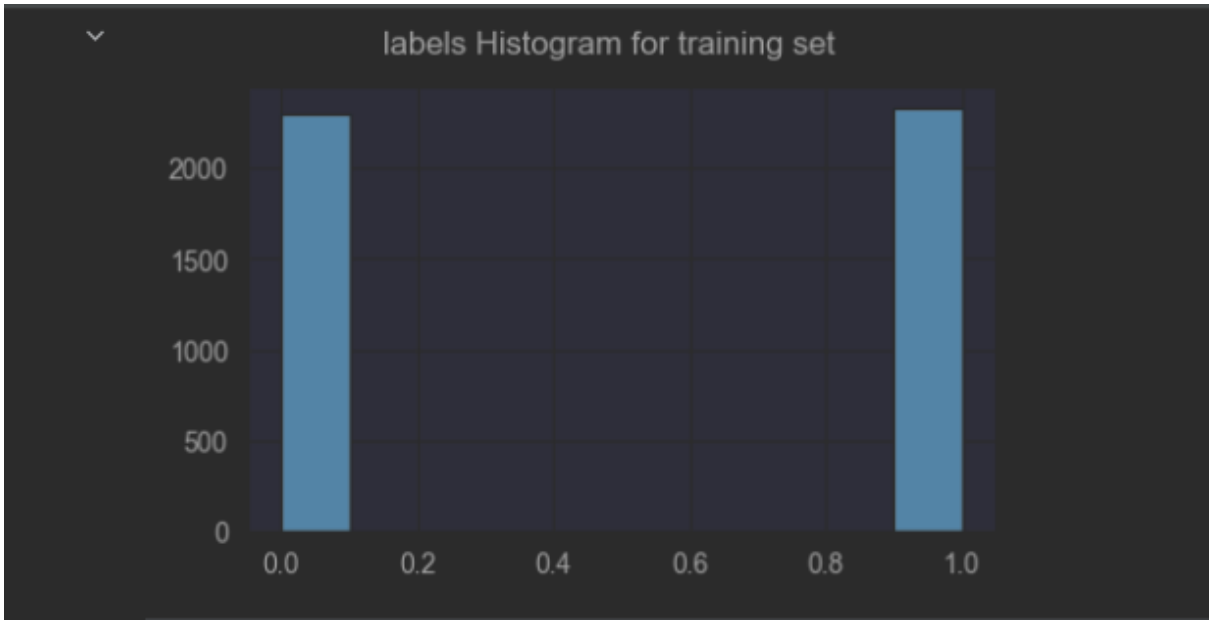


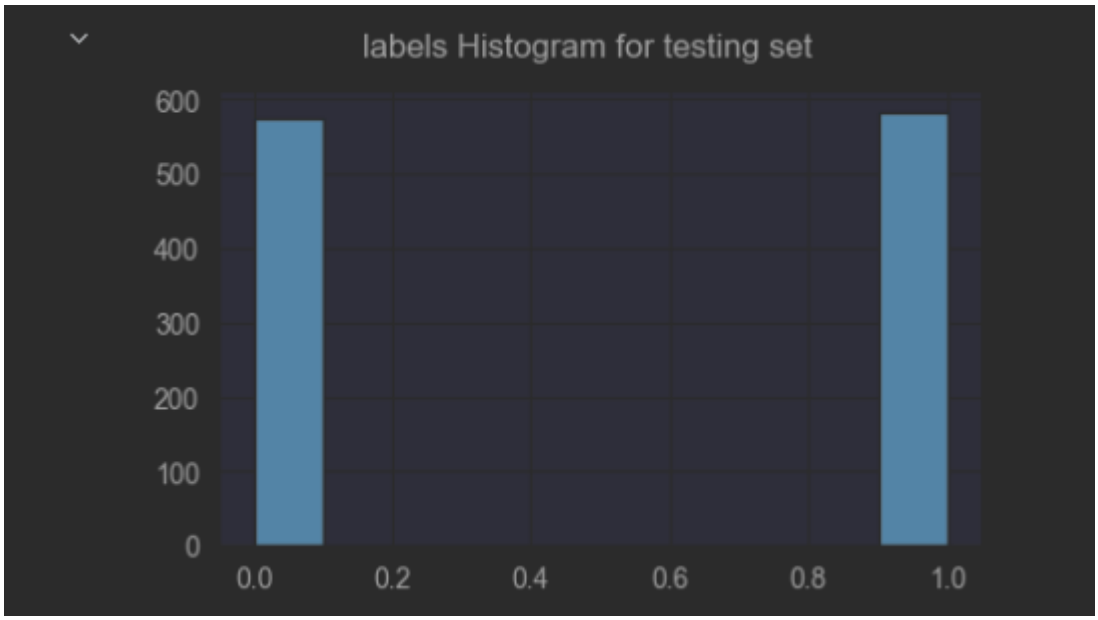
```
Optimal weights (w1,w2)=(0.0001362951524070234, 0.04797483514378552)
The optimal loss = -4.239105455875441
```



comparing results : for lambda=0.001 , we can say that lambda is very small, lambda equals zeros means no regularization at all, very small lambda will result in non significant regularization, so for this case as we expected the results are similar to section 1.3.
for lambda = 100 , we can see the effect of 'real' regularization, if lambda is infinite then the hypothesis will be equals to the bias, large lambda will result in underfitting, hence the loss will be larger, as we can see this is what we got here the loss is less negative (larger) and the convergence rate is very high, model converge after about 35 iterations only.

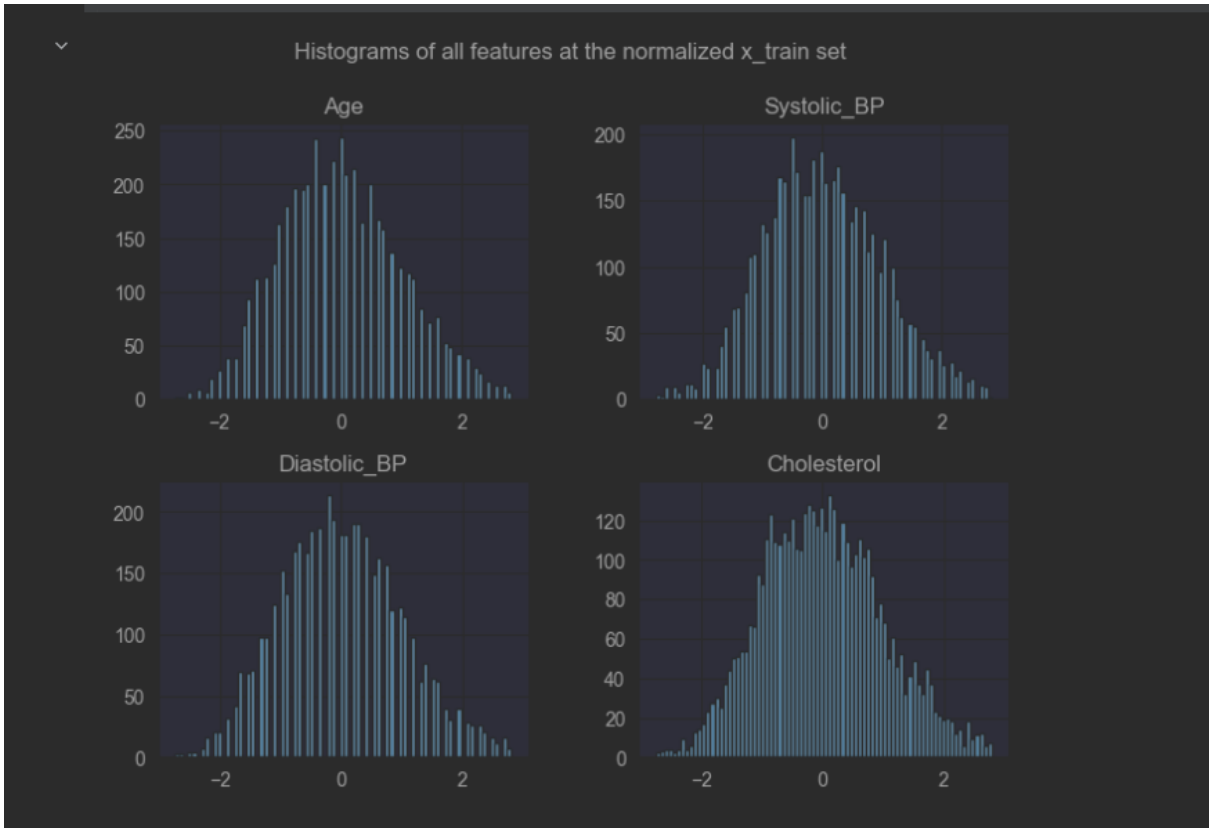
Part II: Linear vs. nonliner classifiers

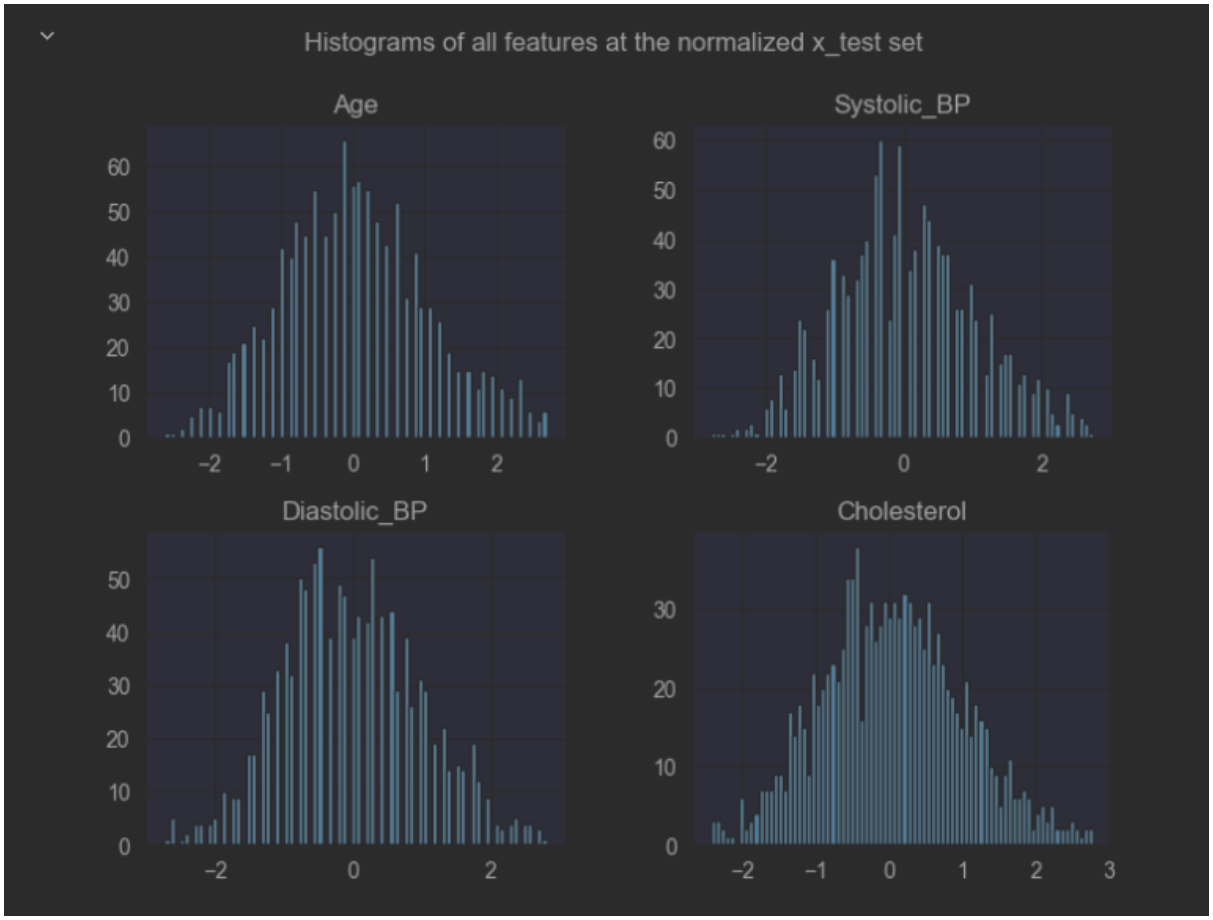





```
test set - 50.259% of samples are labeled as 1
train set - 50.259% of samples are labeled as 1
```

we can see that training and testing sets has about the same ratio of 0 and 1 labels - both sets has slightly more positive labels, the distribution of the labels was kept during the spilt!


Histograms of all features at the normalized x_train set
Age
Systolic_BP
Diastolic_BP
Cholesterol


Histograms of all features at the normalized x_test set
Age
Systolic_BP
Diastolic_BP
Cholesterol

questions:

i. What issues could an imbalance of labels in the training set rise? Search in the scientific

literature at least one possible solution. Explain how this solution should help.

ii. What labels imbalance between train and test cause? What is the solution we gave in

the tutorial?

answers:

i.

If our data contains 6000 examples and let us take it tp the extreme, let's say that 5990 of them are labeled as 0 and only 10 are labeled as 1 - which correspond to disease, then this is extremely unbalanced dataset, if we use this data for training our model do not have an option to learn about the disease category, hence it might learn to predict all examples as 0, and will not be able to perform well on a balanced dataset.

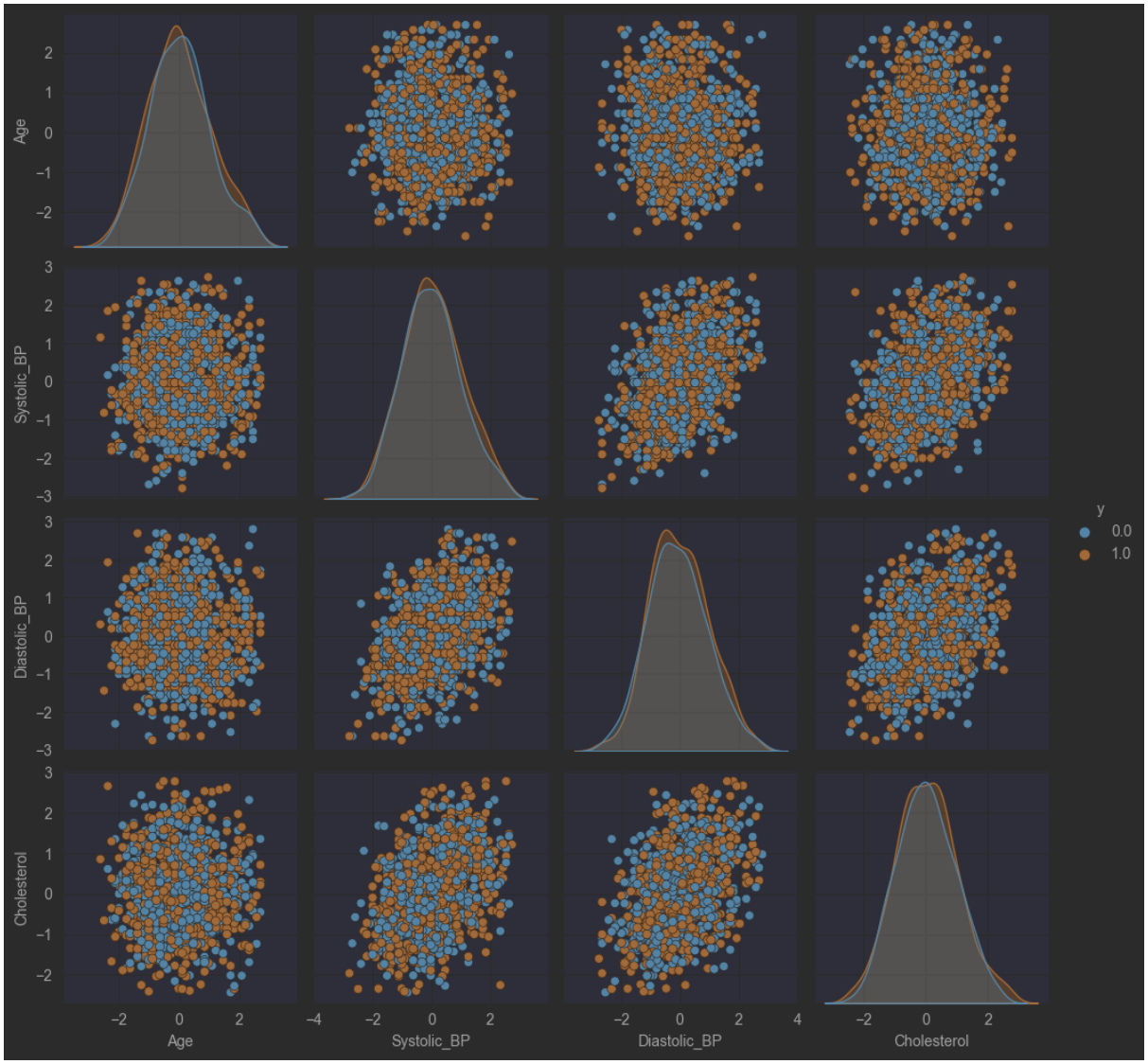
one method from literature to overcome this problem is SMOTE - Synthetic Minority Oversampling Technique, generate new data based on implications of old data. Basically, use the current inputs to generate new input rows that are unique but will have a label based on what the original data implies. In the case above, one simple way to think of this idea would be to add rows with label 1 to the data where the inputs represent total or partial similarities in values to current input features.

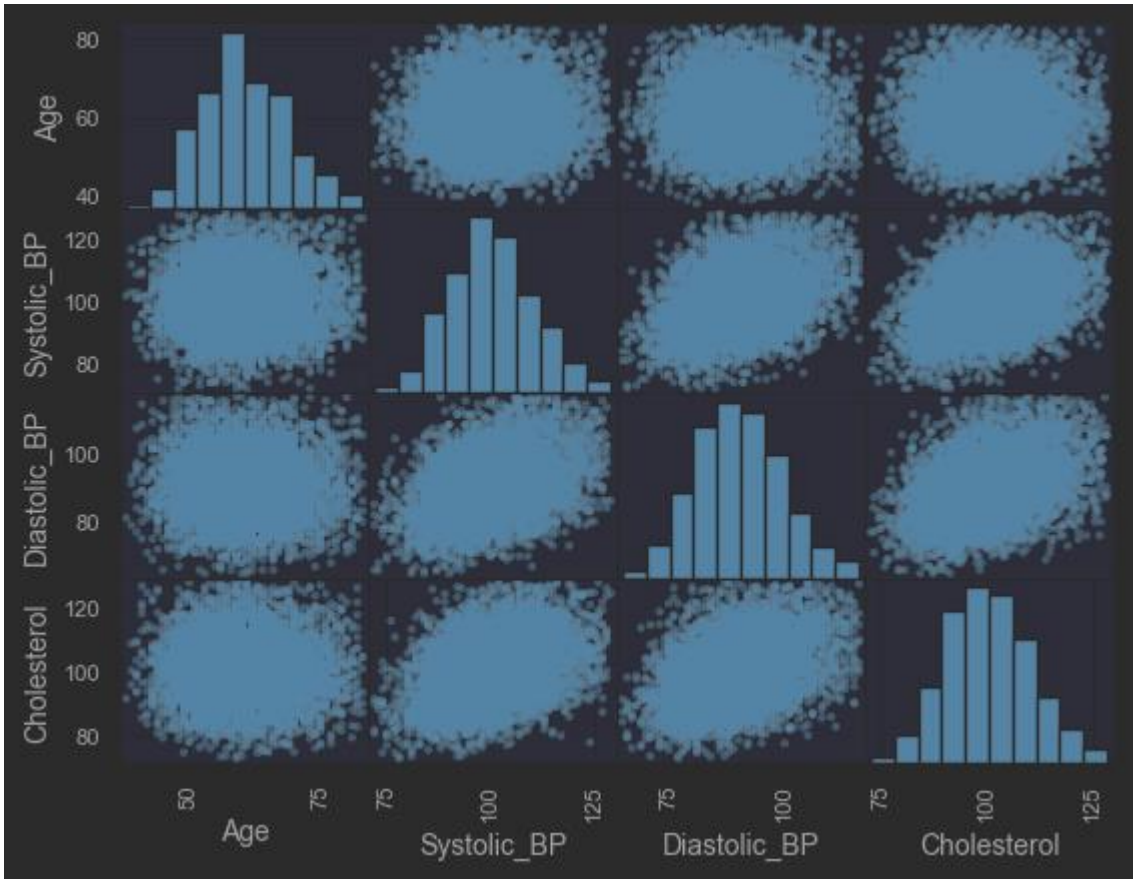
explanation - our goal is to balance the labels, so we can remove some of the example of the larger group, and create synthetically new examples for the smalled group with data augmentation.

ii.

All the model's learning is being done on the training set, if we split the data in an imbalance manner, our model might no perform well on the test set, it might overfit to the training set and will fail to generalize when given the test set.

At the tutorial we overcome this issue by using Y (our labels) as a stratification for the split, meaning we made sure the distribution of labels is being kept at the spite to test and train sets, so both will has the same distribution.


Age
Systolic_BP
Diastolic_BP
Cholesterol
y
0.0
1.0



questions:

i. Was there anything unexpected?

ii. Are there any features that you feel will be particularly important to your model? Explain

why. Base your answer on a valid reference.

answers:

we can see here that high cholesterol, high age and high systolic BP all correlated with positive label.

In addition, most of our features do not have high correlation between them, the only two features which are correlated are systolic and diastolic BP, if 2 or more features were linear correlated we might be able to reduce the number of features and not get inferior results with our model, but this is not the case here.

ii.

According to literature older people tend to a higher risk for diabetes, and having diabetes for longer time increases the risk to have diabetic retinopathy, so we believe age will be particulary important to our model, in addition high cholesterol and high systolic BP are correlated with other eyes condition so they might be important as well.

refs:

https://www.sciencedirect.com/science/article/pii/S0002939403002198

https://europepmc.org/article/med/15078674

creating models , optimizing and tunning the hyperparamater :

Logisticregression :

```
Training set:
ACC = 0.7459
F1 = 0.7445
AUROC = 0.7459

Testing set:
ACC = 0.7159
F1 =  0.7186
AUROC = 0.7159
```

SVM:

```
Training set:
ACC = 0.7437
F1 = 0.7427
AUROC = 0.7438

Testing set:
ACC = 0.7159
F1 =  0.7186
AUROC = 0.7159
```

Random forest:

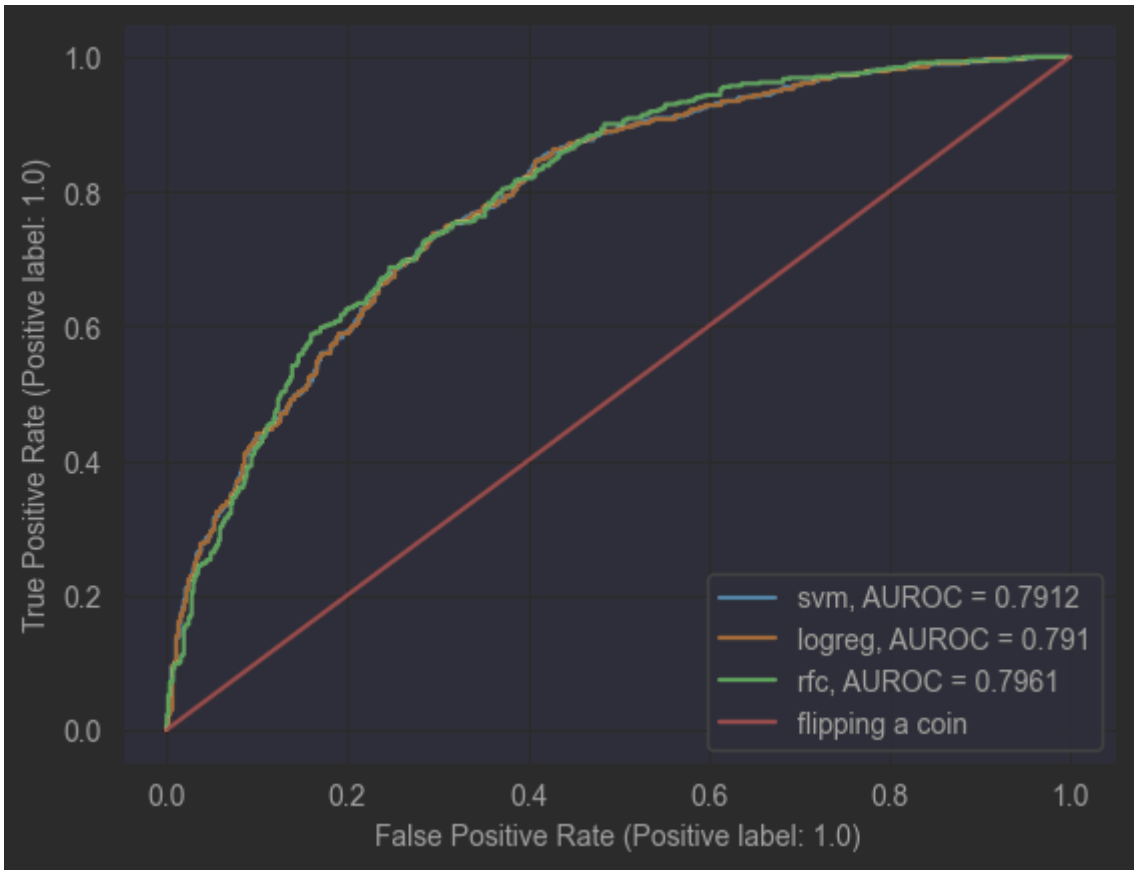
```
Training set:
ACC = 0.7744
F1 = 0.7798
AUROC = 0.7743

Testing set:
ACC = 0.7176
F1 =  0.7277
AUROC = 0.7174
```

naive classifier :

```
In 52
positive = y_test[y_test==1]
# negative = y_test[y_test==0]
print(f'success percentage for naive classifier = {len(positive) / len(y_test) * 100:.3f}%')
```

```
success percentage for naive classifier = 50.259%
```



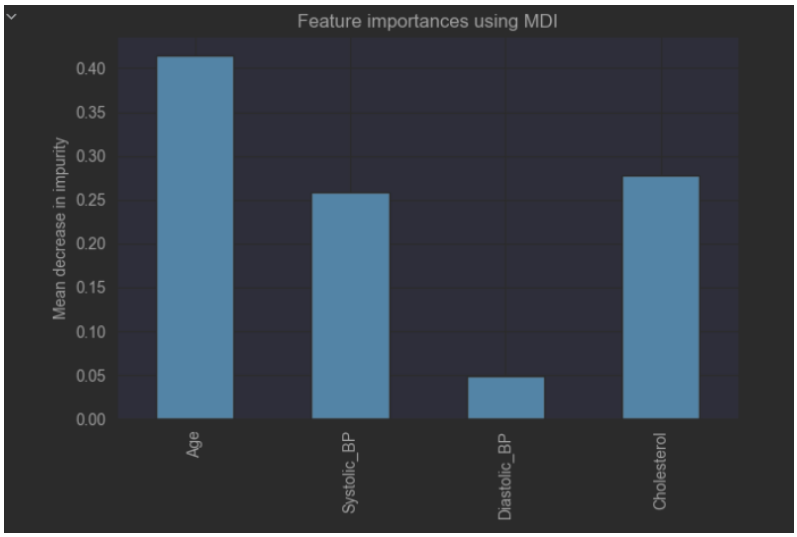
question:

(d) Which performed the best on the testset? Linear or nonlinear models? Compare all of them to a naive classifier. *Notice that the difference in performance between the models can be subtle.
answer:

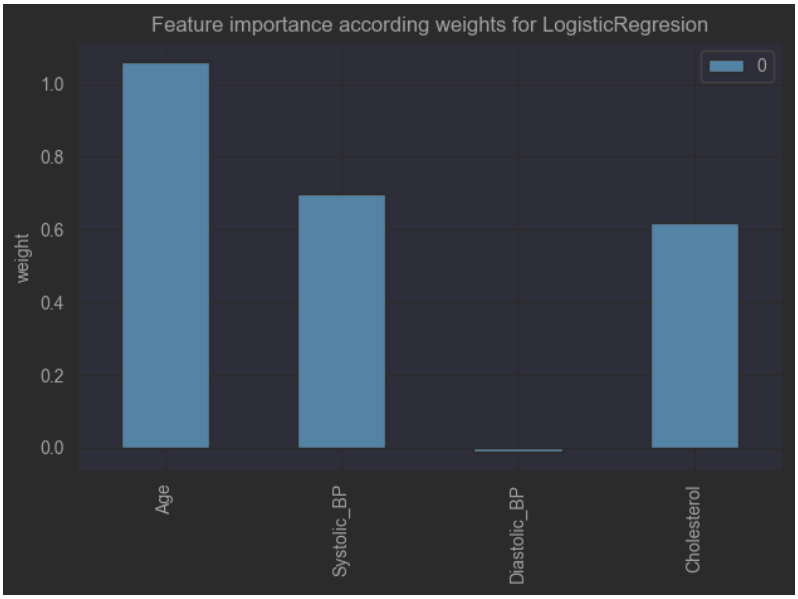
all models performed about the same , all metrics are very similar for the test set at all models .
we can't really chose one model that performed better than the other ,Random forest - got the best AUROC-the metric we tried to maximize,but with 0.005 difference which is very small .
The differnces between the models are very very subtle so if we really had to chose a model we will go with one of the linear models wich are simpler and in this case do not gives an inferior results .
on the bright side ,all models performed better than naive classifier .

feature_importances_

Random forest:



LogisticRegresion:



questions:

i. Report AUC, F1, ACC for both train and test set for the Random Forest model.

ii. Is there a performance difference between train and test? Why? If there is a difference, what can you do to reduce it?

iii. What is the advantage of Random Forest model compared to SVM model?

iv. What are the 2 most important features according to the random forest? What are the two most important according to logistic regression? Does this match up with the feature exploration you did?

i. answer above

ii. Yes, the random forest model performed better on the training set, meaning that our model tend

to overfitting, we can reduce the max depth param to a smaller number, that will reduce the number of branches of the tree and will possibly reduce overfitting.

It is common for there to be a performance difference between the training and test sets, especially if the model is overfitting the training data. This means that the model is performing well on the training data but is not able to generalize well to new, unseen data. This can happen if the model is too complex for the amount of training data, or if the model is not using the right features to make predictions.

To reduce the performance difference between the training and test sets, there are several things you can try:

Collect more training data: This can help the model learn more about the underlying patterns in the data, which can make it more robust when applied to new, unseen data.

Use regularization techniques: These techniques, such as weight decay and dropout, can help prevent the model from overfitting the training data by adding a penalty for large weights or randomly dropping out neurons during training.

Simplify the model: A simpler model is less likely to overfit the training data and may generalize better to new data.

Use cross-validation: This involves dividing the training set into multiple smaller sets, training the model on one set and evaluating it on another, and repeating this process until all sets have been used for both training and evaluation. This can help provide a more accurate estimate of the model's performance on unseen data.

Tune hyperparameters: The hyperparameters of a model, such as the learning rate and the number of hidden units, can have a significant impact on its performance. By tuning these hyperparameters, you can often improve the model's performance on the test set.

iii. Random Forest is suited for multiclass problems, while SVM is intrinsically two-class. Random Forest also works well with a mixture of numerical and categorical features. In addition, SVM doesn't perform well on a large data set because the required training time is higher.
At our results , random forest performed better than SVM, we can see that it scored a bit higher at all metrics both for training and testing sets, it got a higher rate of TP but did worse at the TN.

iv.The 2 most important features for random forest are age and Cholesterol, for Logistic regression Age and Systolic BP. but we can see that Age is by far the most important feature for both model, and that Cholesterol and systolic BP have about the same importance.
this match the exploration we did - according to literature in general, if Age is given as one of the features the model (all models) the age feature will be very important and in most cases will be the most important feature, as happened here for both modals. regarding BP - our dataset contains 6,000 diabetic patients, Diabetes causes damage by scarring the kidneys, which in turn leads to salt

and water retention, which in turn raises blood pressure, in addition Retinopathy is an eye condition that causes changes to the blood vessels in the part of the eye called the ”retina” - those changes to the blood vessels are related with high systolic BP.

Systolic vs diastolic - pressure the heart exerts while beating (systolic pressure). The amount of pressure in the arteries between beats (diastolic pressure) - systolic represents the resistance (elestance) the heart facing when trying to pump the blood , and this is one important as we can at our features importance.