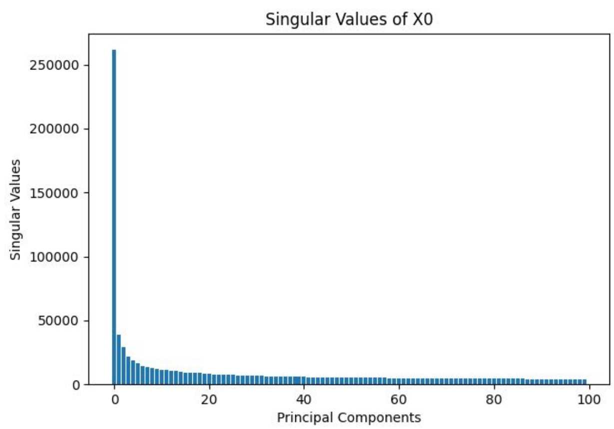
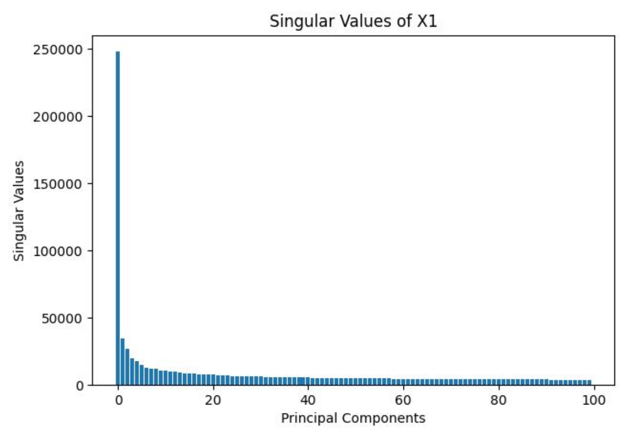
Munib Emre Sevilgen
21602416

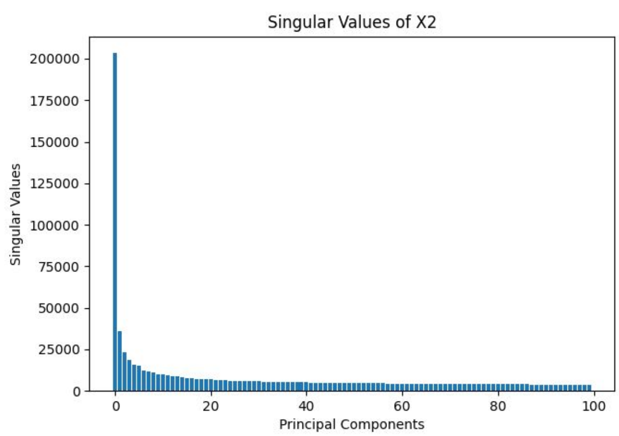
CS464 - 2 / Homework 2
11.12.2020

# 1. PCA on Van Gogh Paintings

## Question 1.1



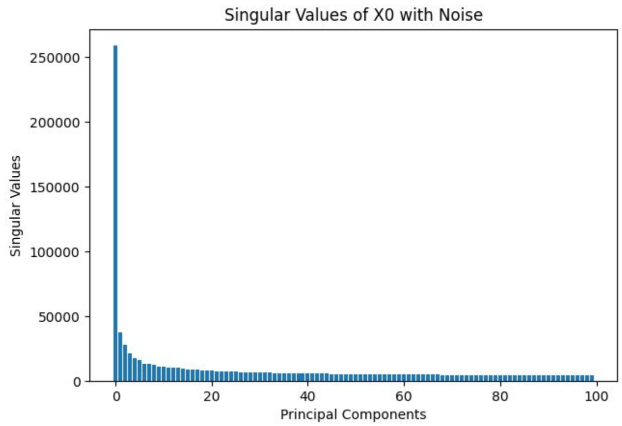



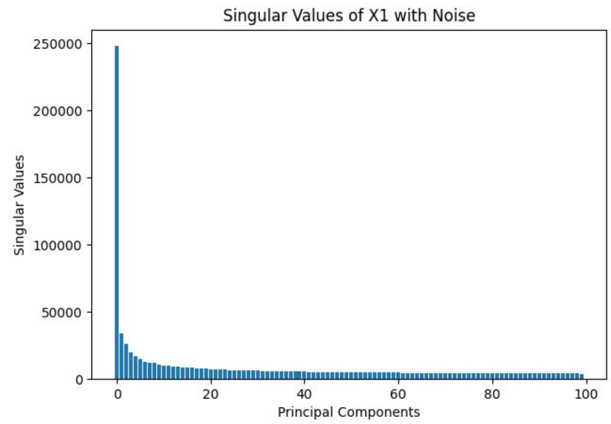



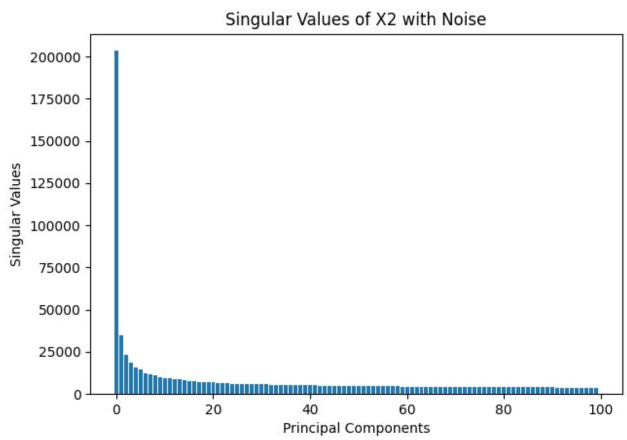

PVE for X0: 0.9571857286629998
PVE for X1: 0.9585480939975898
PVE for X2: 0.9484989613390101

First 10 features can explain the almost 95 percent of the sample which is very good and also it reduces the calculations by ten times.

## Question 1.2




The artificially added noise reduces the overfitting. It increases the chance of the features and prevents the features from dominating. By reverting the SVD calculation the images can be reconstructed. There can be some blur on the image because of the PCA. When we reconstruct the image after PCA, the noise will be reduced because the singular values of the noise are not high, so we don't choose them in PCA.

# 2. Linear Regression on University Admission Records

For the normalization of the features, min-max scaling is used.

## Question 2.1

$J_n = (y - X\beta)^T(y - X\beta) = y^T y - y^T X\beta - \beta^T X^T y + \beta^T X^T X\beta$

$y^T X\beta$ is a scalar. Therefore $y^T X\beta = (y^T X\beta)^T = \beta^T X^T y$

We get $J_n = y^T y - 2\beta^T X^T y + \beta^T X^T X\beta$. When we take a partial derivative with respect to $\beta$ we get $\frac{\partial J_n}{\partial \beta} = \frac{\partial}{\partial \beta} y^T y - 2\frac{\partial}{\partial \beta}\beta^T X^T y + \frac{\partial}{\partial \beta}\beta^T X^T X\beta = 0 - 2X^T y + 2X^T X\beta$

If we set the derivative equal to 0, we get $X^T y - X^T X\beta = 0$ . That is $\beta = (X^T X)^{-1} X^T y$ .
The $X$ matrix is the dataset and the $y$ matrix is the observed outputs in the data.

## Question 2.2

R^2 of model 1: 0.6763875763974101
MSE of model 1: 0.020225014150919457
MAE of model 1: 0.11682271677227223
MAPE of model 1: 0.1772298364219062

R^2 of model 2: 0.7030579464033198
MSE of model 2: 0.015455114335628756
MAE of model 2: 0.10333301546280785
MAPE of model 2: 0.15209587631506538

R^2 of model 3: 0.72542167514538
MSE of model 3: 0.01399138087628686
MAE of model 3: 0.10034541417654777
MAPE of model 3: 0.14677799797769892

R^2 of model 4: 0.7293334611335731
MSE of model 4: 0.012711403566744954
MAE of model 4: 0.09095671520680344
MAPE of model 4: 0.13977484234690501

R^2 of model 5: 0.6904389929831536
MSE of model 5: 0.014919900500106562
MAE of model 5: 0.10091006662015092
MAPE of model 5: 0.1460307216698058

## Question 2.4

- MSE can tolerate the outliers more than the MAE because of the square of differences. Also, because of the same reason, the results can be more similar than the MAE.
- I select the R^2 because it can represent the accuracy of the model independent from the scale of the output and the errors.
- If we do not have much train data, we can't split the data into enough train, validation and test datasets. Therefore we use cross validation instead of a fixed dataset.
- If the dataset size is 50000 then I can split the data to train, validation and test datasets with reasonable dataset sizes. Therefore I use the fixed test dataset approach to reduce the computation time.

# 3. Logistic Regression for Survival Prediction

## Question 3.1

Total time elapsed for Question 3.1: 70.92971086502075 seconds
Accuracy: 0.664804469273743
Precision: 0.5483870967741935
Recall: 0.7391304347826086
NPV: 0.7906976744186046
FPR: 0.38181818181818183
FDR: 0.45161290322580644
F1 Score: 0.6296296296296297
F2 Score: 1.25
Confusion Matrix: [[51, 42], [18, 68]]

The model tends to predict positively because the recall value is very high. However, the F1 score is also high enough to say that the model is good enough.

## Question 3.2

Weights:

```
[[-1.04356094  0.55936115 -0.39442541 -0.05325608 -0.01372612  0.04310723 -0.85528961]]
[[-1.64105457  2.62874017 -1.48908507 -1.18248251 -0.69474882  1.79317329  0.01675055]]
[[-1.57352869  2.63835881 -1.67987777 -1.56631864 -0.87386921  2.76280674  0.03394907]]
[[-1.53680612  2.64024843 -1.77945543 -1.74347397 -0.95562704  3.33311658  0.03718739]]
[[-1.51467304  2.64094996 -1.83865833 -1.84062079 -1.00560468  3.69082885  0.03764275]]
[[-1.50059379  2.64133266 -1.87615137 -1.89899833 -1.03843361  3.92374353  0.03746027]]
[[-1.49135945  2.64157591 -1.90071342 -1.93603477 -1.0604449   4.07875493  0.03716971]]
[[-1.48518919  2.64173909 -1.917124   -1.96032033 -1.07533862  4.18331697  0.03690828]]
[[-1.48101734  2.64185073 -1.92822213 -1.97656942 -1.08547623  4.25445842  0.03670357]]
[[-1.4781748   2.6419277  -1.93578612 -1.98757723 -1.09240728  4.30313593  0.03655201]]
```

Total time elapsed for Question 3.2: 0.31299638748168945 seconds
Accuracy: 0.659217877094972
Precision: 0.5434782608695652
Recall: 0.7246376811594203
NPV: 0.7816091954022989
FPR: 0.38181818181818183
FDR: 0.45652173913043476
F1 Score: 0.6211180124223603
F2 Score: 1.25
Confusion Matrix: [[50, 42], [19, 68]]

The model tends to predict positively because the recall value is very high. However, the F1 score is also high enough to say that the model is good enough.

### Question 3.3

I have normalized the features, therefore it can be said that the difference from 0 shows the importance of the feature. If I don't normalize the features, then the weights would be very far away from each other and some features can dominate the predictions. When we look at the weights of the logistic regression, there is not any feature that dominates the predictions. The most important features are Gender and Fare. Most important categorical feature is Gender and continuous feature is Fare. When we increase the batch_size we reduce the number of computations of the weights. Therefore the training time decreases when we increase the batch size.

## 4. SVM

### Question 4.1

Score of test set 0: 0.716
Score of test set 1: 0.7
Score of test set 2: 0.744
Score of test set 3: 0.728
Score of test set 4: 0.74
For C in 1e-06, 0.0001, 0.01, 1, 10.0, 10000000000.0

Scores are the accuracy values. There is no difference between the results when the C hyperparameter is changed.

### Question 4.2

Score of test set 0: 0.744
Score of test set 1: 0.804
Score of test set 2: 0.76
Score of test set 3: 0.78
Score of test set 4: 0.788
For gamma in 0.0625, 0.25, 1, 4, 1024, scale
For C in 0.0001, 0.01, 1, 10.0, 10000000000.0

Scores are the accuracy values. There is no difference between the results when the C and gamma hyperparameters are changed.

### Question 4.3

The performance of the models doesn't change with different C and gamma hyperparameter values. If SVM was trained directly on image pixels, the performance can increase.