FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Marin Stoyanov	Bulgaria	azonealerts@gmx.com	
Guoying Li	United States	liguoying1019@gmail.com	
Ashutosh Kumar	India	skdubey.ashutosh@gmail.com	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).

Team member 1	Marin Stoyanov
Team member 2	**Guoying Li**
Team member 3	**Ashutosh Kumar**

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed.
Note: You may be required to provide proof of your outreach to non-contributing members upon request.

N/A

Principal components

● **Advantages:**

Dimensionality Reduction: PCA reduces the number of features, simplifying the dataset.

Noise Reduction: It helps in filtering out noise and focusing on the most important features.

Visual Representation: Data can be visualized in a reduced-dimensional space, aiding interpretation.

● **Basics:**

Principal components are new variables which are the linear combinations of the original variables. It's also uncorrelated and most of the information is compressed into the first components.

Principal components are orthogonal.

● **Computation:**

In the Jupyter Notebook, we show how the load_digits dataset look like by presenting one of the digits images, normalizing the data, plot the correlation matrix and extract features by running PCA() function in two methods: one is number of components can explain 90% of variance (Reduced number of components is 30) and the other is the number of components equal to 40. We select 40 as the optimal number of components by running GridSearchCV pipeline. Finally, we apply SVM to make predictions and evaluate with performance metrics, including precision, recall and F1 Score.

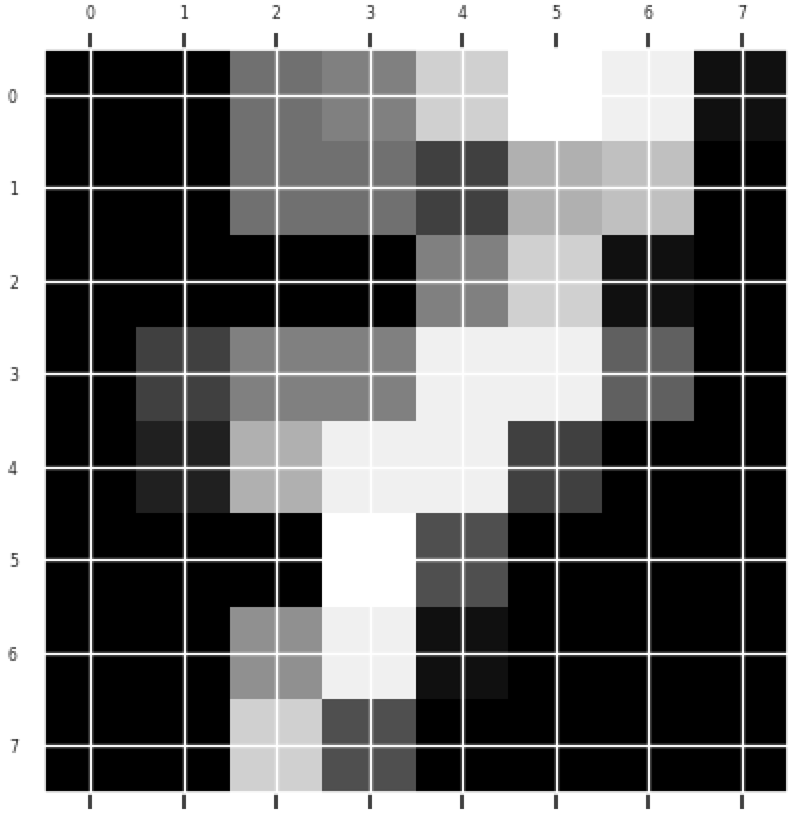


Fig.1. Random example of a image of the digit 7

From the correlation plot below, we can observe highly correlated between variables.

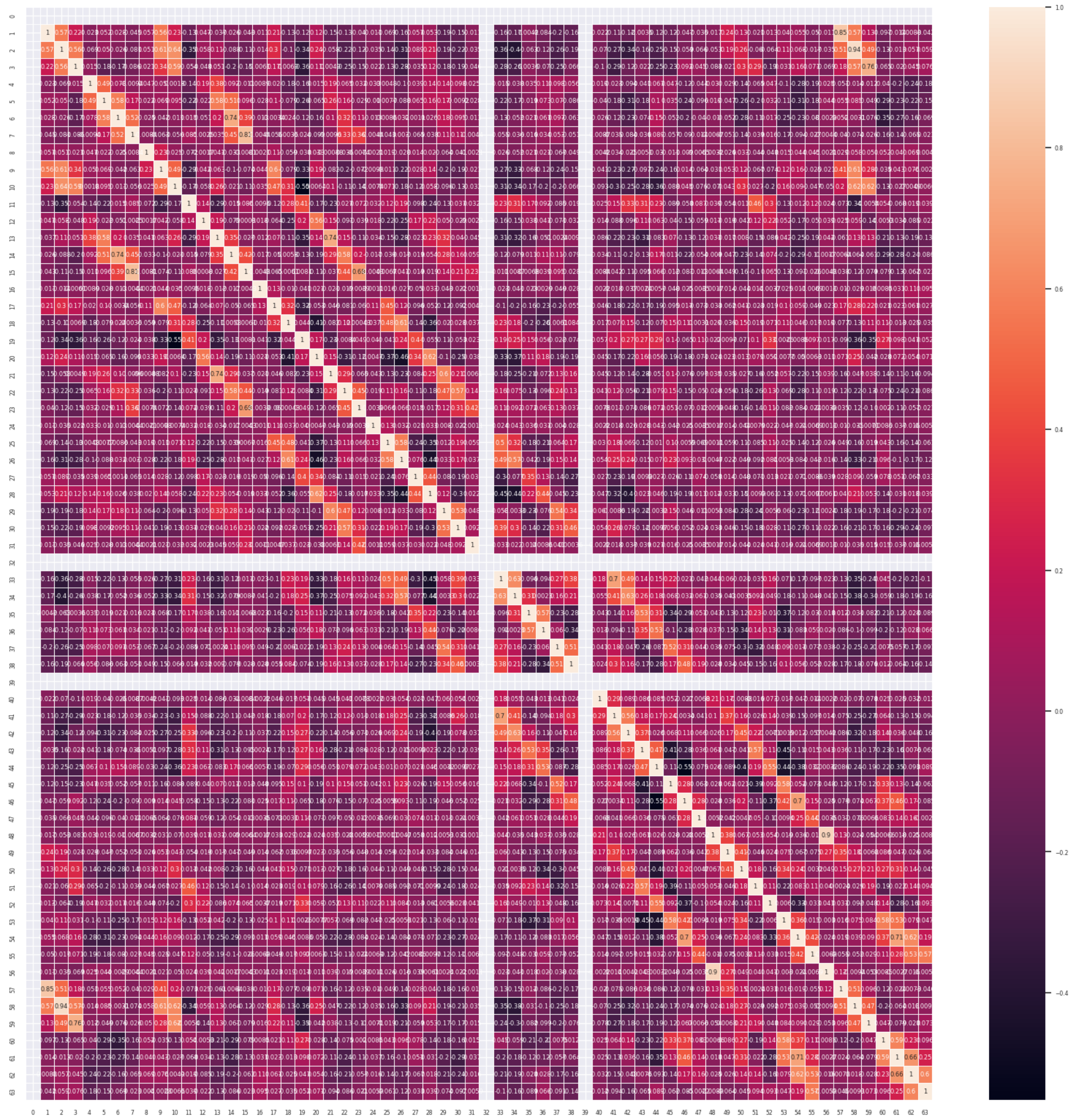


Fig.2. correlation matrix before PCA

By Grid Search, we found the optimal parameters for PCA and SVM as follows:
est parameter (CV score=0.974):
{'pca__n_components': 40, 'svm__C': 100, 'svm__gamma': 0.001, 'svm__kernel': 'rbf'}

The plots [8] below can also show the optimal number of components at 40.

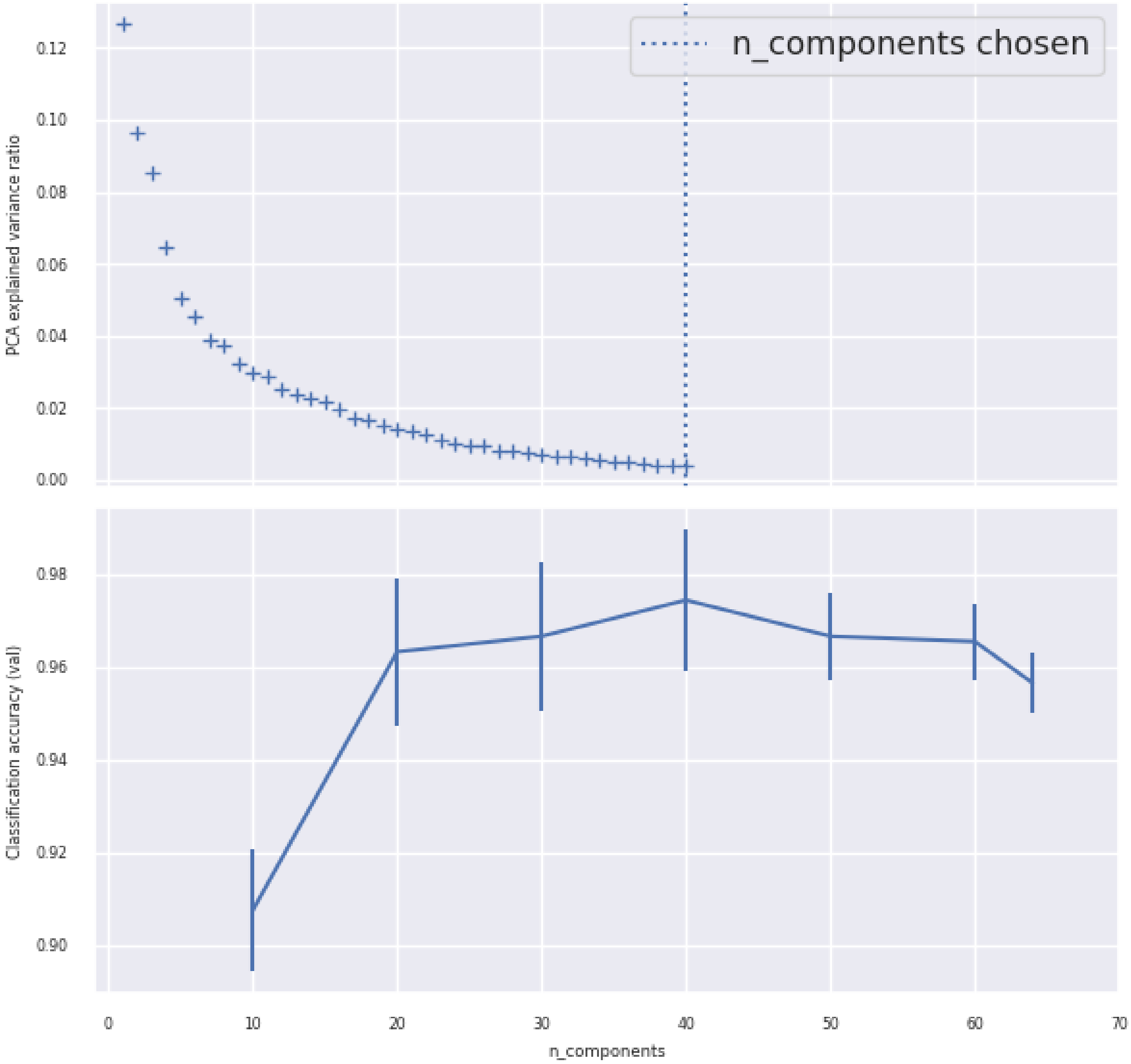


Fig.3. PCA plot and the best classifier results for each number of components

● **Disadvantages:**

Linearity Assumption: PCA assumes a linear relationship between variables, which might not always hold.
Information Loss: While reducing dimensions, some information is inevitably lost.
Interpretability: Principal components might not have a clear, interpretable meaning in the original feature space. In other words, we won't be able to interpret which variables are the top predictors. To know the top predictors, we will need to build the model without PCA.

● **Equations:**

Step 1: standardization: transform all the variables to the same scale

$$z = \frac{value - mean}{standard\ deviation}$$

Step 2: compute covariance matrix

$$Cov(x, x) = Var(x)$$

$$Cov(x, y) = Cov(y, x)$$

Step 3: find the eigenvalues and eigenvectors

If A is a matrix then the eigenvalues of A are the solution of the following characteristic equation:

$determinant(A - \lambda\mathbf{I}) = |(A - \lambda\mathbf{I})| = 0$

Where:

$\mathbf{I}\ is\ an\ n{\times}n\ identity\ matrix.$

$If\ \lambda\ is\ an\ eigenvalue\ of\ A,\ then\ there\ exists\ a\ vector\ \overline{x}\ such\ that$:

$A\overline{x} = \lambda\,\overline{x}$

Then $\overline{x}$ is the eigenvector of matrix A and λ is the eigenvalue.

For example if A is 2 x 2 then:

$A\overline{x_1} = \lambda_1\,\overline{x_1}$

$A\overline{x_2} = \lambda_2\,\overline{x_2}$

And all this can be expressed like this as well:

$$A[\overline{x_1}, \overline{x_2}] = [\overline{x_1}, \overline{x_2}] \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Where

$\Phi = [\overline{x_1}, \overline{x_2}]$ and the matrix with eigenvalues can be defined as: Λ

And this leads to the following equation:

$A\Phi = \Phi\Lambda$

$Normalizing\ the\ eigenvectors\ such\ that\ they\ are\ orthogonal\ we\ have$:

$\Phi\Phi^T = \Phi^T\Phi = \mathbf{I}$

$which\ implies$: $\Phi^T A\Phi = \Lambda \quad => \quad A = \Phi\Lambda\Phi^T$

Step 4: feature vector: compute the eigenvectors and sort them by their eigenvalues in descending order. In this way, we can find the principal components in order of significance. By deciding whether to keep all the components or discard those of low eigenvalues, we can form the feature vectors with the remaining components.

● **Features:**

- Missing Values: PCA handles missing values well, making it suitable for datasets with incomplete information.
- Multicollinearity: It is effective in dealing with multicollinearity, as principal components are orthogonal.

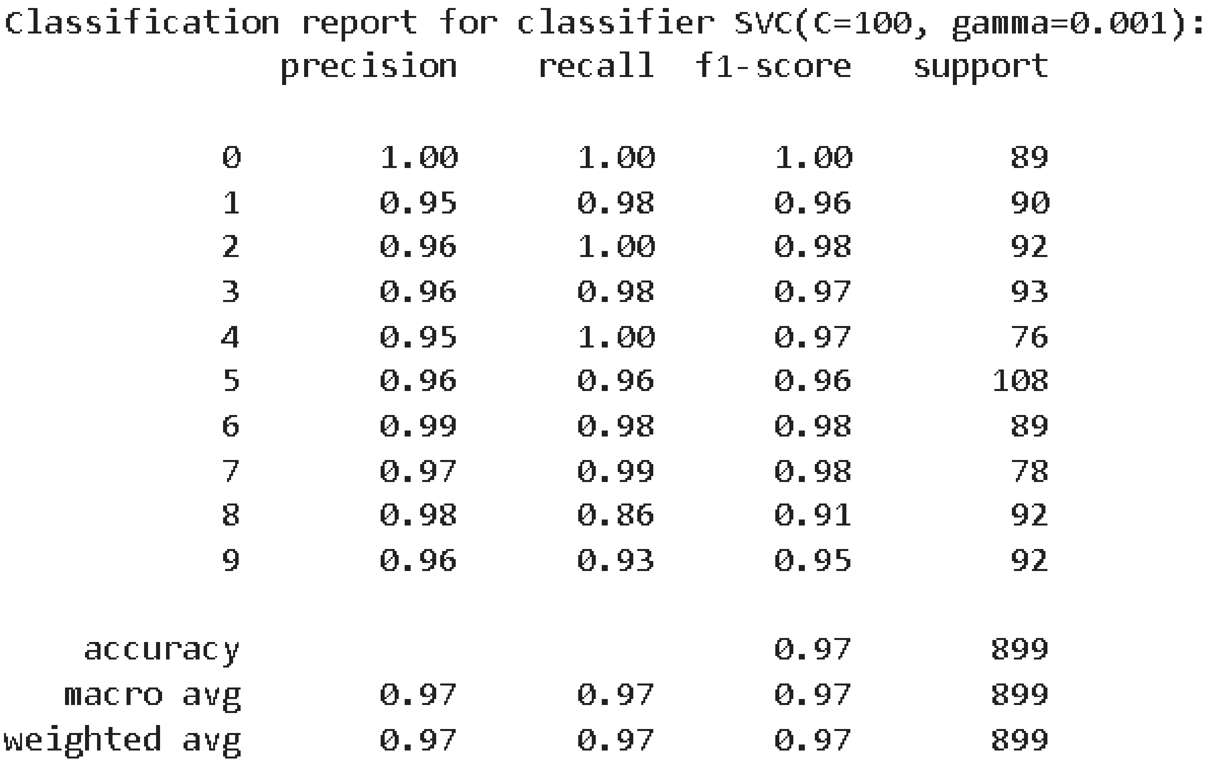
● **Guide:**

- Inputs: High-dimensional dataset
- Outputs: Lower-dimensional representation of the data (principal components)

● **Hyperparameters:**

- Number of Components: The number of principal components to retain, which influences the amount of variance preserved.

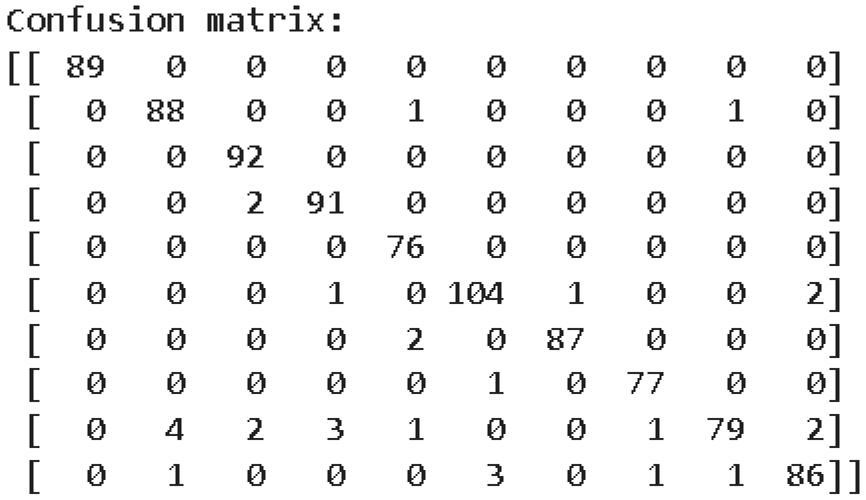
● **Illustration: Visualization**



```
Classification report for classifier SVC(C=100, gamma=0.001):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        89
           1       0.95      0.98      0.96        90
           2       0.96      1.00      0.98        92
           3       0.96      0.98      0.97        93
           4       0.95      1.00      0.97        76
           5       0.96      0.96      0.96       108
           6       0.99      0.98      0.98        89
           7       0.97      0.99      0.98        78
           8       0.98      0.86      0.91        92
           9       0.96      0.93      0.95        92

    accuracy                           0.97       899
   macro avg       0.97      0.97      0.97       899
weighted avg       0.97      0.97      0.97       899
```



```
Confusion matrix:
[[ 89   0   0   0   0   0   0   0   0   0]
 [  0  88   0   0   1   0   0   0   1   0]
 [  0   0  92   0   0   0   0   0   0   0]
 [  0   0   2  91   0   0   0   0   0   0]
 [  0   0   0   0  76   0   0   0   0   0]
 [  0   0   0   1   0 104   1   0   0   2]
 [  0   0   0   0   2   0  87   0   0   0]
 [  0   0   0   0   0   1   0  77   0   0]
 [  0   4   2   3   1   0   0   1  79   2]
 [  0   1   0   0   0   3   0   1   1  86]]
```

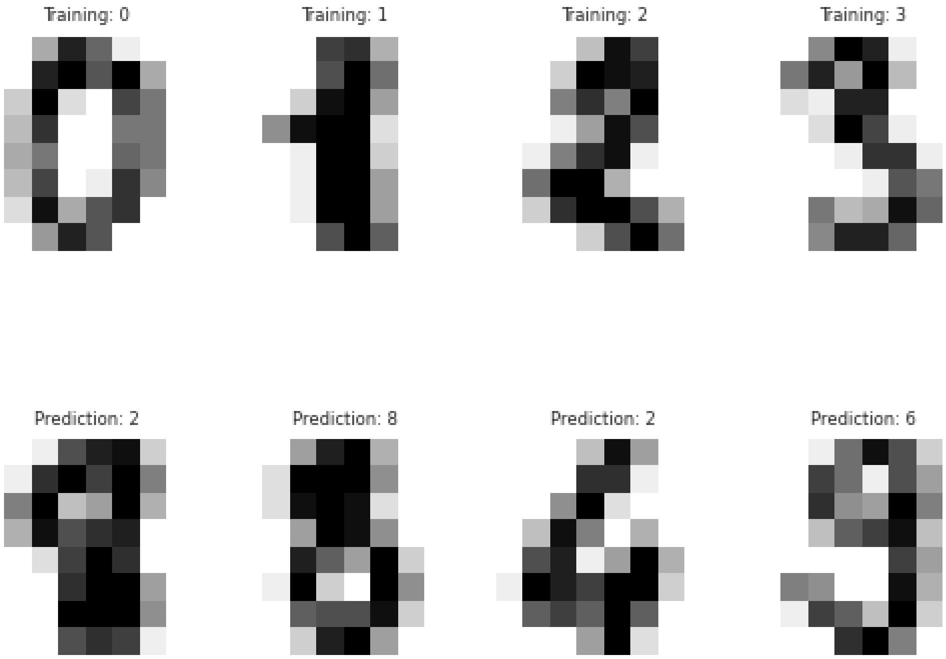


Fig.4. Training vs Prediction

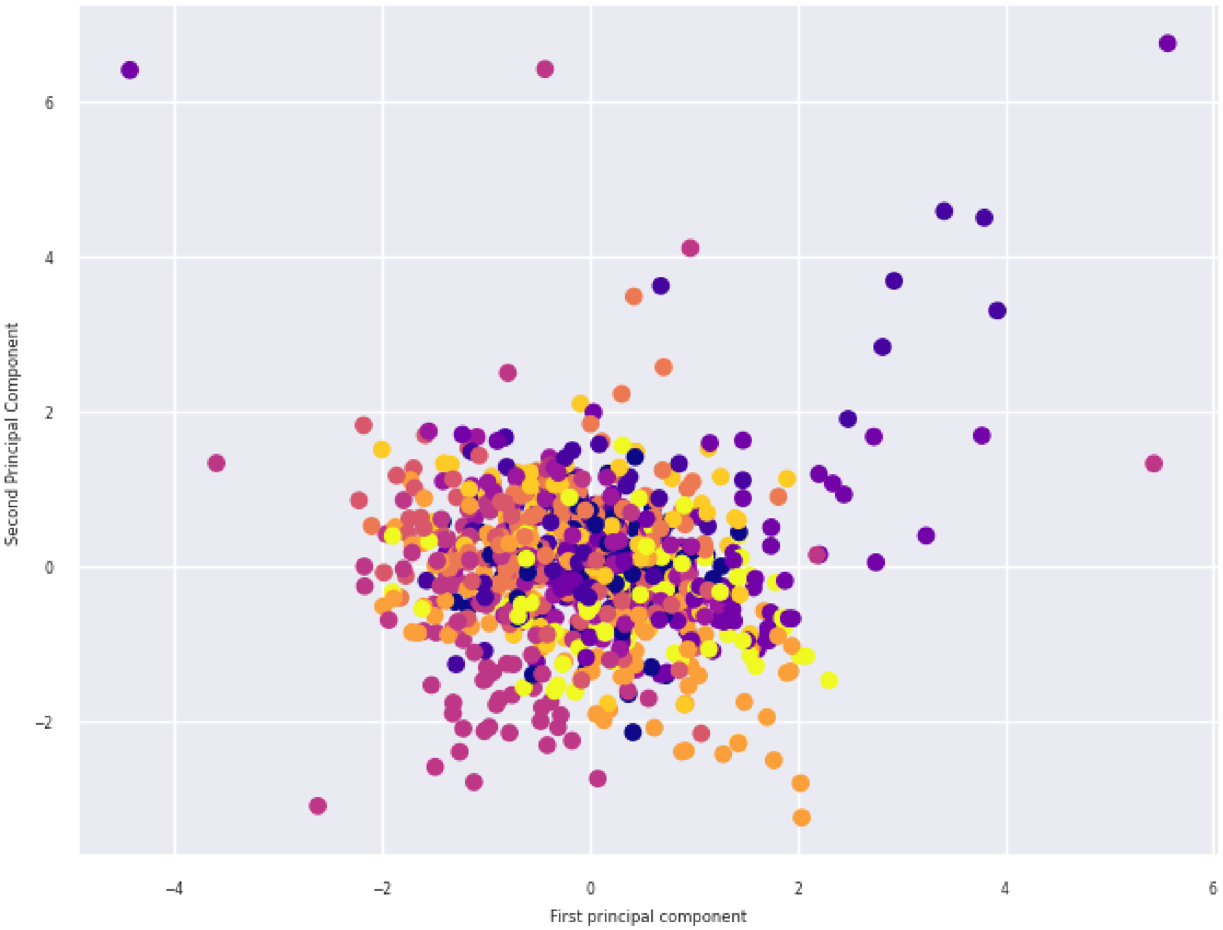


Fig.5. Principal components

- **Journal:**

Avellaneda, Marco and Lee, Jeong-Hyun, Statistical Arbitrage in the U.S. Equities Market (July 11, 2008). Available at SSRN: https://ssrn.com/abstract=1153505 or http://dx.doi.org/10.2139/ssrn.1153505

- **Keywords:**

Dimensionality Reduction, Eigenvectors, Eigenvalues, Multicollinearity, Feature Extraction, Data Visualization, Machine learning, risk model, clustering, k-means, statistical risk models, covariance, correlation, variance, cluster number, risk factor, optimization, regression, mean-reversion, factor loadings, industry classification, quant, trading, dollar-neutral, alpha, signal, backtest;alpha, optimization, regression, risk factor, factor model, style factor, volatility, turnover, momentum, correlation, covariance, variance, equities, Sharpe ratio; Futures term structure, Roll yield, Convenience yield, Contango, Backwardation, Commodity trading strategy;

expected returns, implied volatility, realized volatility, volatility spread; interest rate, bond, risk management, factor model; Performance, Attribution, Fixed Income, Central Bank

LASSO

● **Advantages:** it can do feature selection as well as regularization and thus enhance the accuracy of the prediction as well as the interpretability of the given model. Given these features this model is very useful for higher dimensional data and especially when the features are more than the observations. Thanks to its penalty function this model lowers the coefficients of the not important features to zero and thus lowers the variables count so in practice this is how a feature reduction and selection can be conducted [2].

● **Basics:** Lasso is a model from the so-called penalty regression type which is part of the linear

regression family.

● **Disadvantages:** The main problems here are that it cannot deal with multicollinearity and since it is shrinking the coefficient to zero, there is the fact that some of the data is being lost. Also Lasso is very sensitive to noisy data, outliers and missing data.

● **Equations:**

The penalty that it applies toward the loss function looks like this [1]:

$$Lasso = RSS + \lambda \sum_{i=1}^{m} |\beta_i|$$

Where:

$\lambda\ is\ the\ regularization\ parameter$

RSS is the Residual Sum of Squares from the linear regression [3] and its formula looks like this:

$$RSS = \frac{\sum_{i=o}^{n} (y_i - \widehat{y_i})^2}{n}$$

Where:

$n\ is\ the\ total\ number\ of\ observations$

$y_i\ is\ the\ i^{th}\ observed\ value.$

$\widehat{y}_i$ *is the corresponding predicted value*

- **Features:** Lasso can do feature selection thanks to its penalty term which is derived from the L1 norm of the regression coefficients. This norm is simply the sum of the absolute values of the coefficients. The effect of this penalty term is to push the estimated coefficients towards zero, which leads to simpler models with fewer non-zero coefficients. This property of LASSO encourages the creation of models that are easier to interpret and less likely to overfit the data, as they rely on a smaller subset of the available features. This is particularly beneficial when dealing with high-dimensional datasets, where the risk of overfitting is high and model interpretability is crucial. [4] Lasso is applied for fighting overfitting thanks to its regularization feature and also is suitable for working with high dimensional data.

- **Guide:** Usually the input is a high dimensional data and the output is feature importance metrics and thus feature selection and extraction capability for the end user or the data scientist that is applying this methodology [5].

- **Hyperparameters:** the most important hyperparameter is λ which is responsible for the regularization. The value of λ must be a positive float number or zero. When $\lambda = 0$ then Lasso turns into ordinary least squares method used in the linear regression. [6]

- **Illustration:**

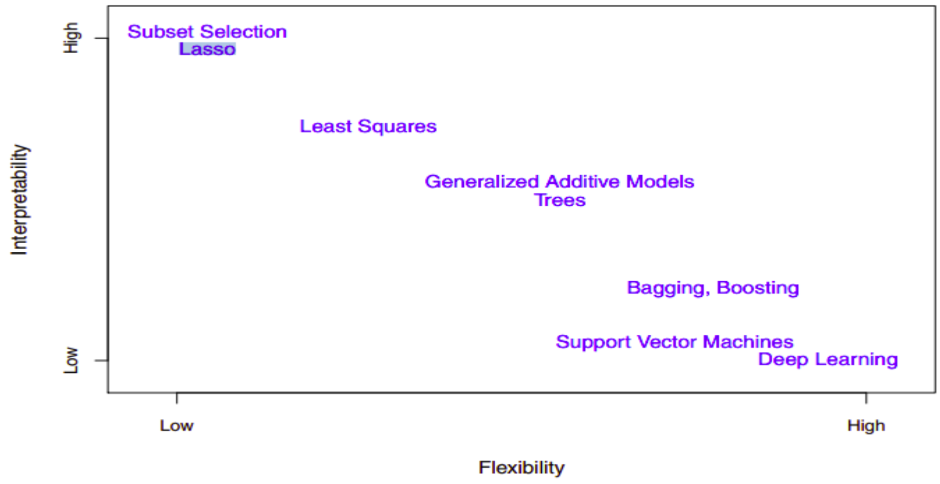


Fig.6. Trade-off between interpretability and flexibility of different machine learning methods [4]

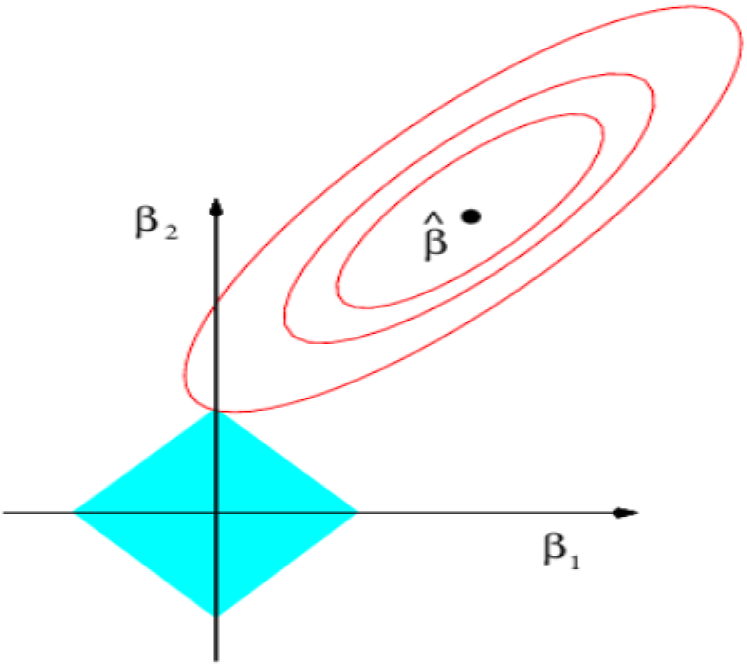


Fig.7. Constraint regions and contours of RSS [4]

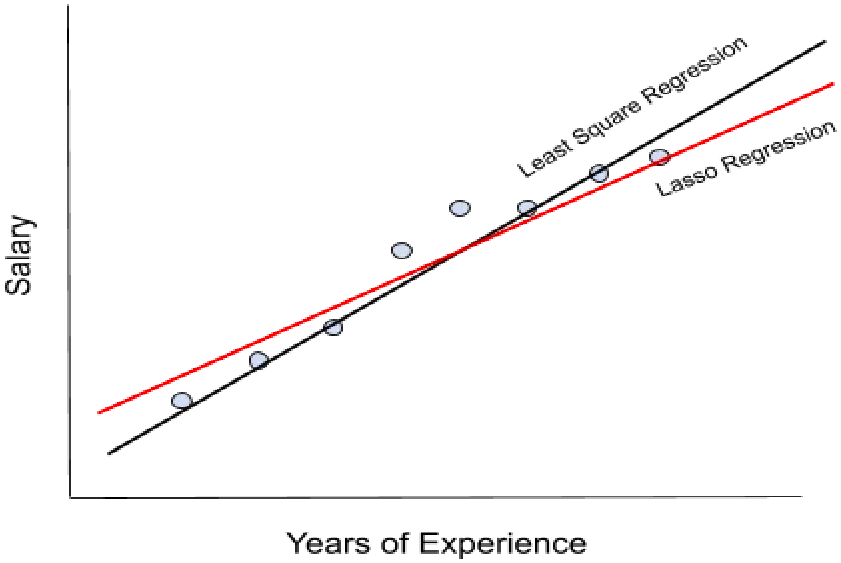


Fig.8. Example of the Lasso effect over Least Square Regression

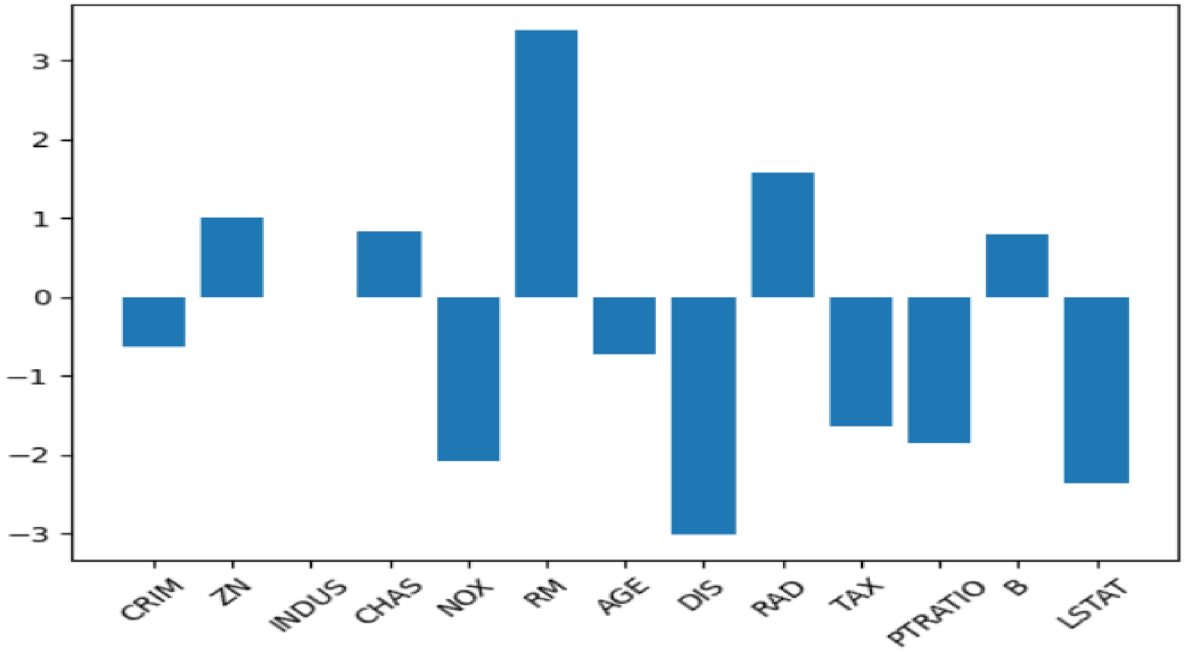


Fig. 9. Lasso regression BEFORE hyperparameter tuning

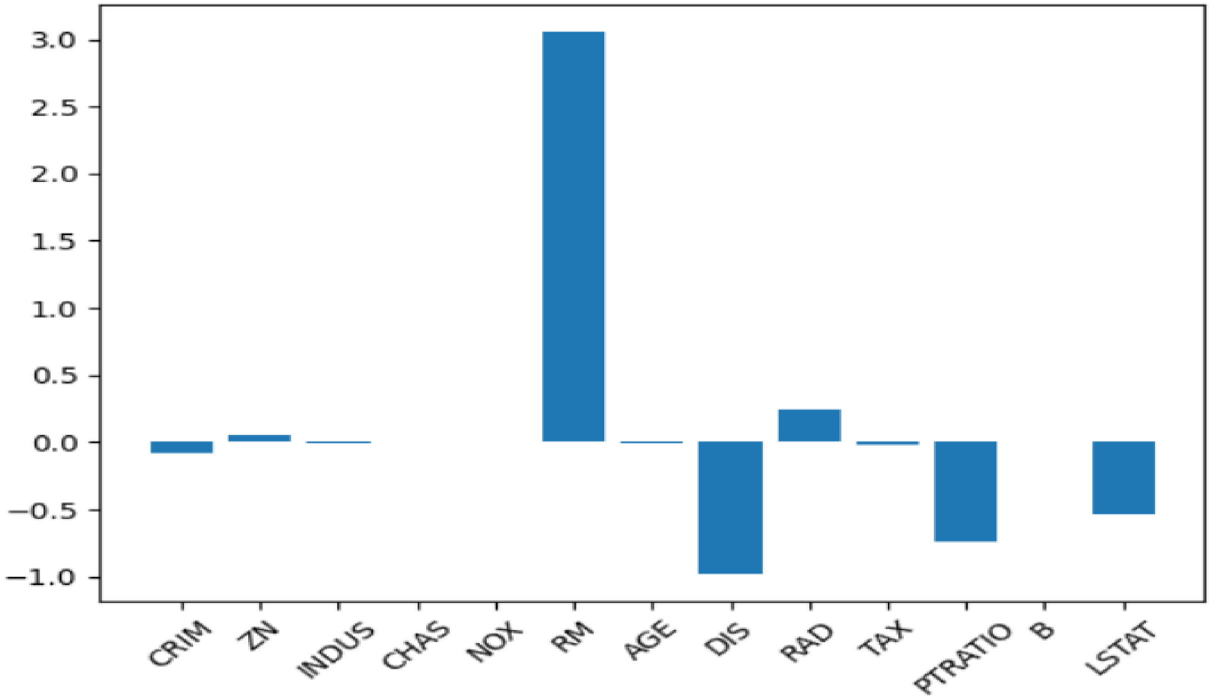


Fig.10. Lasso regression AFTER hyperparameter tuning

● **Journal:** “*Lasso Regressions and Forecasting Models in Applied Stress Testing*”

by Jorge A. Chan − Lau, published by the International Monetary Fund.

● **Keywords**:
LASSO, Regression, Regularization, Feature Selection, High − Dimensional Data,

Overfitting, Model Selection, Machine Learning, Finance, Stress Testing

K-MEANS CLUSTERING:

Clustering is a method of unsupervised learning which organizes the data points into a number of groups based on the similarities in attributes of data points. K-means clustering is an example of clustering which is based on centroid-based algorithms or distance-based algorithms and the number of clusters are already defined or fixed**. [14]**

● **Advantages:**

- It’s very easy to implement and understand.
- It can be easily adapted to new examples
- Re-computation of centroid is an advantage for K-means clustering

● **Computation:**

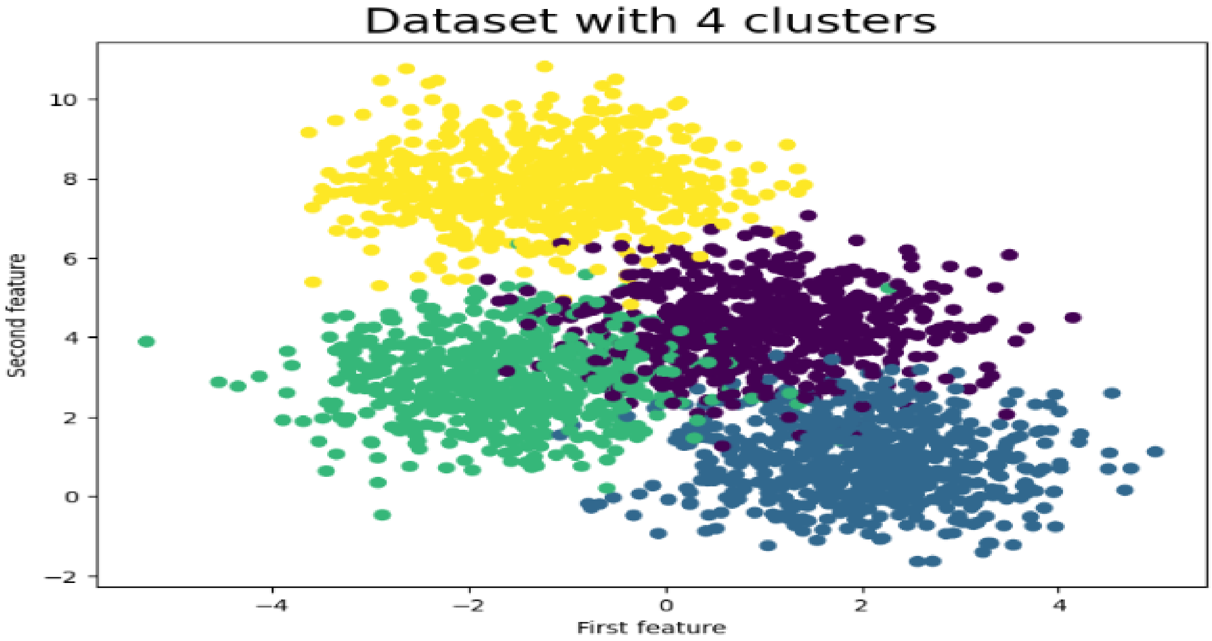


Fig.11. Visualization of 4 clusters

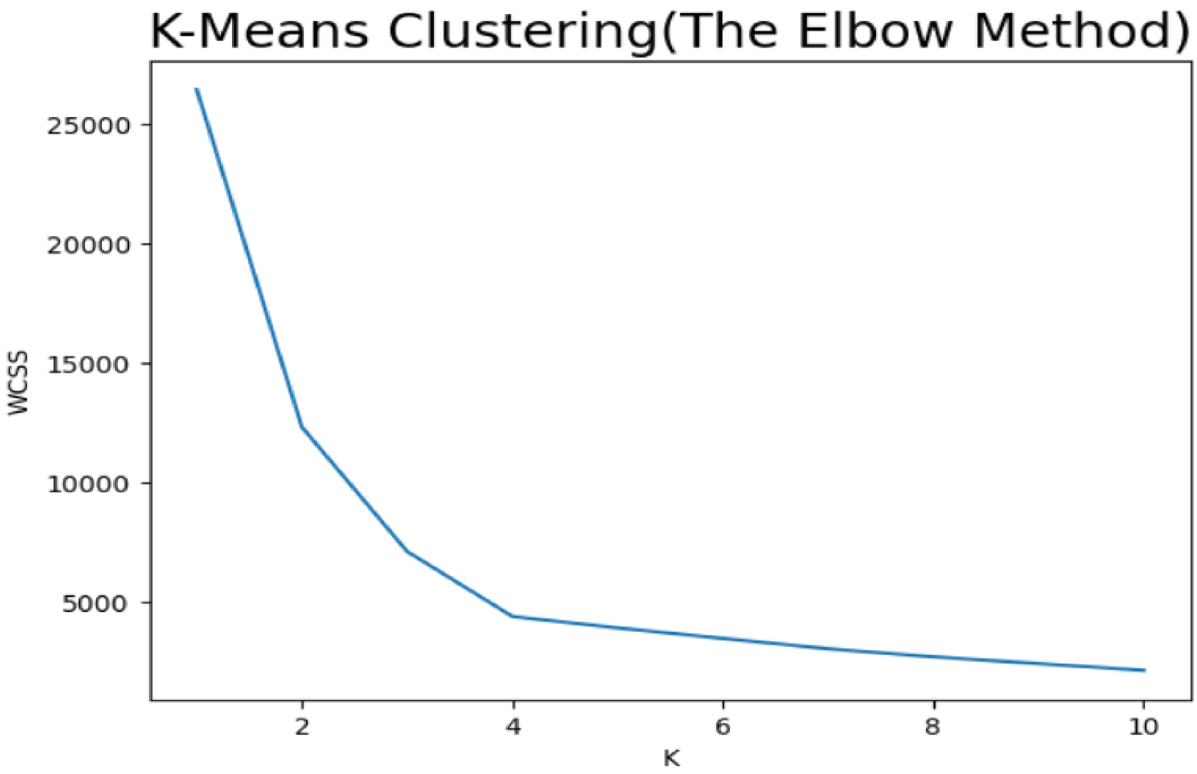


Fig.12. The elbow method

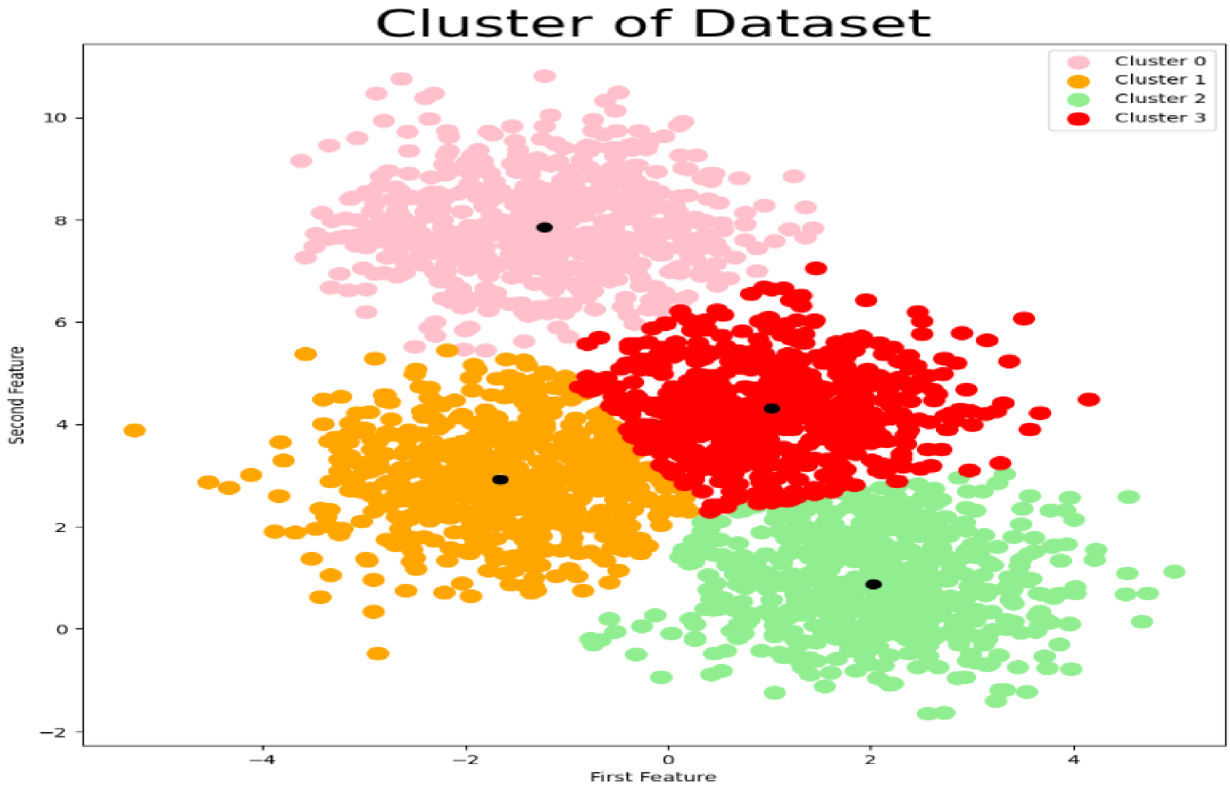


Fig.13. Clustering results

- **Disadvantages:**
  - Selection of the number of clusters before modeling and output is strongly impacted by this number.
  - K-means only works with linear cluster boundaries.
  - It's very sensitive to rescaling which means if we normalize or standardize our data then the output might be completely changed.
  - It doesn't work well for big data and if clusters are having complex geometrical shapes.

- **Equations:**

Step 1:

Selecting K centroids where K is the number of clusters. Let *X*=$\{x_1, x_2, \ldots, x_n\}$ be our data points and *V*=$\{v_1, v_2, \ldots, v_n\}$ be the centroids.

Step 2:

Every centroid is randomly selected and then the distance between all of them and the data points is calculated

Step 3:

Then assign the data points to the centroids closest to them. And after that recalculate the new centroids by computing the mean of the grouping.

$$v_i = \frac{1}{c_i}\sum_{j=1}^{c_i} x_j$$

$where\ c_i\ is\ the\ number\ of\ data\ points\ in\ the\ ith\ cluster.$

Step 4:

We repeat and go back to step 2 if no change is found in the clusters' composition. Otherwise we get the distance of each data point and the new centroid

Mathematically, the objective of the K-means algorithm is to minimize an objective function which in our case is a squared error function given by

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^j - C_j||^2$$

Where

$$||x_i^j - C_j||^2$$

is the Euclidean distance between data points *xi* and the centroid *Cj*

- **Features:**
  - It works well with various types of attributes
  - It can deal with noise and outliers
  - It can handle high dimensionality

- **Guide:**

  Inputs- Dataset as a collection of data points of different attributes.

  Output- Assignment of each data point to one of the K clusters and then gives the value of centroid of clusters.

- **Hyperparameters:**

  The number of clusters- this is fixed before modelling, and it can be decided with the help of different methods like Silhouette analysis or elbow method. [15]

  Number of Iterations- It determines how many times the assignment or steps followed in the algorithm to be done and it helps the algorithm time.

- **Illustration:**

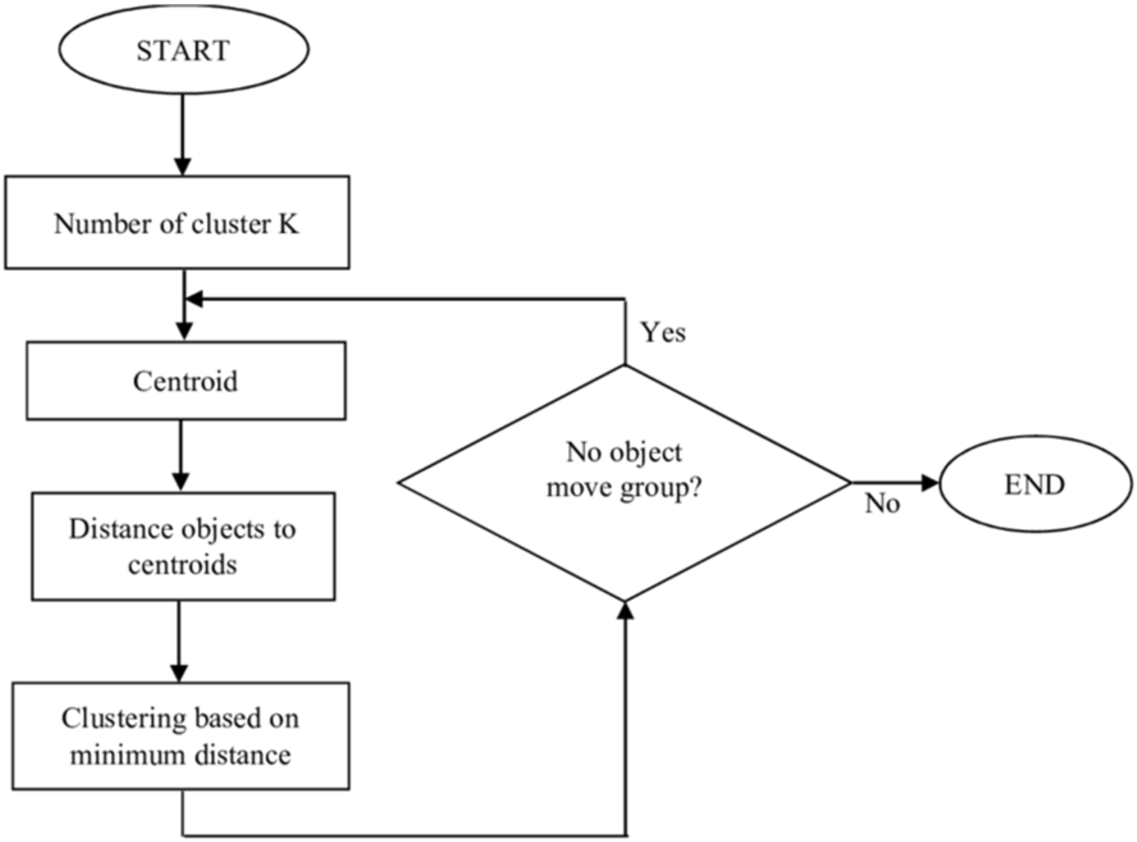

START
Number of cluster K
Centroid
Distance objects to centroids
Clustering based on minimum distance
No object move group?
Yes
No
END

Fig. 14. Algorithm's logical block schema

- **Journal:**

1. Article by Priya Pedamkar - https://www.educba.com/k-means-clustering-algorithm/

2. Combination of K-means clustering with Genetic Algorithm: A review by Diyar Qader Zeebaree, Habibollah Haron, Adnan Mohsin Abdulazeez and Subhi R. M. Zeebaree. Available at- International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 24 (2017) pp. 14238-14245 https://www.academia.edu/download/55576223/Combination_of_K-means_clustering_with_Genetic_Algorithm_A_review.pdf

- **Keywords:**

Clusters, K-means algorithm, Centroids, Centroid-based algorithms, Distance metrics, Euclidean distance, Elbow method, Silhouette analysis, Random initialization, Noise, Outliers, Attributes, Dimensions, Dimensionality, Normalization, Standardization, Scalability, Data analysis, Dataset, Data points, optimization

Technical Section

Tuning Lasso's Hyperparamether:
For tuning the hyperparameter (which can be called either lambda or alpha) we chose to use GridSearchCV methodology combined with the numpy linspace method for dividing the linear space between 0.01 to 1 by 100 equally spaced values and then trying each and every single one of them for fitting the Lasso regression with. This is how feature selection and reduction can be applied to the dataset, because the fitted model has a property called: best_params_ which shows which are the best parameters and thus we know where to concentrate.

Tuning K-Means Hyperparameter:
Tuning in K-means clustering is done with the finding of K value that is the number of optimal clusters. There is no direct method to find the optimal value of K so we do the tuning with the help of various methods. Here, we did the tuning or found the optimal value of K by using Silhouette analysis and Elbow method which help us to find the optimal value of K. In this given project we got the optimal value of K as 4.

Support Vector Machines (SVM) with Principal Component Analysis (PCA)

1. SVM Hyperparameters:

- C (Regularization parameter): The C parameter trades off correct classification of training examples against maximization of the decision function's margin. We compare the list of Regularization parameters: [1, 10, 100, 1000]. By Grid Search, we find 100 as the optimal regularization parameter.

- Kernel Parameters: from the example below, we use a kernel (e.g., radial basis function - RBF), thus we need to tune kernel-specific parameters (e.g. gamma). We compare the list of Kernel parameters: [0.001, 0.0001]. By Grid Search, we find 0.001 as the optimal kernel parameter.

- Kernel Choice: there are various choices, such as linear, polynomial, and RBF. We compare the list of Kernel methods: ['linear', 'rbf']. By Grid Search, we find RBF as the optimal kernel choice.

2. PCA Hyperparameters:

- Number of Components: we can observe from the plot of cumulative explained variance and by Grid Search in the list of number of components ([10, 20, 30, 40, 50, 60, 64]), we decide 40 as the optimal number of components.

By using Grid search, we can find the optimal hyperparameter

Furthermore, with the optimal hyperparameters above, we have performance metrics, such as precision, recall, and F1-score, which are at 97% on average. In addition, further advanced techniques are Bayesian Optimization and Ensemble Methods.

Marketing Alpha

Lasso

The Lasso regression is used mostly for feature selection and dataset reduction because of its penalty function that lowers the coefficients to zero and thus shows which are the not so important features that could be dropped.It is very useful to apply hyperparameter tuning so that we could see what is the best value suitable for lambda and thus to extract the best features while disregarding the ones that have coefficients close to zero (thanks to the penalty function of Lasso). This is how we receive a smaller dataset which is helpful for any further analyses in both computationally and explanatory ways.

K-Means Clustering

K-Means Clustering method is less computationally intensive and hence it's suitable for very large datasets. Once we have the optimal number of clusters defined then the result of K-means clustering is relevant as we see in the above example where we defined the number of clusters as four and hence, we get all the data points perfectly fit into these clusters. However, if we run the algorithms many times, we will have different results in this method. Also, it's not good for the data points having similar attributes as all of them move into the same cluster and hence defining the structure of data points will be difficult using this method.

Principal components

● Advantages:

Dimensionality Reduction: PCA reduces the number of features, simplifying the dataset.
Noise Reduction: It helps in filtering out noise and focusing on the most important features.

Visual Representation: Data can be visualized in a reduced-dimensional space, aiding interpretation.

● Features:

Missing Values: PCA handles missing values well, making it suitable for datasets with incomplete information.
Multicollinearity: It is effective in dealing with multicollinearity, as principal components are orthogonal.

We compare the correlation matrix before PCA (fig.15) and after PCA (fig.16): we can observe highly correlated variables have been removed after PCA techniques were applied.

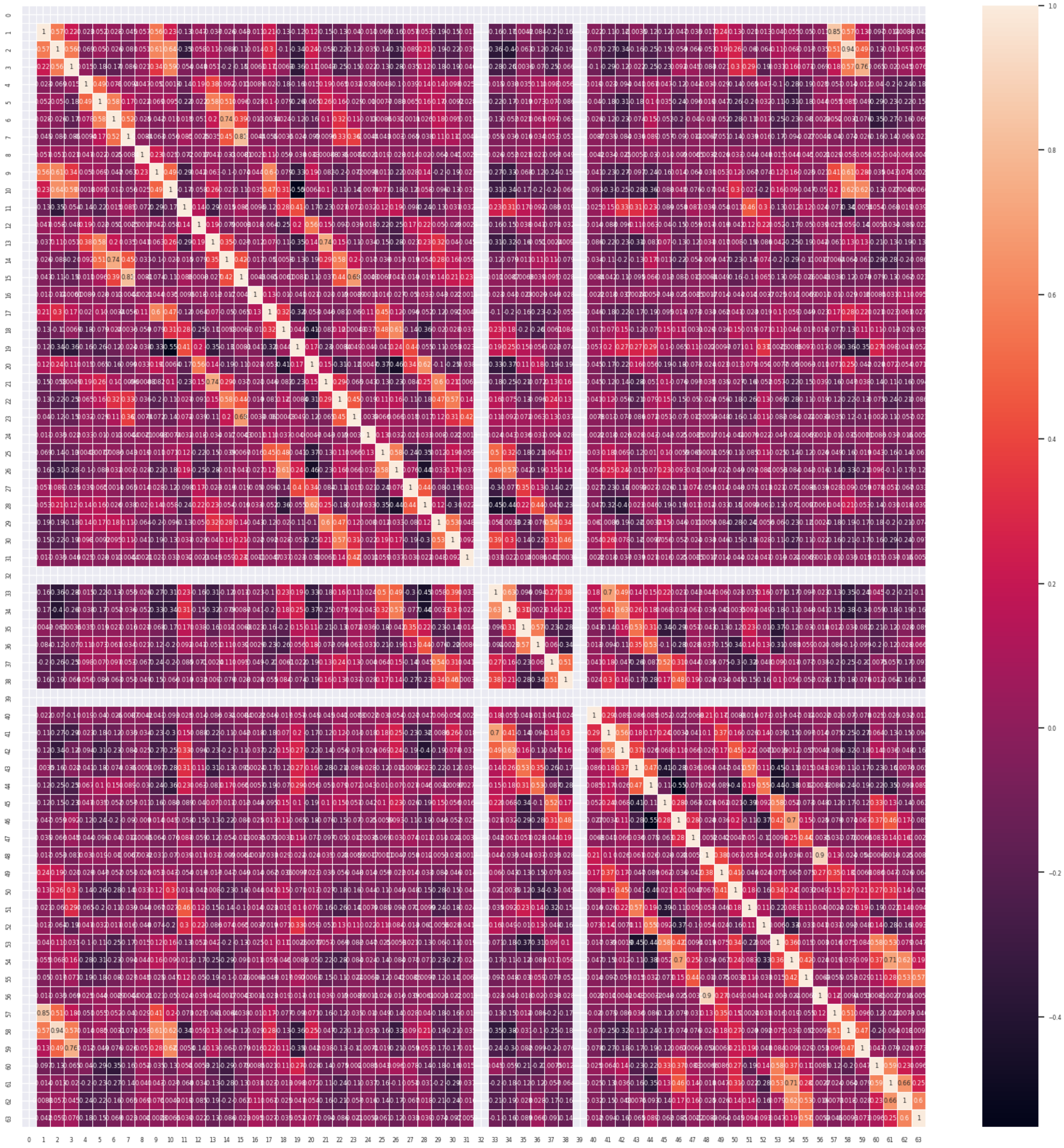


Fig. 15. Correlation matrix before PCA

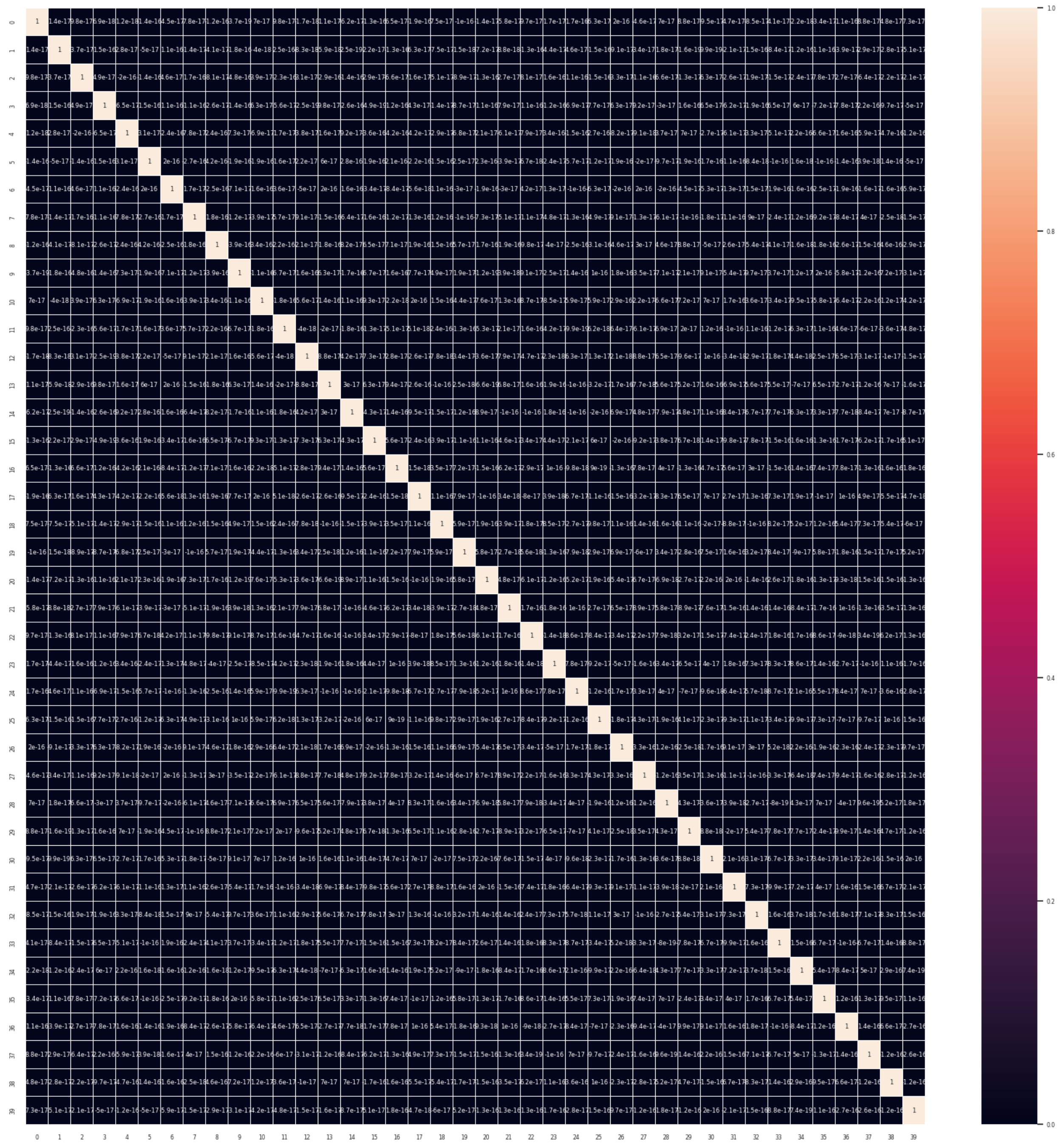


Fig. 16. Correlation matrix after PCA

Learn More

1. Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the lasso". Journal of the Royal Statistical Society. Series B (methodological). Wiley. 58 (1): 267–88. JSTOR 2346178

2. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani; An Introduction to Statistical Learning; Springer
3. Tahera Firdose, Lasso Regression: A Comprehensive Guide to Feature Selection and Regularization, Medium
4. Jorge A. Chan-Lau, Lasso Regressions and Forecasting Models in Applied Stress Testing, International Monetary Fund
5. Lasso documentation in sklearn.linear_model.Lasso
6. KP Sinaga, MS Yang: Unsupervised K-means clustering algorithm; IEEE access, 2020
7. M Yedla, SR Pathakota, TM Srinivasa: Enhancing K-means Clustering Algorithm with Improved Initial Center ; International Journal of computer science and Information Technology, 2010
8. Pooya Tavallali, Peyman Tavallali & Mukesh Singhal : K-Means tree: an optimal clustering tree for unsupervised learning; The Journal of Supercomputing; Springer
9. Avellaneda, Marco and Lee, Jeong-Hyun, Statistical Arbitrage in the U.S. Equities Market (July 11, 2008). Available at SSRN: https://ssrn.com/abstract=1153505 or http://dx.doi.org/10.2139/ssrn.1153505
10.

References:

1. Lari Giba, Lasso Regression Explained, Step by Step, https://machinelearningcompass.com/machine_learning_models/lasso_regression/
2. Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the lasso". Journal of the Royal Statistical Society. Series B (methodological). Wiley. 58 (1): 267–88. JSTOR 2346178
3. Jim Frost, Mean Squared Error (MSE), Statistics By Jim, https://statisticsbyjim.com/regression/mean-squared-error-mse/
4. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani; An Introduction to Statistical Learning; Springer
5. Tahera Firdose, Lasso Regression: A Comprehensive Guide to Feature Selection and Regularization, Medium
6. Lasso documentation in sklearn.linear_model.Lasso
7. Jorge A. Chan-Lau, Lasso Regressions and Forecasting Models in Applied Stress Testing, International Monetary Fund
8. https://scikit-learn.org/stable/auto_examples/compose/plot_digits_pipe.html

9. Digits dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html
10. SVC: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
11. Logistic regression: https://scikit-learn.sourceforge.net/dev/auto_examples/plot_digits_pipe.html
12. Logistic Python example: https://medium.com/@dlwilkinson/a-model-that-identifies-a-hand-written-number-using-logistic-regression-and-python-29308efd386a
13. Recognizing hand-written digits: https://scikit-learn.sourceforge.net/dev/auto_examples/classification/plot_digits_classification.html#example-classification-plot-digits-classification-py
14. https://www.tutorialspoint.com/machine_learning_with_python/clustering_algorithms_k_means_algorithm.htm
15. https://www.educba.com/k-means-clustering-algorithm/
16. https://www.geeksforgeeks.org/difference-between-k-means-and-hierarchical-clustering/
17. K-mean clustering example from WQU Machine Learning in Finance Module 2