GROUP WORK PROJECT # _1__
Group Number: _____7073_______

MScFE 632: Machine Learning in Finance

FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Pulkit Gaur	India	pulkit.gaur.iit@gmail.com	
Shivansh Kumar	India	shivansh.business23@gmail.com	
Lawal Nimotalai Abduganiyu	Nigeria	nabduganiyu@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	**Pulkit Gaur**
Team member 2	**Shivansh Kumar**
Team member 3	**Lawal Nimotalai Abduganiyu**

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.

Task

Marketing Handbook for cutting-edge machine learning methods for trading strategies.

Team Member A : K-means Clustering
Team Member B: PCA (Principal Component Analysis)
Team Member C: Lasso Regression

Step 1

K-mean clustering

K-means clustering is an unsupervised learning algorithm used for data clustering, which groups unlabeled data points into groups or clusters.

Advantages:

- It is very easy to implement and very computationally efficient
- It is scalable
- It has fast convergence and performs very well with smaller datasets compared to other clustering methods
- It is a clear and intuitive grouping of the data which can be interpreted easily by data scientists and analysts

Basics:

K-means is a centroid-based unsupervised ML algo used for clustering. It will partition data into predefined clusters. It groups similar data points together.

Computation:

Please refer to the colab notebook for this

Disadvantages:

- The number of clusters needs to be predefined and finding the optimal number is challenging
- It is sensitive to how we select the initial centroids
- It assumes that the clusters are spherical
- It is also sensitive to outliers

Equations:

- **Manhattan distance:**
  - $|a - b|_1 = \sum_i |a_i - b_i|$
- **Minkowski distance:**

$$\left(\sum_{i=1}^{n} |X_i - Y_i|^P\right)^{1/p}$$

- **Maximum distance:**

$$|a - b|_{\infty} = max_i |a_i - b_i|$$

- **Canberra distance:**

$$\sum_i |x_i - y_i| / |x_i + y_i|$$

- **Single linkage / nearest linkage:**

$$d(u, v) = min(dist(u[i], v[j]))$$

- **Complete Linkage:**

d(u,v) = max(dist(u[i], v[j]))

- **Cluster Assignment:**

$$C_i = arg\,min_j ||x_i - \mu_j||^2$$

- **Centroid Update:**

$$\mu_j = 1/|C_j| \sum_{x_i \in C_j} x_i$$

Features:

- **Scalability:** Works well with large datasets
- **Efficiency:** The algorithm is very fast and efficient
- **Data Types:** Used with numerical data but can handle categorical data as well
- **Non-Deterministic:** The final cluster can vary because of different random centroid initializations

Guide:

Inputs:

- k: number of clusters
- Data points to be clustered
- No. of maximum iterations
- Initialization methods of centroids
- Stopping criteria

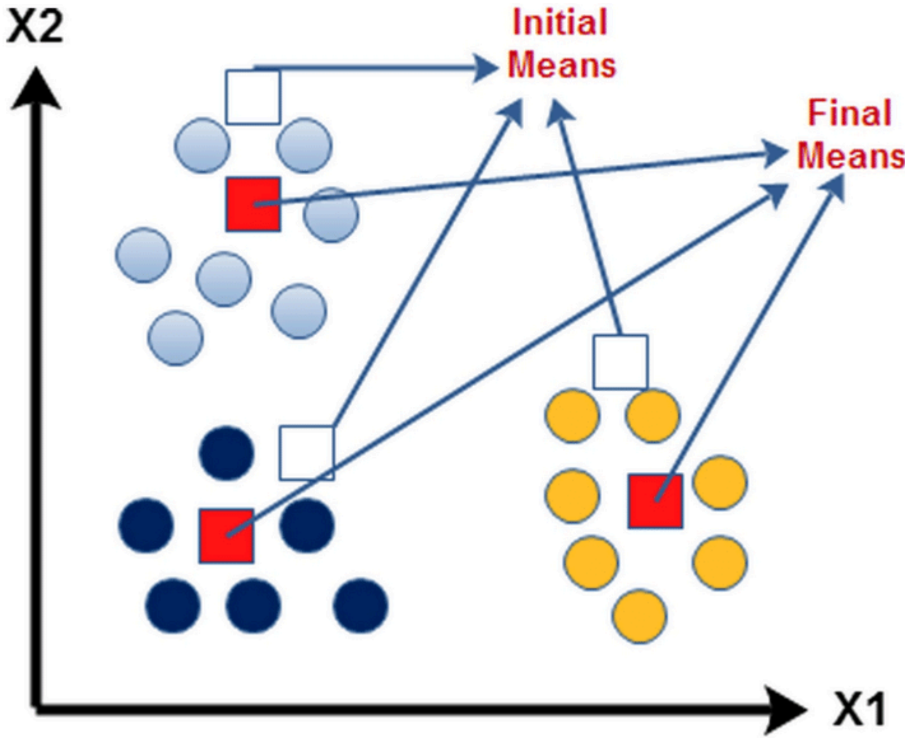
Outputs:

- Cluster assignments
- Positions of centroids

Hyperparameters:

- k: number of clusters
- Distance metric: type of distance metric used
- Max iterations: No. of maximum iterations
- Initialization method: Initialization methods of centroids
- Tolerance: Stopping criteria

Illustration: Here is the simple illustration of K-mean clustering, in which you can see how it started with some random centroids, and after running the k-means algorith we got our final centroids.



Journal:

https://medium.com/@cemalozturk/market-analysis-with-k-means-clustering-algorithm-identifying-support-and-resistance-levels-f49b963924f5#:~:text=The%20K%2DMeans%20algorithm%20can,points%20with%20similar%20price%20movements.

Keywords:

- **Unsupervised learning**
- **Clustering**
- **Centroids**
- **Euclidean distance**
- **High dimensional data**

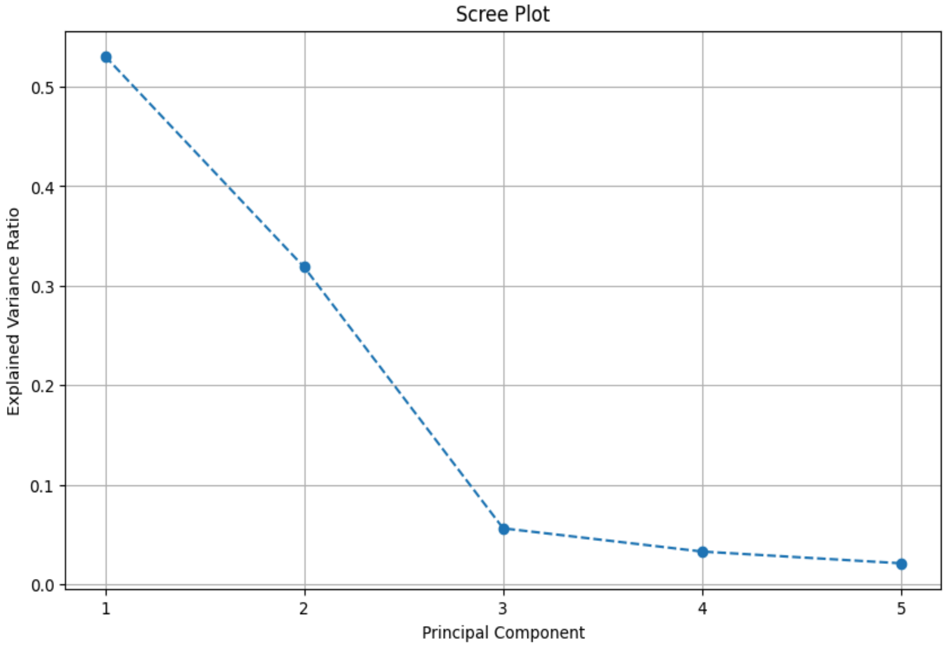
- **partitioning**

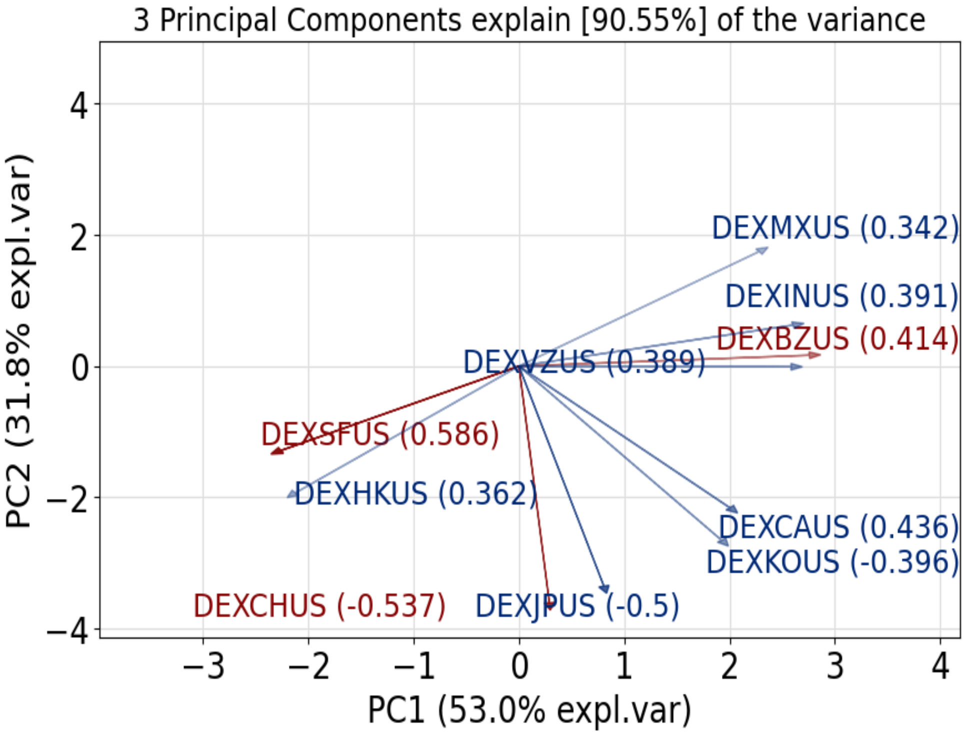
Principal Components analysis

PCA (Principal Components Analysis**)** is an unsupervised learning technique that is widely used for feature extraction and dimensionality reduction, in finance it is used for exploratory data analysis and the denoising of signals from the stock market data (Jaadi).

Now Let's briefly discuss all the aspects of PCA :

- **Advantages:**
    - **Feature Extraction:** PCA can help transform the data onto a new feature space which will act as a form of data compression that will only maintain the most relevant information that is needed for the learning algorithm.
    - The compression process will also help improve the storage space as well as the computational efficiency which will eventually improve the predictive performance of the model by reducing the curse of dimensionality ("Principal component analysis").
    - PCA also helps in identifying latent correlation patterns between features, it finds the directions of maximum variance in high-dimensional data and projects the data onto a new subspace with fewer dimensions than the original one.

- **Computation:**
    - For our computation, we have gathered forex data of 10 major currencies against dollar and we are trying to understand the relationship between them using PCA.
    - We have used the Scikit learn library ("PCA — scikit-learn 1.5.2 documentation") to perform PCA.
    - We have used the Scree plot to select a number of components for PCA. (Chauhan)
    - And at last, to understand the PC's completely we have used biplot.
    - **Scree Plot:** Here the image below is the scree plot which helps us select the number of PC.
        - 

- **Biplot of PC:** Here you can see the biplot of our PCA which has provided us a lot of insights we can understand the relationship between different features also the explained variance with it.
  - 

You can find the detailed computation report in the colab file included with the results.

- **Disadvantages:**
  - **Outlier Sensitivity:** PCA is sensitive to outliers, as outliers can drastically affect the covariance matrix which can lead to inefficiency in principal components.
  - **Dimensionality reduction trade-off:** As we all know PCA is a dimensionality reduction method due to which it can suffer from information loss.
  - **Lack of Robustness:** Small changes in data can lead to significantly difference principal components.
  - **Loss of Interpretability:** PCA transforms the original features into principal components, which is a linear combination of original features but sometimes it is difficult to interpret the new components in the context of original data.

- **Equations:**
  - To calculate PCA We will start from first with standardizing the features of a dataset. The StandardScaler transforms the data such that each feature has a mean of 0 and a standard deviation of 1.
    - **StandardScalar Formula** : $Z_i = (x_i - \mu) / \sigma$
      - Where Z_i = Standardized Value
      - X_i the original value
      - μ is the mean of the feature
      - σ is the std dev of the feature

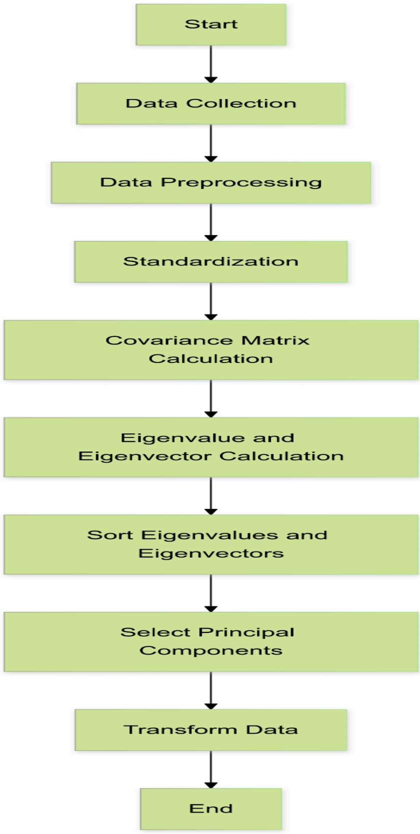
  - Now we will calculate the covariance matrix of the standard data.
    - **Covariance Matrix:** $\Sigma = 1 / n - 1 Z^T Z$
      - Where Z is the matrix of standardized data, and n is the number of data points.
  - After finding the covariance matrix we will move into finding the **eigenvalue** and **eigenvector** of the covariance matrix.
    - $\Sigma v = \lambda v$
    - Characteristic equation: $det(\Sigma - \lambda I) = 0$, where I is the identity matrix
  - We will now sort the **Eigenvalue** in descending order, and arrange the corresponding **Eigenvector** accordingly.
  - At last, we will Select **Principal Components** i.e. The eigenvectors corresponding to the largest eigenvalues are the principal components. Typically, we select the top k principal components that capture the most variance.

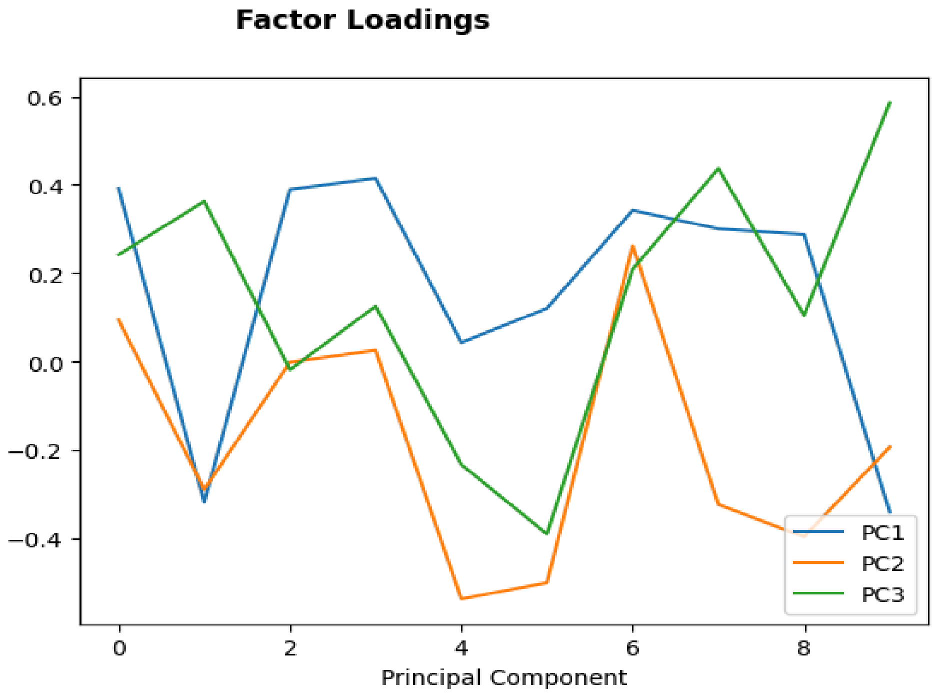
- **Features (Chauhan):**
  - **Feature extraction:** PCA reduces the number of features by transforming the original features into smaller sets called principal components.
  - **Data compression:** It helps in compressing high dimensional data while retaining most of the variance which makes it easier to visualize and analyze in the context of high dimension.
  - **Noise Reduction:** PCA also reduces noise in high dimensional data by focusing on only necessary principal components which capture most of the variance and filter out the unnecessary noise.
  - **Unsupervised learning:** PCA is an unsupervised learning technique which means it does not require labeled data.

- **Reducing Multicollinearity:** While working on high dimensional data there is a high chance of multicollinearity PCA also solves that as the principal components derived by PCA are orthogonal to each other.
- **Computationally efficient:** PCA is also very computationally friendly as it removes the curse of dimensionality which makes it efficient to be applied on large data sets.
- **Variance capture:** PCA analysis captures variance in data pretty well within its first few principal components.

- **Guide (Chauhan):**
    - **Inputs:**
        - **Data Matrix Z:** An n X m matrix where n is the number of data points and m is the number of features.
        - **Number of Components k:** The number of Principal components to retain.
    - **Outputs:**
        - **Transformed data Z:** n X K matrix representing the data projected onto the first k principal components.
        - **Eigenvector V:** An p X k matrix containing the first k eigenvector(principal components)
        - **Eigenvalue λ:** A k X k diagonal matrix containing the eigenvalues corresponding to the first k principal components.

- **Hyperparameters:**
    - PCA is not a typical ML model which needs lots of hyperparameter tuning as it is traditionally deterministic, however, there are a few parameters that we need to consider when working with PCA
        - **Number of PC k (Chauhan):** The number of PC to retain. It needs to be tuned based on our desired level of dimensionality reduction approach and the amount of variance we want to preserve.
        - **Scaling:** Standardizing the data before PCA is also a very important parameter to consider; it ensures all features contribute equally to the analysis, which is very essential for the robustness of PCA.

- **Illustrations:**
  - **PCA Working Flowchart:** Here the image below is the working flow chart of PCA.
    - 
Start
Data Collection
Data Preprocessing
Standardization
Covariance Matrix Calculation
Eigenvalue and Eigenvector Calculation
Sort Eigenvalues and Eigenvectors
Select Principal Components
Transform Data
End

  - **PCA Factor Loadings graph:** Here the image below shows the 3 PC factor loadings that we have slected.
    - 

- **Journal:**
  - Yu, Huanhuan, Rongda Chen, and Guoping Zhang. "A SVM stock selection model within PCA." *Procedia computer science* 31 (2014): 406-412.

- **Keywords:**
  - Dimensionality Reduction
  - Eigenvectors, Eigenvalue
  - Covariance Matrix
  - Explained Variance
  - Loadings
  - Feature Extraction
  - Data Compression

LASSO regression

Lasso (Least Absolute Shrinkage and Selection Operator) is a type of regression in supervised learning used for feature selection and regularization. It is a method that adds a penalty term to the loss function to regularize the model.

Advantages:

- **Sparsity:** Lasso regression shrinks the coefficients of irrelevant features of the model to zero giving a model that is simpler, with fewer features, and easier to interpret
- **Reducing Overfitting:** The addition of a penalty term to regularize the model helps to control the complexity of the model and reduce overfitting.
- **Handling Multicollinearity:** Lasso performs better than standard regression when the model features are highly correlated.
- **Improved Generalization:** Lasso regression, being a regularized model, generalizes better to new data making it perform better than standard regression.

Equations:

The regularization or penalty term in Lasso regression is of the form:

$$\alpha \sum_{i=1}^{n} |\theta_i|$$

The Lasso regression loss (or cost) function is the Mean Squared Error + the penalty term, i.e

$$= \sum_{i=1}^{n} (y_i - \widehat{y}_i)^2 + \alpha \sum_{i=1}^{n} |\theta_i|$$

where θ_i are the coefficients of the predictors (or features).

The goal is to find values of θ_i that minimize the cost function of the model. Using the gradient descent to achieve this, we:

- Initialize θ_i
- Update θ_i iteratively by computing

  $$\theta^{|next\ step|} = \theta^{|previous\ step|} - \eta\nabla_{\theta} MSE\ (\theta^{[previous\ step]} - a \odot \left\{\begin{matrix} 0 \\ \text{sign()} \\ \vdots \\ \text{sign()} \end{matrix}\right\}$$

  where $sign(\theta_i) = \{-\ 1\ if\ \theta i\ >\ 0\ ,\ 1\ if\ \theta i\ <\ 0,\ [-\ 1, 1]\ if\ \theta i\ =\ 0\}$

  This method entails the use of a subgradient to adjust the gradient descent because the Lasso loss function is not differentiable $\theta_i = 0$.

Features:

- Lasso regression produces a model that is simpler and easier to interpret
- It works well to handle multicollinearity in the model's features.
- It results in smaller coefficients which prevents overfitting
- It can be combined with different loss functions making it versatile for solving different supervised learning problems.
- Lasso can introduce bias but often reduces variance making it more generalizable on new data.

Disadvantages:

- **Linear assumption:** Lasso regression assumes a linear relationship between the features and the predicted variable and so, will not be able to capture non-linear relationships.
- **Correlation Bias:** If two features are highly correlated, lasso might select one feature and shrink the coefficient of the other to zero even if both are important features.
- **Randomness:** Lasso selects features in a random manner when they have similar importance which may yield different results on the same data.
- **Lasso is sensitive to the scale of the features**: Features need to be scaled before running Lasso regression on the data.

Computation:

- Please refer to the .ipynb file.
- A lasso regression model was used to predict returns of NVDA stocks (as the target variable) with the returns of Apple, Amazon, Cisco, Google, IBM, and Microsoft stocks as the features.

Guide:

- **Inputs:**
  - **Target variable (y):** Matrix of data representing the dependent variable to be predicted.
  - **Features (X):** Matrix of data with columns of independent variables.
  - **Regularization parameter (α):** which controls the strength of the penalty term
  - **Coefficients:** Initial values of the coefficients.

- **Outputs:**
  - **Intercept:** The constant term
  - **Coefficients:** The resulting coefficients of the model features with some shrunk to zero
  - **Model performance metrics:** such as R-squared and adjusted R-squared

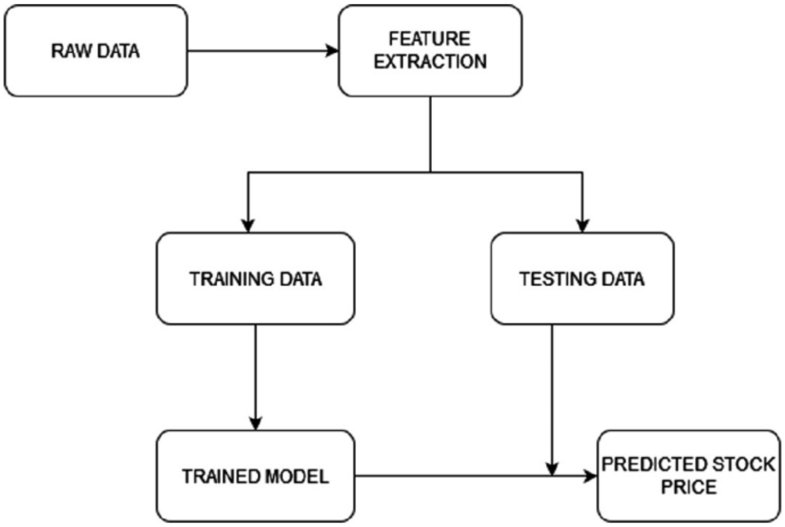
Hyperparameters:

These are the parameters of the model that require tuning.

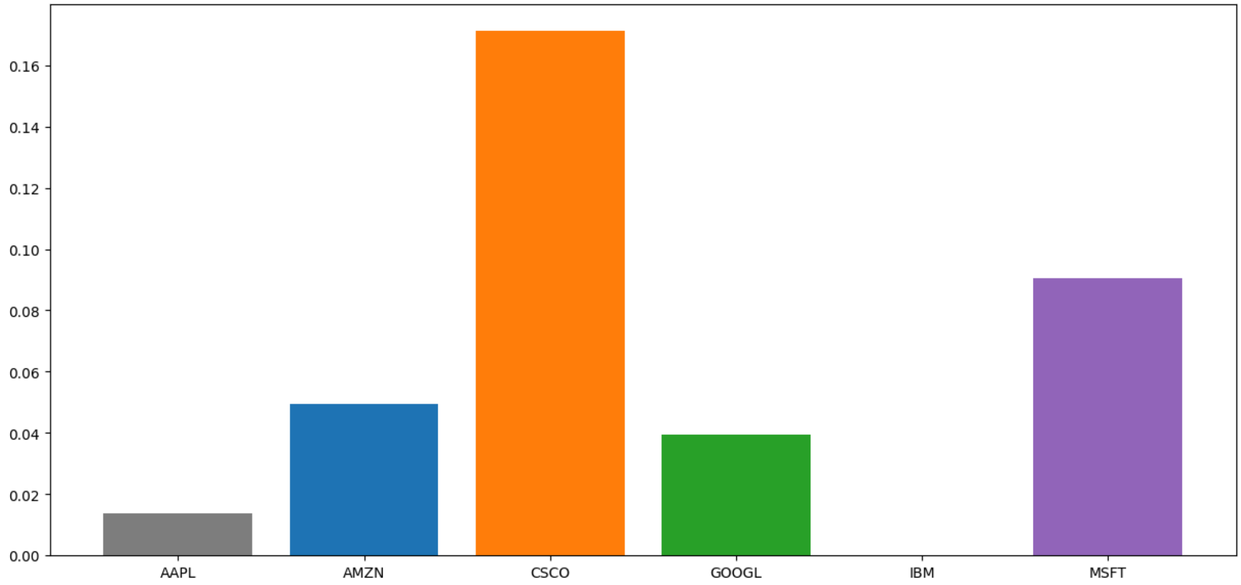
- **Regularization parameter (α):** It controls the strength of the penalty term. The higher the α, the more the model is regularized. There is the risk of overfitting when α is too small and underfitting when α is too large. To make the optimal choice of α, we use cross-validation.

Illustrations:

The diagram below shows a summary of the Lasso regression process



After running Lasso regression, we see that the IBM feature has been excluded from the prediction.



Journal:

Nie, X. and Deng, G. (2020) Enterprise Financial Early Warning Based on Lasso Regression Screening Variables. *Journal of Financial Risk Management*, **9**, 454-461.
doi: 10.4236/jfrm.2020.94024.

Keywords:

- Regression
- Features selection
- Regularization
- Penalty term
- Sparsity
- Coefficient reduction
- Supervised learning

Step 3

Technical Section:-

- **K-means Clustering:**
  - We can tune the number of clusters param k by using methods like the elbow method or silhouette score.
  - For the initialization of centroids, we can choose either random initialization or k-means++
  - For distance metrics, we can try different distance metrics according to the dataset and see what performs better
- **Principal Components Analysis:**
  - As we have already discussed the working of PCA and how it is widely used as a dimensionality reduction technique that transforms the original features into a set of linearly uncorrelated variables called principal components. PCA is generally considered as a parameter-free model which means there are not many hyperparameters to tweak but there are some minor tuning that we can do to optimize the performance of our model.

    The hyperparameter tuning that we are going to discuss are number of PC and Scaling.

  - Scaling is nothing but Standardizing the data before PCA so that all features contribute equally to the PC which will lead to an accurate representation of the explained variance across the PC. In our example we have used data from different currencies of different countries in terms of dollars and as we know each country's currency values are different against the dollar we need to standardize the data for easy analysis.
  - Now let's discuss the most important Hyperparameter of PC which is the n_components or PC, this is the most critical hyperparameter in PCA it directly affects the dimensionality of the transformed data. We can use different tuning strategies for choosing n_components like explained variance, scree plot analysis, or domain-specific requirements but in our example, we have used explained variance as well as scree plot analysis, we started with randomly selecting 5 n_componenets Then after running the PCA algorithm we plot the scree plot to understand how many n_compnents are explaining the most of the explained variance.

Lasso Regression

- The Lasso regression hyperparameter (**α**) is tuned to find the optimal value. Using a value too small for alpha will allow the model to fit the model more and may result in overfitting. When **α** is too large, the model is highly regularized and this may lead to underfitting.
- To get an optimal value for **α**, we perform k-fold cross-validation across different subsets of the data to prevent overfitting. To perform the k-fold cross-validation:

1. **Split the data:** The dataset is split into k-folds of equal size.
2. **Iterate:** For each fold:
    - **Train:** Use k-1 folds as the training set to train the Lasso regression model.
    - **Test:** Use the remaining fold as the test set to evaluate the model's performance.
3. **Average:** Calculate the average performance metric (e.g., mean squared error, accuracy) across all k iterations to evaluate the performance of the Lasso regression model.

Step 4

Marketing Alpha:-

- **K-means:**
  - From the advantages and features of the k-means algorithm, we can see how beneficial it can be for finance data as it allows for creating meaningful clusters of large datasets fast and efficiently. Thus, it can be used to identify patterns of spending of customer groups, different investment portfolios, credit risk profiles of people, etc. The simplicity and efficiency of K-means algo can help to analyze large amounts of data in real-time and can be very beneficial for real-time trades. It can also help in finding similar profiled assets, so can help in portfolio diversification.

- **Principal Components Analysis:**
  - In finance, a rapidly evolving landscape, leveraging ML techniques has become essential. In this section, we aim to convey all the insights and findings from our report on PCA. We will understand how PCA can be a very effective method in our working arsenal to derive a better outcome. We have already discussed the advantages of PCA on how they can help in feature extraction in high dimensional data which can compress our data while preserving the most relevant information in our own computation example we have a 10 currency exchange rate against the dollar and we have reduced the dimension to 5 and still maintains more than 90% of relevant information. It not only made our computation faster but also required less storage space.
  - We converted our 10 column data into 5-column using the feature extraction features of PCA which transformed the original features into PC, it compresses our data increasing its computational efficiency, and also retained most of the variance that we have used in visualization in our computation section.
  - It also removed all the unnecessary noise in the data and use only the important PC which captured most of the variance which in our case was 3 PC as it captured more than 90% of the explained variance.
  - In unsupervised learning this method work like a charm as it does not require any labelled data and also it reduces multicollinerity in the high dimensional data.
  - At last it removes the curse of dimensionality and make out computation faster and also captures the variance in the data well within first few PC.
  - We also saw the ability of PCA in our example of forex rate of different countries against dollars, we can see in the biplot above that how PC1, PC2, PC3 are explaining more than 90% of explained variance means it is capturing the general trends of different currencies against the dollars for the observed period.

- We can saw DEXJPUS(-0.5) i.e Japanese yen is negatively correlated with PC1, meaning the yen has likely weakened against the dollar.
- Currencies like INR, MXN, BZ, VZ all are clustered together indicating that are highly correlated also as these are emerging markets they show similar trends against the USD.
- And at last if we talk about general trends the INR, CAD, and MXN show stability against the USD the CNY and JPY are very volatile and also shows weakening against the USD.
- CHF have also shown stability ans strength, overall using PCA we could easily capture the general trend, volatility as well as strength stability of different currencies which makes PCA a wonderful tool in our arsenal to understand high dimensional data.

- **LASSO REGRESSION**

  - Lasso regression can be used to select the most relevant features for our prediction model, thereby enhancing the denoising of our model, making it less cumbersome and easier to interpret.
  - In predicting the returns on NVDA stock, we used LASSO regression to select the most important features from the returns of AAPL, GOOGL, AMZN, CSCO, MSFT, and IBM. The model selected 5 of the 6 features excluding IBM. The correlation heatmap shows that IBM has the least correlation with the target variable, NVDA.
  - The use of a penalty term in the Lasso regression loss function enhances its generalization and makes it applicable to more real-world situations.

Step 5

Learn More:-

- **K-means:**
  - **Theoretical Analysis of the k-Means Algorithm – A Survey:** Johannes Bl¨omer∗ Christiane Lammersen† Melanie Schmidt‡ Christian Sohler§
  - **Early Warning of Financial Risk Based on K-Means Clustering Algorithm**

- **PCA**
  - Lameira, Pedro. "Pair trading: Clustering based on principal component analysis." Work Project. NOVA-School of Business and Economics (2015).
  - Yu, Huanhuan, Rongda Chen, and Guoping Zhang. "A SVM stock selection model within PCA." *Procedia computer science* 31 (2014): 406-412.
  - Abdi, Hervé, and Lynne J. Williams. "Principal Component Analysis." Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, no. 4, 2010, pp. 433-459. Wiley Online Library, doi: 10.1002/wics.101.

- **LASSO REGRESSION**
  - Nie, X. and Deng, G. (2020) Enterprise Financial Early Warning Based on Lasso Regression Screening Variables. *Journal of Financial Risk Management*, **9**, 454-461.
    doi: 10.4236/jfrm.2020.94024.
  - Roy, S.S., Mittal, D., Basu, A., Abraham, A. (2015). Stock Market Forecasting Using LASSO Linear Regression Model. In: Abraham, A., Krömer, P., Snasel, V. (eds), Afro-European Conference for Industrial Advancement. Advances in Intelligent Systems and Computing, vol 334. Springer, Cham.
    https://doi.org/10.1007/978-3-319-13572-4_31

References

1. Chauhan, Nagesh Singh. “Dimensionality Reduction with Principal Component Analysis (PCA).” *KDnuggets*, 21 May 2020, https://www.kdnuggets.com/2020/05/dimensionality-reduction-principal-component-analysis.html. Accessed 18 September 2024.
2. Jaadi, Zakaria. “Principal Component Analysis (PCA) Explained.” *Built In*, https://builtin.com/data-science/step-step-explanation-principal-component-analysis. Accessed 18 September 2024.
3. “PCA — scikit-learn 1.5.2 documentation.” *Scikit-learn*, https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html. Accessed 18 September 2024.
4. “Principal component analysis.” *Wikipedia*, https://en.wikipedia.org/wiki/Principal_component_analysis. Accessed 18 September 2024.
5. https://www.researchgate.net/figure/Illustration-of-K-means-clustering_fig5_355474076
6. https://en.wikipedia.org/wiki/K-means%2B%2B
7. https://www.publichealth.columbia.edu/research/population-health-methods/least-absolute-shrinkage-and-selection-operator-lasso
8. WQU, Machine Learning Lesson Notes, Module 1, Lesson 2, “Regression and Hyperparameters”. https://vm.wqu.edu/lab/tree/work/mscfe-machine-learning/module-1/lesson-2/machine_learning_module_1_lesson_2.ipynb

9. Shailabh, Varma. 27, June, 2024 “Unlocking the Power of LASSO Regression: A Comprehensive Guide”, Pickl.AI