GROUP WORK PROJECT # 1 MScFE 632: Machine Learning in Finance
GROUP NUMBER: 7764

FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Mphikeleli Mbongiseni Mathonsi	South Africa	mphikelelimm@gmail.com	
Santosh Kumar	India	santoshkumaragm@gmail.com	
Tu Thi Cam Phan	Australia	tuphan1441@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	Mphikeleli Mbongiseni Mathonsi
Team member 2	Santosh Kumar
Team member 3	Tu Thi Cam Phan

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

Step 1

Category 1: LASSO Regression

LASSO regression is one of the methods of supervised learning techniques and is mainly used for regression and feature selection. This extends linear regression by using an L1 penalty term added to the loss function; hence, it sums the absolute values of the coefficients. Due to this penalty, some of the regression coefficients shrink exactly to zero, effectively removing less important predictors and yielding a sparse model. Mathematically, the objective function of LASSO can be represented as:

$$Minimize: \frac{1}{2N}\sum_{j=1}^{p}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

Where:

- N: Number of observations.
- y_i: Actual output.
- $\hat{y}_i$: Predicted output.
- λ: Regularization parameter controlling the degree of shrinkage.
- β_j: Coefficient of the j-th predictor variable.

LASSO has its greatest strengths in high-dimensional datasets where the number of predictors is larger than the number of observations. It helps to reduce multicollinearity and avoids overfitting by selecting only the most relevant features. Unlike Ridge regression, an L2 penalty which pulls coefficients toward zero, in LASSO, it can force coefficients to be zero, hence useful in the dimensionality reduction process.

Applications in Finance

Other interesting applications of LASSO occur in portfolio optimization, credit risk modeling, and forecasting of stock return in finance. The identified most influential predictors simplify models found by LASSO, doing so without large sacrifice of predictive power. For instance, LASSO has been applied for effective selection of relevant economic indicators out of hundreds to predict equity returns by researchers (Fan & Li, 2001).

Keywords

- Feature selection
- Regularization
- Linear regression
- L1 penalty
- Shrinkage method
- Sparse modeling
- Dimensionality reduction
- Ridge vs. LASSO

- Overfitting control
- Predictive modeling

Category 2: K-Means Clustering

Basics

In machine learning k-means clustering algorithms come under unsupervised learning methods. This algorithm divides the data into a previously defined number of clusters. It works on the principle of grouping similar types of data sets with each other and degrouping those parts of data sets which are dissimilar.

It works in following way:

- Step 1: First, we initialize the algorithm randomly.
- Step 2: Then we define the number of (K) centroids.
- Step 3: Then the algorithm associates each data point with a centroid based upon the euclidean distance method.
- Step 4: Then there is an update of centroids by finding groupings.
- Step 5: The algorithm repeats above step number 2 & 3 till stabilization of centroids or until maximum iterations.

The formula (objective function) behind this algorithm is as follows:

$$E = \sum_{i=1}^{k} \sum_{x \epsilon C_i} |x - m_i|^2$$

Where:

K – Clusters (Number).

C- Points in each cluster.

m- Each cluster's centroid.

Key words:

- Unsupervised machine learning
- Centroid.
- Assignments of clustering.
- Key metric like euclidean distance
- Hyper parameters
- Optimal convergence
- Variance in clusters

Category 3: Principal components

- **Basics**

Principal Component Analysis (PCA) is a statistical method to reduce the dimensionality and pull the features out. The technique turns a dataset of correlated features into the much smaller dataset of uncorrelated variables called primary components. These components are a linear combination of the original features and are sorted based on the amount of variance they explain.

    - **Step by step in PCA:**
        - **Standardization:** The mean of all these features is set to 0 and the standard deviation of 1 to ensure that each feature contributes equally to the analysis.
        - **Covariance Matrix:** PCA calculates the covariance matrix to measure the interactions between the features.
        - **Eigenvectors and Eigenvalues:** It decomposes the covariance matrix into eigenvalues (variance explained) and eigenvectors (directions of principal component).
        - **Projection:** The data is projected along the principal components, thus, reducing the dimensions while the key features are preserved.
- **Keywords**
    - **Dimensionality Reduction:** PCA is a procedure whose main purpose is to diminish the amount of input data without losing its essential elements.
    - **Orthogonality:** Principal components are orthogonal to each other, as a result there is no redundancy or correlation of components.
    - **Eigenvalues:** The variance, which is explained by each principal component is denoted by this number.
    - **Eigenvectors:** Each principal component in the feature space corresponds to its corresponding eigenvector.
    - **Variance Maximization:** PCA attempts to capture the greatest variance allowed by these principal components.
    - **Covariance Matrix:** Shows how entities are interconnected with each other and is used as the main element in the dimensionality reduction technique.
    - **Feature Transformation:** It is the process of converting dependent features into principal components that are uncorrelated to one another.
    - **Unsupervised Learning:** PCA is an algorithm that is used in unsupervised learning techniques. It helps to extract overall trends and structure from the data set.
    - **Explained Variance Ratio:** Explained the variability captured by each principal component as a ratio of total variance.
    - **Data Compression:** PCA reduces the amount of data needed in the transformation, usually done as a pre-process. It can be ascertained from the results.

Step 2

Team Member A Works on Model 1 – LASSO Regression

Advantages

- **Feature Selection**: LASSO can eliminate irrelevant predictors by shrinking their coefficients to zero, hence yielding a more interpretable model.
- **Handles High Dimensionality:** Applicable when the number of features is larger than the number of observations.
- **Reduces Overfitting:** Regularization prevents the model from capturing noise, improving generalization to unseen data.
- **Sparse Models:** It generates sparse solutions, which are computationally efficient and easier to interpret.
- **Improves Model Parsimony**: By only selecting the most influential predictors, LASSO allows for model simplicity without sacrificing predictive power.

Computation:

Codes.ipynb

Disadvantages

- **Bias in Coefficients:** LASSO tends to shrink coefficients. This results in biased estimates.
- **Sensitivity to Scaling**: LASSO requires the data to be standardized; else it gives undue importance to variables with larger magnitudes.
- **Single Regularization Parameter:** Performs poorly when the relationship between predictors and response is complicated or non-linear.
- **Correlated Features:** If the predictors are highly correlated, LASSO arbitrarily chooses one and discards the rest.
- **Model Instability:** Small changes in the data can lead to different selected features, reducing reproducibility.

Equations

Mathematically, the objective function of LASSO can be represented as:

$$Minimize: \frac{1}{2N}\sum_{j=1}^{p}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

Where:

- N: Number of observations.
- y_i: Actual output.
- $\hat{y}_i$: Predicted output.

- λ: Regularization parameter controlling the degree of shrinkage.
- β_j: Coefficient of the j-th predictor variable.

Features

- Does feature selection automatically.
- Works well in high-dimensional data.
- Is able to handle collinearity partially.
- Scaling is required for input features
- Sparse coefficients are generated.
- Guide: Inputs and Outputs

Input:

- Independent variables(features)
- Dependent variable(target)
- Regularization parameter λ

Outputs:

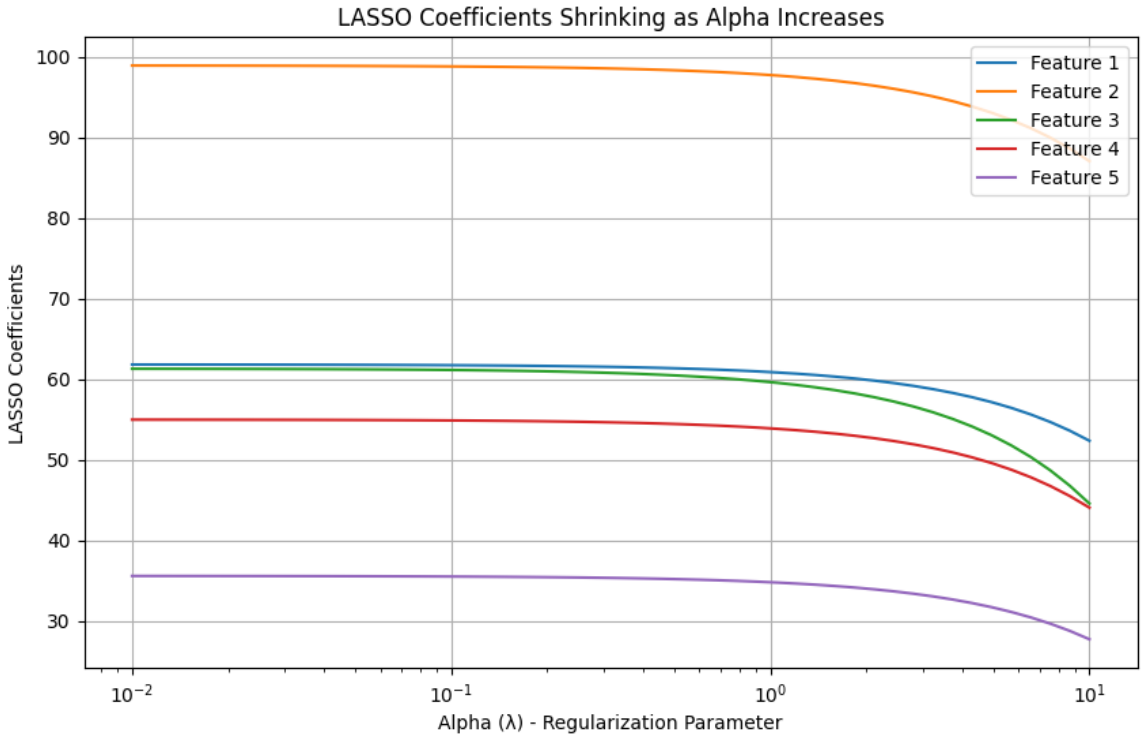
- Optimized model coefficients
- Predicted values
- Performance metrics (e.g., MSE, R-squared)

Hyperparameters

- Alpha (λ): Controls the higher the alpha, the more the shrinkage.
- Tolerance: When optimization must stop.
- Max Iterations: Number of iterations to converge to a solution.

Illustration

- LASSO Coefficients vs. Regularization Parameter: A plot showing how the coefficients shrink as λ increases.



Journal Reference

- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267-288.
    - This foundational paper introduced LASSO regression and its use for variable selection.

Team Member B Works on K-Means Clustering:

Advantages

- In implementation it is easy and in computation also it is efficient.
- It is scalable and can work well with bigger data sets.
- It is easy to interpret.
- With less number of iterations it can converge quickly when we compare with other clustering methods such as hierarchical clustering.
- It is widely used in many applications like segmenting customers, detecting anomalies, compression of image etc.

Computation

Codes.ipynb

Disadvantages

- How we initialize the method K Means clustering is very sensitive to that. Results are dependent on initial initializations of centroids.
- This method needs to specify the required number of clusters before initialization of the method.
- K means clustering does not perform well when clusters are overlapping or non-spherical.
- When data contains outliers or noises etc then K means is very sensitive to that. The search of centroids may be skewed then.
- Many times, it may happen that the method may converge to a solution that is not very optimal.

Equations

For K means clustering we have the following objective function. This objective function minimizes the sum of squared differences among different points and the centroids.

$$E = \sum_{i=1}^{k} \sum_{x \epsilon C_i} |x - m_i|^2$$

Where:

K – Clusters (Number).

C- Points in each cluster.

m- Each cluster's centroid.

Features

- It is an unsupervised learning method, and it requires no labels.
- It uses algorithms which are iterative. (Like EM- expectation maximization etc).
- Metric that it uses by default is Euclidean distance.
- Clusters in the method are based upon computation of centroids.
- When we calculate distance in the method, we require features that are numeric.

Guide

- List of inputs are Data having features that are numeric, we have to define the number of clusters initially.
- The outputs are Assignments of clusters, All centroids.

Hyperparameters

- n_clusters - Cluster's number (K).
- max_iter: Iterations (Maximum) that are allowed.
- init: Method of initialization (It can be k-means++, random etc).
- tol: Convergence tolerance.
- random_state: To reproduce the seed.

Illustration

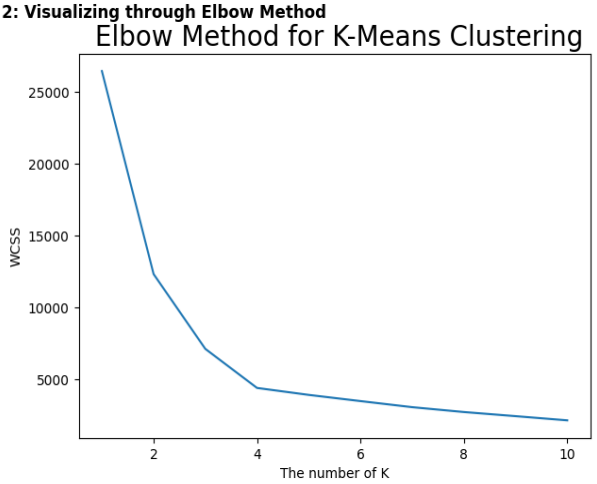
Following is a flowchart Which depicts k-means algorithm:

- Step 1: First, we initialize the algorithm randomly.
- Step 2: Then we define the number of (K) centroids.
- Step 3: Then algorithm associates each data point with a centroid based upon the euclidean distance method.
- Step 4: Then there is an update of centroids by finding grouping mean.
- Step 5: The algorithm repeats the above step number 2 & 3 till stabilization of centroids or until maximum iterations.

The scatter plot in section 1.1.2 (Computation of K Means algorithm)

Figure 1: Visualizing dataset

Figure 2: Visualizing through Elbow Method



Journal

- Reference: "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis" – Future Generation Computer Systems, 2021.

Keywords

- Unsupervised machine learning
- Centroid.
- Assignments of clustering.
- Key metric like Euclidean distance
- Hyper parameters

- Optimal convergence
- Variance in clusters

Step 2

Team Member C Works on PCA

Advantages:

- **Variate Preservation:** The variability of the original dataset is effectively retained in a reduced number of components by PCA. By giving priority to the components with the highest variance, PCA guarantees that the most significant information in the data is saved, even after dimensionality reduction.
- **Noise Reduction:** PCA eliminates the effect of noise by ignoring those features which have low variance, which are generally spatially spread out or represent noise or representations of less important patterns. This leads to making a cleaner dataset that is more focused on the meaningful structure of the data.
- **Visual Plotting, Analysis:** Through the PCA method, whose main objective is to project a hyperdimensional (2D, 3D, etc.) data set onto a lower-dimensional space, one gets an understandable visualization of the patterns and the nature of the clusters and the relationships that are almost never discernible from the original data pool.
- **Enhanced Model Performance:** PCA is used to address multicollinearity through the transformation of correlated features into uncorrelated principal components. This simplification creates a more stable and efficient learning algorithm, particularly for those that are sensitive to correlations of features, such as linear regression and logistics regression.

Computation:

Codes.ipynb

Disadvantages:

- **Loss of Interpretability:** While PCA diminishes dimensionality, the principal components are diverse linear combinations of the original features. Therefore, this conversion makes it very problematic to interpret what each component actually means of the original attitudes. For instance, if a principal component encompassed multiple variables, comprehending its real-world implication could be difficult, particularly in applications where the relationships between the features are distinct.
- **Variance Dependency:** PCA puts emphasis on the scale of the input data as it selects features according to the variance. The features that are high in magnitude might have a major effect on the principal components, so the results are inaccurate. In order to solve this, data normalization

or standardization is supposed to be done before using PCA. This step, however, not only increases the complexity of preprocessing but also gives rise to error if it is carried out in an improper way.

- **Not Ideal for Nonlinear Data:** PCA assumes that the interactions among the features are linear, which is not the case for datasets with alternative relationships. Therefore, in such cases, PCA may not be able to reveal the actual structure of the data that it is supposed to represent, hence producing dimensionality reduction that is not optimal. Other approaches such as kernel PCA or t-SNE are more appropriate for nonlinear data.

Equations:

PCA Step by Step

1. **Covariance Matrix:**

$$C = \frac{1}{n-1}(X - \bar{X})^{\top}(X - \bar{X})$$

2. **Eigenvalues and Eigenvectors:**

$$Cv = \lambda v$$

In which, λ is the eigenvalues or variance explained, and v is the eigenvectors or principal components.

3. **Transform Data:**

$$Z = X \cdot V$$

Features:

Scalability: PCA has a very effective way in the presence of a large number of features due to its scalability thus being fully well suited for the purpose of handling data in various dimensions. Nevertheless, the computational cost could become significantly large while the dataset size is enlarged, mainly when the covariance matrix or the vector decomposition is computed, but randomized PCA and other advanced methods can be an efficient solution for this trouble. can be an efficient solution for this trouble. can be an efficient solution for this trouble. can be an efficient solution for this trouble. can be an efficient solution for this trouble too.

Sensitivity to Scale: The PCA technique is sensitive to the scaling of the features in relation to each other, since it is based on variance for the identification of the main components. Features with larger magnitudes will have more impact on the components. As a result, scaling the dataset (e.g., standardizing the features to have zero mean and unit variance) is a preprocessing step that must be done as it is impossible to meaningfully compare the PCA results if the data was analyzed differently during scale differences.

Linear Relationships: The main aim of PCA lies in the detection of linear relationships among features in the dataset and the performance of the algorithm is at its best when the data has linearity. It captures the directions of maximum variance, which are the linear patterns in the feature space.

Guide:

- **Input**

The information about each of the variables being the space a sample occupies should be transformed by PCA from the data matrix X . This input is the intersection of n observations (rows) and p features (columns). Note that one feature is equivalent to one and only one variable and one observation represents a sample. This datum is often standardized by subtracting the means of each feature from every row and dividing by the standard deviation of the corresponding column, thereby scaling the data with mean equal to 0 and standard deviation equal to 1, so that all features contribute proportionally to the result of PCA. Also, the other important entry is the number of components to keep which tells us the dimensionality of the reduced dataset. Such an adjustable parameter is typically based on the cumulative explained variance, so the system practically ensures that the selected components subsume a big enough part of the dataset's variability.

- **Output**

The most significant by-products of principal component analysis are the key components that permit a transformed dataset in a new space that is smaller. Each component is simply a linear combination of some input variables and is arranged by its risk of capturing more or less random data. Other than the above, PCA also reports the explained variance ratio for each component. Hence, this is a portion of the variance explained by each component with respect to the total variance. Thereby it can be immediately clear to the users which component is influential and how many they should keep for other analysis.

Hyperparameters:

- $n\_components$**:** Number of principal components to retain.

- svd_solver**:** Algorithm for decomposition (e.g., `auto`, `full`, `randomized`).

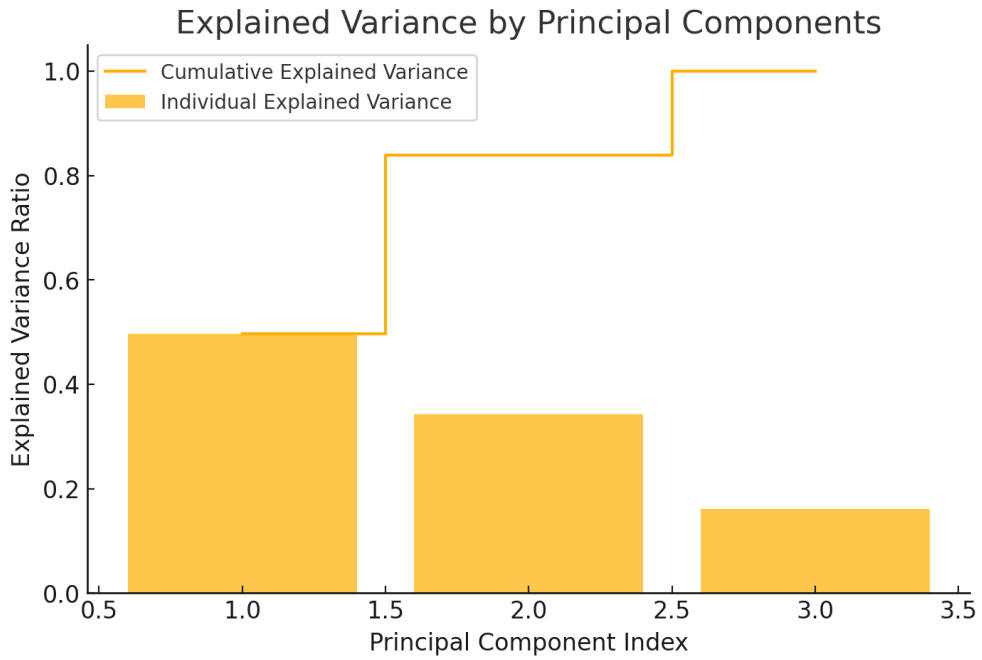
- $whiten$**:** Whether to whiten the components, making them uncorrelated and unit variance.

Journal: (Jolliffe & Cadima, 2016)

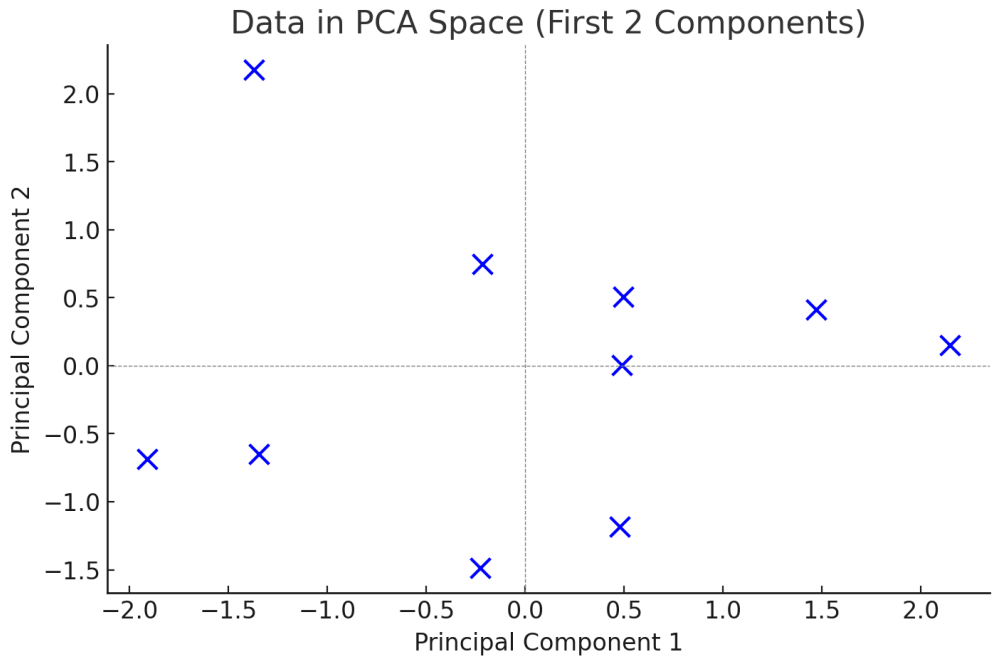
- Jolliffe, I. T., & Cadima, J. (2016). Principal Component Analysis: A Review and Recent Developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065).

Illustration:

We make up data for visualize as an example in Codes.ipynb



- Bar chart shows us the included sum of the variance, depending upon the principal component.
- Cumulative variation, by contrast, shows the cumulative amount of variance, which tells us how many points are needed to represent most of the variability in the data.



- The graph displays a set of observations in a 2D space whose axes are made from the two first principal components of the data. Each dot marks one case in the new, squeezed space.

- It helps one visualize which patterns or clusters are hidden within the data when the number of dimensions is decreased, that is, dimensionality reduction is being done.

Step 3 - Technical Section

Hyperparameter Tuning for LASSO Regression

Hyperparameter tuning in LASSO regression is mostly about choosing the regularization parameter, which will decide how much shrinkage is to be used in the coefficients.

1. Key hyperparameter:
    - α (Regularization Parameter)
        - Determines the bias-variance trade-off
        - High α increases the shrinkage factor and hence over-simplifies the model while increasing bias
        - Low α decreases the bias but may overfit.
2. Tuning Strategies:
    - Cross-Validation

      Split the data into training and validation sets, and test different α values to choose the one that gives the minimum validation error.

      Example: Perform k-fold cross-validation to test various values of α and then choose the one that gives the best MSE.

    - Grid Search:

      Define a range of values for α (such as [0.001,0.01,0.1,1,10]) and evaluate the model performance for each value to find the best parameter.

    - Model Interpretation:

      Look at the coefficients of the model to make sure that only meaningful predictors remain.

      Use plots to see how coefficients shrink as α increases.

3. Illustrative Example:
    - Do hyperparameter tuning using Python's LassoCV from scikit-learn, which does all cross-validation for LASSO.

Tuning of Hyperparameter in k Means Clustering algorithm

Hyperparameter are decided before the beginning of the learning process and they are very important for the performance of the model. The following are hyperparameters for k Means Clustering algorithm and their impact on model's performance has also been explained.

- n_clusters: It is the number of clusters in which data will be divided into. Choosing this optimally is important as it influences the clustering quality. We use the elbow method to find the optimal number of clusters.
- max_iter: It is the maximum iterations that algorithms run to find the optimal centroids. If it is larger than there may be better convergence but will increase cost also.
- init: While initializing the algorithm we need to specify the method for centroid initialization. For example, k-means++, random etc. This hyperparameter is used for that.
- tol: To find convergence in an algorithm defines the tolerance level.
- random_state: This hyperparameter is used for controlling the extent of randomness while initializing centroids. It Is also important for reproducibility.

Above hyperparameter can be tuned using following methods-

- Elbow Method--- Using this method we find the most optimal cluster numbers. The Elbow method does this by WCSS (sum of squares within clusters).
- The Silhouette method--- It is also used in finding the most optimal cluster numbers and also in measuring the cluster quality.
- Grid-search: It tries various hyperparameter combinations and finds optimal configuration.

Hyperparameter Tuning for PCA (Pedregosa et al., 2011)

In Principal Component Analysis (PCA), to tune the hyperparameters, the most important thing is to select the number of principal components that should be retained. Such a choice is essential for the proper balance between the reduction of the dimensionality of the given data and the keeping of the significant variance. Let's take a look at the most important hyperparameters and the ways they are tuned in PCA.

- **Key Hyperparameter: Number of Components (n_components)**

Description: This hyperparameter denotes the number of principal components to retain. It determines the dimensionality of the reduced dataset.

This hyperparameter specifies the number of principal components to retain. It determines the dimensionality of the reduced dataset.

Tuning Strategy:

- **Explained Variance Threshold:** The cumulative explained variance ratio is used to decide the optimal number of main attributes. The threshold of 90%-95% is a usual one ensuring, thus, most of the variance is kept. For instance, suppose the first 3 components together can make 92% of the variance disappear so you would be able to set n_components = 3 away from this process.
- **Cross-Validation:** It may be the case that n_components for this case may need some evaluations with some of the downstream models (e.g. classifiers) when changing the number of the components.

- **Other parameters**

1. svd_solver**:** Determines the algorithm for decomposition ($auto, full, randomized$)
    - Example: For large datasets, randomization is often applied to save time of computation.
2. $whiten$: If this is $True$, working with PCA for the inputs, it forces each principal component to have unit variance, thereby achieving decorrelation and standardization within the components.
    - Example: Mainly implemented in this case while PCA and machine learning are involved.

Step 4 - Marketing Alpha

Due to the current financial scenario that revolves around data, techniques such as machine learning will prove to emerge as a game-changer in alpha generation and better decision-making. With the use of tools - LASSO Regression, k-Means Clustering, Principal Component Analysis (PCA), investment managers, and quantitative strategists can uncover complex structures into intelligible patterns and make more accurate predictions. Here we have outlined the advantages and features of such techniques to show their importance in application in finance.

1. LASSO Regression: Shrinking Complexity for Robust Predictions.

LASSO refers to the Least Absolute Shrinkage and Selection Operator; this regularization thus aims at making models simpler and better by shrinking coefficients and implementing only the most important predictors.

Advantages:

- **Feature Selection:** LASSO reduces the noise in models with high-dimensional data by automatically selecting the most relevant features as it shrinks less important coefficients to zero.
- **Prevent Overfitting** through the addition of penalty to coefficients whose sizes are large: the end result is robust predictions with noisy financial datasets.
- **Interpretability:** The sparsity imposed by LASSO yields models which are interpretable despite having many input variables.

Applications in Finance:

- **Risk Assessment:** Identifying key factors influencing portfolio risk.

- **Return Prediction:** Selecting the most relevant macroeconomic and market indicators for forecasting asset prices.
- **Factor Models:** Improving factor-based alpha generation by eliminating redundant predictors.

What LASSO does is that it takes care of all forms of high dimensional datasets proficiently and in style; hence, its valued importance is realized when extracting actionable insights from complex finance data.

2. k -Means clustering: Discovering patterns in data.

K-means is an unsupervised learning model that divides the data into very different clusters based on the similarities, thus allowing a meaningful divide of financial data on its source.

Advantages:

- **Data segmentation:** k-means discovering the natural groupings of data like customer behavior, properties of stock, or market regimes.
- **Scalability:** The simplicity and computational Efficiency features of k-means make it quite superior for very large, real-time applications.
- **Pattern Recognition:** It reveals hidden structures in the data, which forms insights on relationships above what the conventional form of understanding does not capture.

Applications to Finance:

- **Customer Segmentation:** Grouping clients according to risk appetite or investment goals for a more personalized advisory service.
- **Portfolio Diversification:** Clustering assets with similar risk-return profiles such that diversification improves.
- **Market Regime Analysis:** The different states of the markets can be recognized for dynamic trading strategies.

The cluster and pattern digging of k-Means accompany strategic decision-making on the cluster and/or pattern driving into deeper understanding of the financial markets and customer behavior.

3. Principal Component Analysis (PCA): Reduction of Complexity with Maximum Variance.

PCA maps high-dimensional data into a smaller set of uncorrelated variables (principal components) which capture the maximum variance. This can be used as a very powerful dimensionality reduction and feature extraction mechanism.

Advantages:

- **Dimensionality Reduction:** reducing again the data, keeping only the most important components, taking redundancy out.

- **Noise Cancellation:** Exclusion of non-important features, thus more purified datasets for analyses.
- **Visualization:** Aids single intuitive visualization of high-dimensional data in two to three-dimensional spaces to uncover recognizable patterns and trends.

Applications to Finance:

- **Dimension Reduction Portfolio:** The reduction dimension is projected to improve optimization efficiency for correlated-asset portfolios.
- **Risk Management:** Identifying systemic risk factors that contribute to portfolio volatility.
- **Factor analysis:** The identification of potential underlying factors that drive asset returns for reliable alpha strategies.

PCA enables financial professionals to manage the complexity of modeling data while keeping the essential information, so it is imperative in high-dimensional financial environments.

Eventually, these three techniques of machine learning, each possessing unique strengths, combine together to develop an all-encompassing framework in financial decision making:

- Lasso Regression enhances predictive models by selecting the most relevant features.
- k-MEANS CLUSTERING helps for data that discovers patterns and groups, thus enabling strategic segmentation.
- PCA retains variability in such a way as dimensionality could be reduced and simplified for good insights from complex data sets.

These tools working together usually enable investment managers to address challenges such as portfolio optimization, market segmentation, and alpha generation with an unparalleled level of specificity and efficiency. This is how the integration of different machine learning methods supports decision making, increases productivity, and opens doors to new avenues in the fast-changing financial world.

Step 5 - Learn More: References on Machine Learning Algorithms

Below is a compilation of journal articles and authoritative resources that epitomize strengths in algorithms under machine learning, namely LASSO regression analysis, Elastic Net, Principal Component Analysis, and k-Means clustering. References that provide theoretical foundations, advance, and apply real-world scenarios, especially in the financial domain, are shown below.

1. Jolliffe, I. T., & Cadima, J. (2016). "Principal Component Analysis: A Review and Recent Developments." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065).
2. Clustering Algorithms in Finance: Jain, A. K. "Data clustering: 50 years beyond k-means." Pattern Recognition Letters, 31(8), 651-666, 2010.

3. Application of k-Means in Portfolio Management: Bholowalia, P., & Kumar, A. "EBK-means: A Clustering Technique based on Elbow Method and k-means in WSN." International Journal of Computer Applications, 105(9), 17-24, 2014.
4. k-Means and Customer Segmentation: Reddy, P. S., & Kumar, M. "Clustering Algorithms for Customer Segmentation in the Retail Industry." International Journal of Computer Science and Information Technologies, 7(3), 2016.

This comprehensive review covers the theory and modern developments of PCA & K means clustering emphasizing its applications in dimensionality reduction and feature extraction in fields such as finance and engineering and uses of K means clustering in various types of applications like customer segmentation etc.

References

- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267-288.
- Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456), 1348-1360.
- Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Methodological)*, 67(2), 301-320.
- Jolliffe, I. T., & Cadima, J. (2016). Principal Component Analysis: A Review and Recent Developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065).
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267-288.
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Jolliffe, I. T., & Cadima, J. (2016). Principal Component Analysis: A Review and Recent Developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065).
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267-288.
- A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis" – Future Generation Computer Systems, 2021.