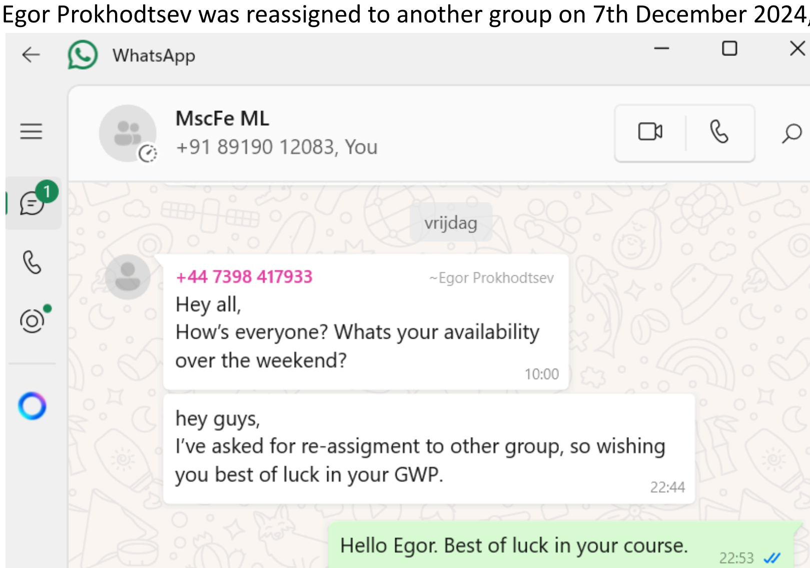
| FULL LEGAL NAME | LOCATION (COUNTRY) | EMAIL ADDRESS | MARK X FOR ANY NON-CONTRIBUTING MEMBER |
|---|---|---|---|
| Egor Prokhodtsev | United Kingdom | e.prokhodtsev@gmail.com | X |
| Regulavalasa Krishna Vamsi | India | krishnavamsi8262@gmail.com | |
| Enock Joseph Mwesigwa | Uganda | mwesigwaejoseph@gmail.com | |

**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** | **Enock Joseph Mwesigwa** |
| **Team member 2** | **Regulavalasa Krishna Vamsi** |
| **Team member 3** | |

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.

Egor Prokhodtsev was reassigned to another group on 7th December 2024, and so did not contribute to the GWP.



## Step 1

### Team Member A (Enock) - RIDGE Regression (Category 1)

**Basics**

Ridge Regression is a technique applied in linear regression problems where there is the issue of multicollinearity among the predictors or features, therefore causing overfitting of the linear regression model. Ridge regression regularizes the model by adding a penalty term to the cost function, shrinking the coefficients of correlated predictors to reduce their influence in the model.

**Keywords:**

Ridge Regression, regularization, multicollinearity, linear regression, penalty, overfitting, coefficients, bias-variance tradeoff, supervised learning

### Team Member B - Principal Components (Category 3)

**Basics**

PCA is a technique for dimensionality reduction. It transforms the correlated variables into a set of uncorrelated variables known as principal components. The PCA simplifies the datasets by retaining the maximum variance in fewer dimensions, aiding in visualization and noise reduction. It identifies the most important directions that explain the variance in the data, which are known as principal components.

**Keywords:**
Dimensionality reduction, PCA, eigenvectors, eigenvalues, covariance matrix, explained variance, loadings, feature extraction, data compression

## Step 2

### Team Member A (Enock) - RIDGE Regression (Category 1)

**Advantages of Ridge Regression**

1. Ridge regression addresses multicollinearity in financial data where there are usually many highly collinear predictors (such as stocks of competitor firms). It shrinks highly correlated predictor coefficients to reduce the effect of multicollinearity in the linear regression model.
2. Ridge regression preserves all features in the model without eliminating any, as usually occurs in the LASSO regression (where some coefficients are reduced to zero). This is important where every feature is required in the model.
3. Ridge regression reduces variance in the data series by introducing bias. This bias enables the linear regression model to perform better when dealing with different data by reducing the model's sensitivity to the noise in the data.
4. Ridge regression can improve the numerical stability of the linear regression model when the design matrix is not well-conditioned, and this enables the model to give more accurate predictions.
5. Ridge regression prevents model overfitting by reducing the complexity of the model through shrinking correlated feature coefficients. This enables the model to better fit the noise in the data.

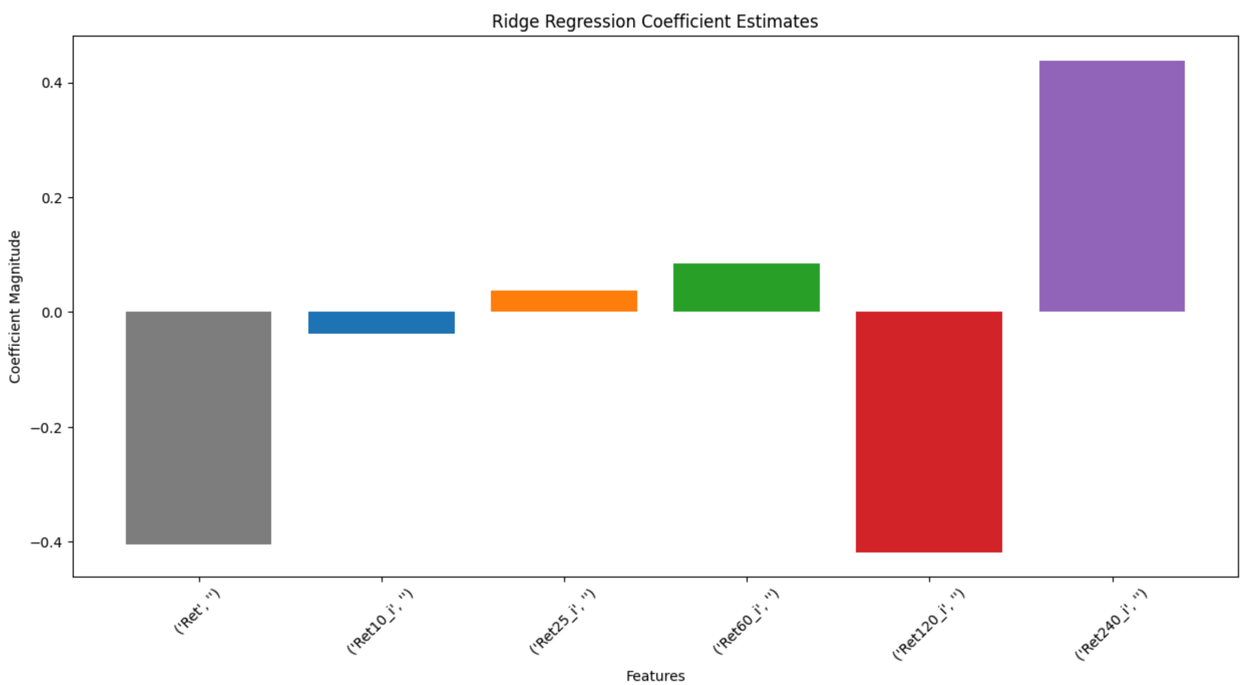
**Computation of Ridge Regression**

Attached in the Jupyter Notebook is code that shows the application of ridge regression. I use a dataset of daily financial returns of Amazon.com, Inc. (AMZN) obtained from Yahoo Finance spanning from January 2000 to November 2024.

I computed annualized rolling returns for time windows spanning 10, 25, 60, 120, and 240 days to obtain the performance of the stock in different time periods. I then used the rolling return columns as features and set the target variable as the 25-day forward return. I split the data into a training set and a test set using a 50:50 ratio. I trained a ridge regression model with the regularization factor set to 0.7, on the features. The model Mean Squared Error was 0.14 which indicates the accuracy of the model.

The regularised coefficients of the features are shown in the table and plot below.

| **Feature** | **Estimated coefficient** |
|---|---|
| Ret | -0.40 |
| Ret10_i | -0.04 |
| Ret25_i | 0.04 |
| Ret60_i | 0.08 |
| Ret120_i | -0.42 |
| Ret240_i | 0.43 |




**Explanation of the coefficients**:

- The daily return (Ret) coefficient is strongly negative meaning that higher daily returns are associated with a decrease in the predicted 25-day forward return.
- The 10-day rolling annualized return(Ret10_i) has a slightly negative coefficient, indicating a minor negative influence on the predicted 25-day forward return.

- The 25-day rolling return(Ret25_i) has a small positive coefficient. This indicates a weak direct relationship between the past 25-day return and the future 25-day return. This may be because it aligns closely with the prediction target and may contain overlapping data.
- The 60-day rolling annualized return (Ret60_i) shows a stronger positive coefficient, indicating that medium-term trends are somewhat predictive of the future 25-day return.
- The 120-day rolling annualized return(Ret120_i) has a strong negative coefficient, indicating a reversal effect at this time where stocks that have performed strongly over a longer period might experience pullbacks in the shorter term.
- The 240-day rolling annualized return (Ret240_i) has the strongest positive coefficient, indicating that long-term trends (around a year) are the most predictive of the 25-day forward return.

The above computations show that the ridge model effectively captures both short-term fluctuations and long-term trends in Amazon stock returns and may do better than a linear regression model that has no regularization term.

**Disadvantages of Ridge Regression**

1. Ridge regression has no feature selection ability and cannot eliminate irrelevant and redundant features from the model, hence not very useful in reducing the complexity and interpretability of models with many features. This is because while it penalizes the feature coefficients, it does not reduce them to zero, hence keeping all features in the model.
2. Ridge Regression assumes a linear relationship between the features and the target variable. In finance, where relationships are often nonlinear, this assumption may oversimplify complex dynamics, leading to failure to capture critical market behaviours, and therefore suboptimal predictions.
3. The performance of Ridge Regression depends a lot on the choice of the regularization parameter ($\alpha$). Choosing an inappropriate value for $\alpha$ could lead to over or under-penalization of the feature coefficients, leading to model underfitting or overfitting. Ridge regression requires hyperparameter tuning which adds to the model complexity.
4. Ridge Regression minimizes the squared error, which is highly sensitive to outliers. Extreme values in the predictor data can disproportionately affect the model. This is very relevant in finance where extreme phenomena are commonplace.
5. Ridge regression cannot handle missing values. For example, in the computation of the model, missing values had to be removed or handled before the model was trained on

the data. Financial data usually has missing data accounting for weekends and market holidays.

6. Ridge regression can over-shrink coefficients of features that are really important in the model, leading to underestimation of their effect on the target variable, hence suboptimal predictions.

**Equations for Ridge Regression**

Ridge regression adds a penalty term to the linear regression model's cost function, shrinking the large coefficients of the model features. The ridge regression penalty term is given as:

$$\alpha \sum_{i=1}^{p} \theta_i^2$$

Where:

- $\alpha$ is the hyperparameter that determines the extent of coefficient shrinking. An $\alpha$=0 negates the penalty term and the cost function remains for a regular linear regression. As $\alpha$ increases above zero, the penalization increases, with feature coefficients approaching zero as $\alpha$ approaches infinity.
- $\theta_i$ are the coefficients of the model features $i$ ($i = 1, 2, ..., p$). $i \neq 0$ to prevent distortion of the model intercept/ baseline.

The modified cost function of ridge regression becomes:

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\theta)^2 + \alpha\sum_{j=1}^{p}\theta_j^2$$

Where:

- $y_i$ is the observed target value at i-th observation
- $x_i$ is the predictor value for the i-th observation
- $x_i^T\theta$ is the predicted target value
- $n$ is the number of observations

The closed form value for the regularized coefficients is given by:

$$\theta = (X^TX + \alpha I)^{-1}X^Ty$$

Where:

- $X$ is the design matrix of the predictors
- $I$ is the identity matrix scaled by $\alpha$
- $y$ is the target variable vector.
- $X^T$ is the transpose of the design matrix

- $\hat{}(-1)$ indicates matrix inverse

In supervised machine learning, gradient descent can be applied to ridge regression to update the model parameters using the following expression:

$$\theta \leftarrow \theta - \eta(\frac{\partial}{\partial\theta}\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^{T}\theta)^2 + 2\alpha\theta \odot r)$$

Where:

- $\eta$ is learning rate
- $r$ is a column vector with $(\theta_1, \theta_2, ..., \theta_p)$ and $r_0 = 0$ (intercept is not regularized)
- $\odot$ is the operator for element-wise multiplication

**Features of Ridge Regression**

1. Ridge regression resolves the issue of multicollinearity among predictors in financial data by adding a penalty to the large coefficients, effectively making the model more reliable than regular linear regression even when correlated predictors are used.
2. Ridge regression retains all features in the model, and does not shrink any coefficient to zero, unlike the LASSO regression. This is important where all predictors are needed to make sense of the model outputs.
3. Ridge regression is more robust to outliers compared to the regular linear regression model because ridge regression reduces the influence of extreme values on the model predictions using its penalty term. This improves the reliability of the model when different datasets are used.
4. Ridge regression requires the tuning of its hyperparameter $\alpha$, as this parameter determines the extent of regularization. Tuning can be done through cross-validation with real test data.
5. Ridge regression can resolve overfitting models by reducing the noise in the data through shrinking feature coefficients.
6. Ridge regression does not change the model intercept and excludes it from the penalty term. This preserves the baseline of the target variable.

Ridge regression can be used in a closed form but also in gradient descent which enables it to work well in supervised machine learning methods to handle big datasets.

**Guide: Inputs and outputs**

**Inputs**

- Predictor variables: the independent features of the model, such as historical returns, interest rates, and volatility.
- Target variable: the dependent variable that is to be predicted by the model, such as future returns of stocks or portfolios.
- Preprocessing of data such as standardization, handling missing values,
- Regularization parameter alpha (α) controls the extent of shrinking that is applied to the feature coefficients.
- Learning rate (in case gradient descent is applied) to control step size during optimization of the coefficients.
- Data that is divided into training data and test data. Training data is used to fit the model, and test data is used for evaluating model performance.

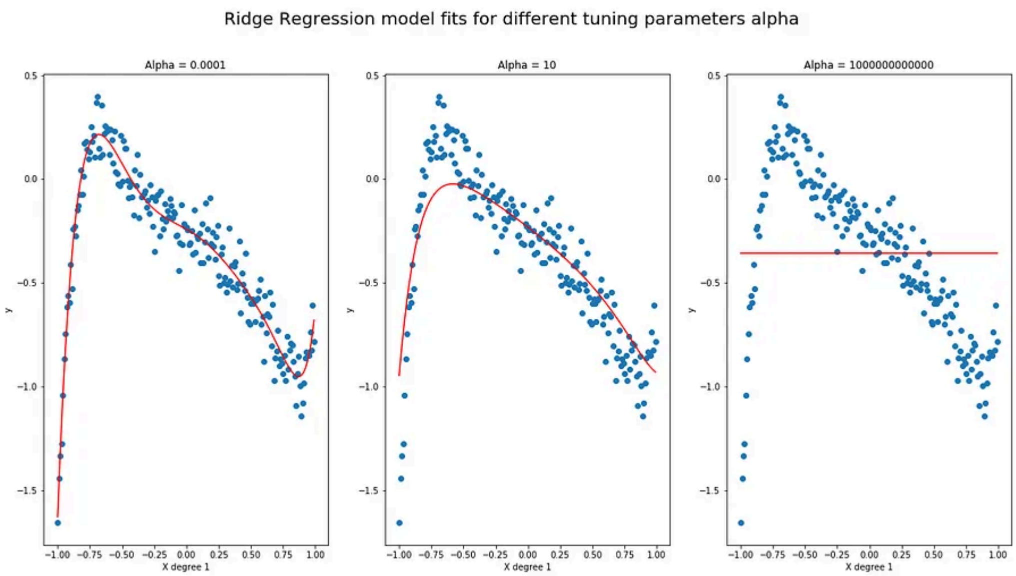
**Outputs**

- Estimates of feature coefficients representing the direction and importance of the feature in the model.
- Predicted values of the target variable such as forecasted stock returns
- Performance metrics of the model such as mean squared error (MSE), $R^2$, and mean absolute error (MAE).
- Tuned hyperparameters such as alpha (α), learning rate (η)
- Model intercept (baseline).

**Hyperparameters for Ridge Regression**

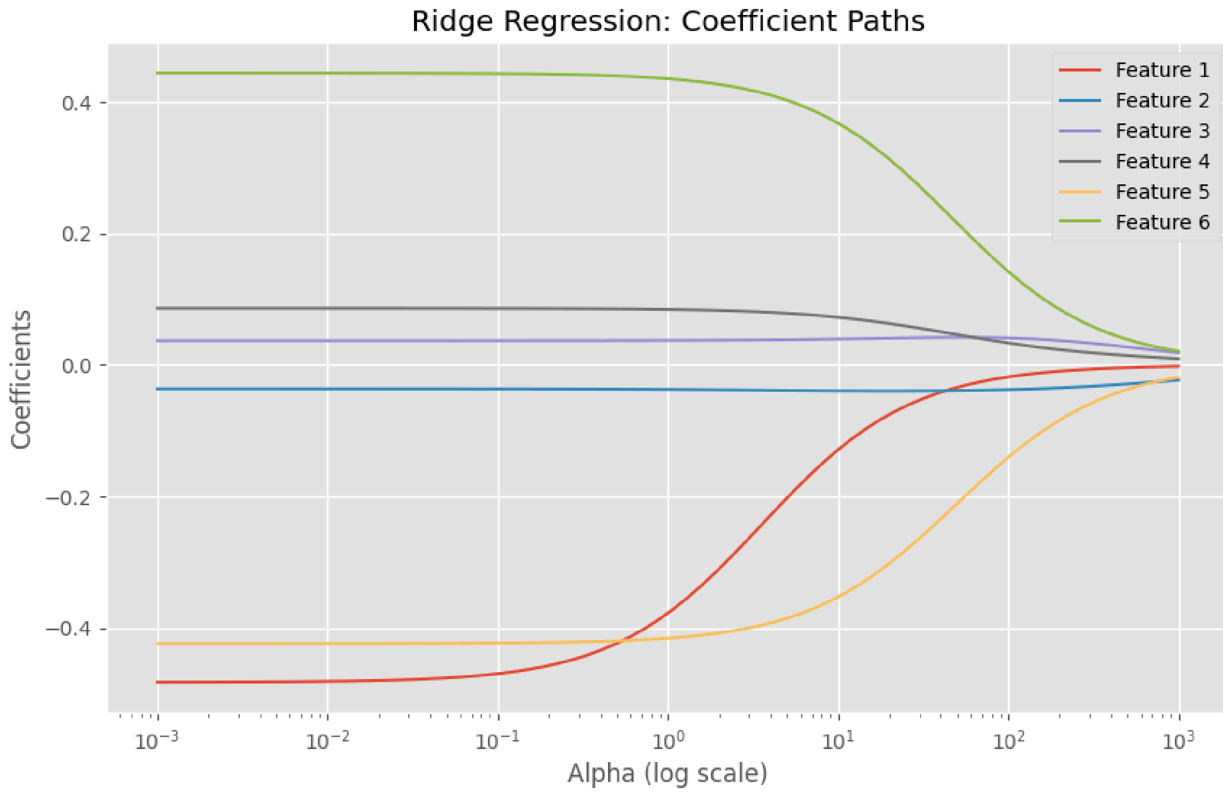
- Alpha (α) is the hyperparameter in ridge regression that regulates the extent of action of the penalty term on the feature coefficients. Its value can range from zero to positive infinity. The higher its value, the more stringent the action of the penalty term.

**Illustration of Ridge Regression**

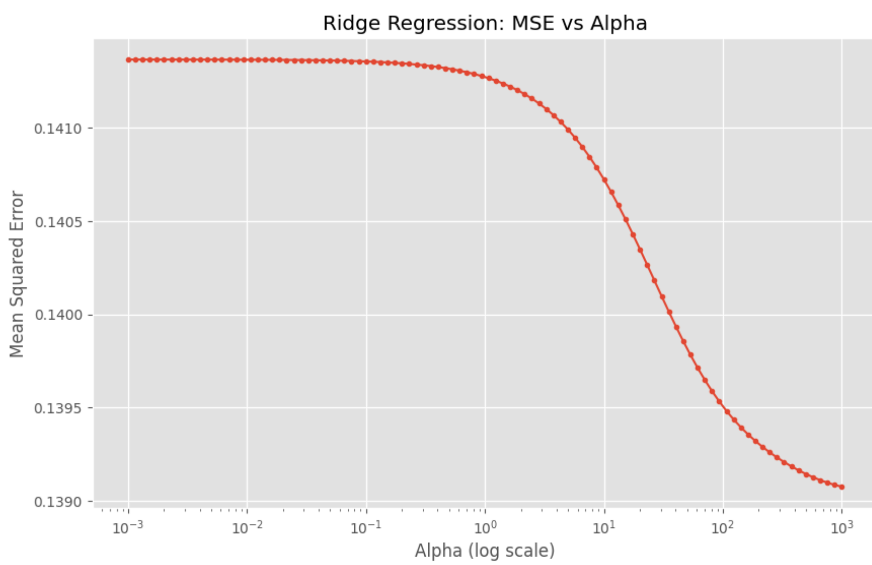



The above figure sourced from (Brooks) shows the effect of different alpha values on the performance of the model in fitting the data. Very low alphas leave the model overly sensitive to noise in the data and can lead to an overfitting model due to low bias and high variance. Very high values of alpha can lead to underfitting of the model due to high bias and minimal variance.
This figure illustrates the importance of tuning the hyperparameter alpha to the optimum value.

The action of the size of the alpha hyperparameter on the feature coefficients is well displayed in the figure below. As the alpha increases, high coefficients shrink even more, while low coefficients are minimally affected, showing how the ridge penalty shrinks large coefficients. This figure is derived from the dataset used in the computation section.




A linear regression model seeks to minimise the mean squared error (MSE). In the figure below, we see how increasing the alpha of the ridge regression can lead to a decrease in the MSE and potentially gaining more accurate predictions.



Although a low MSE is desired, a very low MSE can lead to underfitting the data. This also highlights the importance of tuning alpha to an optimum value.

**Journal showing the application of Ridge Regression in Finance**

The journal article " Modelling Stock Prices of Energy Sector using Supervised Machine Learning Techniques" written by (Benali and Karima) evaluates the application of various regression machine learning models, including ridge regression, to the prediction of stock prices of the energy sector in Morocco. The article concludes that ridge regression performed better than LASSO regression on the test data, showing potential applicability to predicting future oil stock prices. This implies its accuracy and reliability when applied to real financial data.

## Team Member B - Principal Components (Category 3)

**Advantages of Principal Components**

1. Feature Extraction: PCA transforms the dataset into a new feature set, reducing only the most important information relevant for the learning algorithm; therefore, it compresses data.
2. Increased Efficiency: PCA reduces the number of dimensions, which improves storage, speeds up computation, and enhances the model's performance by resolving issues associated with high-dimensional data.
3. Discovery of Latent Relationships: PCA can identify directions of maximal variance and project the data in a way that exposes latent correlations between the features.

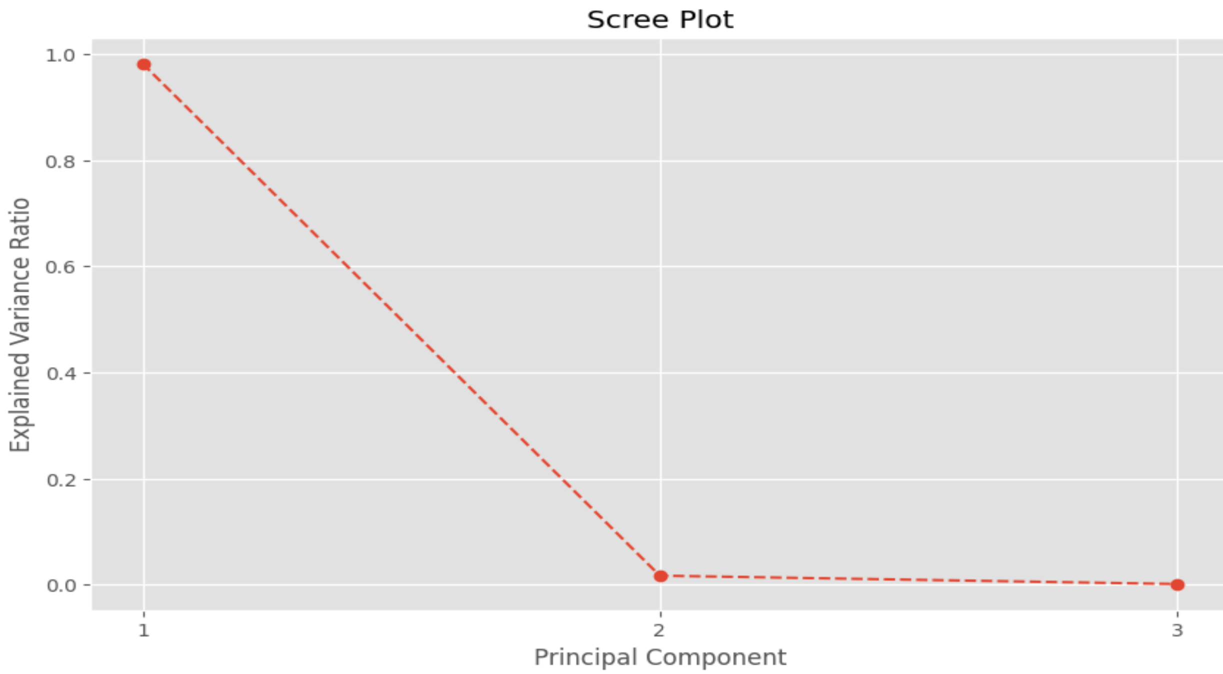
### Computation:

We used Principal Component Analysis on forex data for 10 major currencies against the US dollar to examine relationships. For computation, we used Scikit-learn, which has a very optimized implementation of PCA.

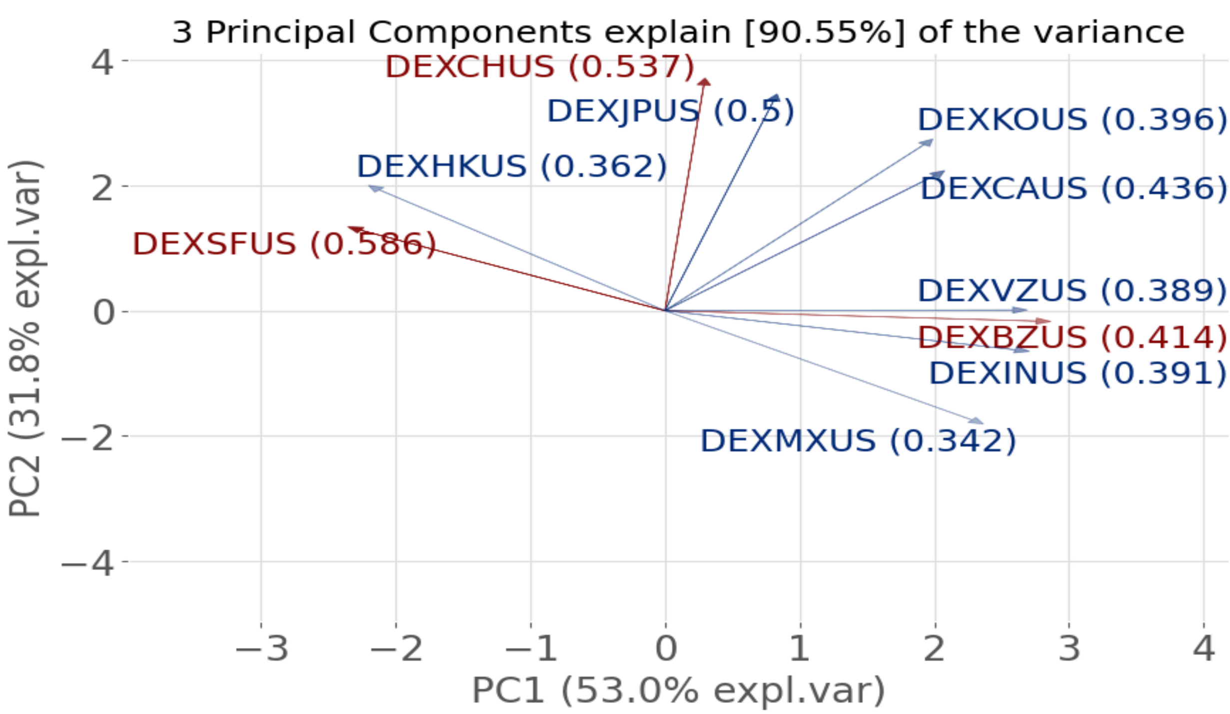
We then used a Scree plot to select the number of components by finding the point where the rate of explained variance drops rapidly.

Last but not least, we utilized a biplot for an intuitive interpretation of principal components and the contributions these might make to the data by clearly viewing relationships between the variables.

**SCREE PLOT:**




**BiPlot of Principal Component:**



### Disadvantages

1. Sensitive to Outliers: PCA is sensitive to outliers. The principal components are less effective when outliers distort the covariance matrix.
2. Information loss - This is a limitation of PCA as a dimensionality reduction technique. Important information is lost with the reduction in the number of components.
3. Sensitive: The PCA is unstable in nature; slight changes in data would cause significant differences between principal components.
4. Loss of Interpretable-ness: PCA reduces the original feature to principal components. Normally, the new components it results in are hard to relate with the original data.

### Equations:

- Standardization of Features: The dataset is standardized by making use of the StandardScaler. This transforms the data such that each feature will have a mean of 0 and a standard deviation of 1.

  **StandardScaler Formula**:

$$Z_i = \frac{x_i - \mu}{\sigma}$$

  Where:

  $Z_i$ = Standardized Value

  $x_i$ = Original value

  μ = Mean of the feature

  σ = Standard deviation of the feature

- Covariance Matrix: After standardizing the data, we calculate the covariance matrix.

  **Covariance Matrix Formula**:

$$\Sigma = \frac{1}{n-1} Z^T Z$$

  Where:

  $Z$ = Matrix of standardized data

  $n$ = Number of data points

- Eigenvalue and Eigenvector Calculation: To find the eigenvalue and eigenvector of the covariance matrix, we use the following equation:

$$\Sigma v = \lambda v$$

Where:

$\Sigma$ = Covariance matrix

$v$ = Eigenvector

λ = Eigenvalue

- The characteristic equation is:

$$det(\Sigma - \lambda I) = 0$$

Where:

$I$= Identity matrix

- Sorting of Eigenvalues and Eigenvectors: After finding eigenvalues and eigenvectors, we sort them in descending order and place their corresponding eigenvectors accordingly.
- Selection of Principal Component: The eigenvectors are principal components which correspond to the maximum eigenvalues. In most cases, we choose top k.
- k principal components that maximize the explained variance of the data.

### Features:

Feature Extraction PCA transforms the original features in the high-dimensional set into lower dimensions. The lower-dimensional changes are known as principal components. Data Compression Compression of the high-dimensional data without losing most of the variance in the process Data compression makes the visualization and analyses of the high-dimensional datasets much easier. Noise removal Focus on the principal component that captures most of the variance for the high-dimensional datasets.

1. Unsupervised Learning: PCA is unsupervised learning because it doesn't require any labeled data to be analyzed.
2. Multicollinearity Reduction: It reduces multicollinearity in high dimensional data by making orthogonal principal components.
3. Computational Efficiency: The process of PCA is computationally efficient because large datasets are simplified through a reduction in dimensionality.
4. Variance Capture: Using the first few principal components, PCA captures the most significant variance in data.

### Guide

**Inputs:**

- The Data Matrix Z: Here, n is the data points, and m is the number of variables or features.

- Number of Components, k: It is referring to the number of PCs that will be retained

**Outputs**

- Projected Data Z : The $n \times k$ matrix of projections of data onto the first k principal components.
- Eigenvector V : The $p \times k$ matrix of first k eigenvectors which are the principal components.
- Eigenvalue $\lambda$ : $k \times k$ diagonal matrix which contains the eigenvalues of the first k principal components.

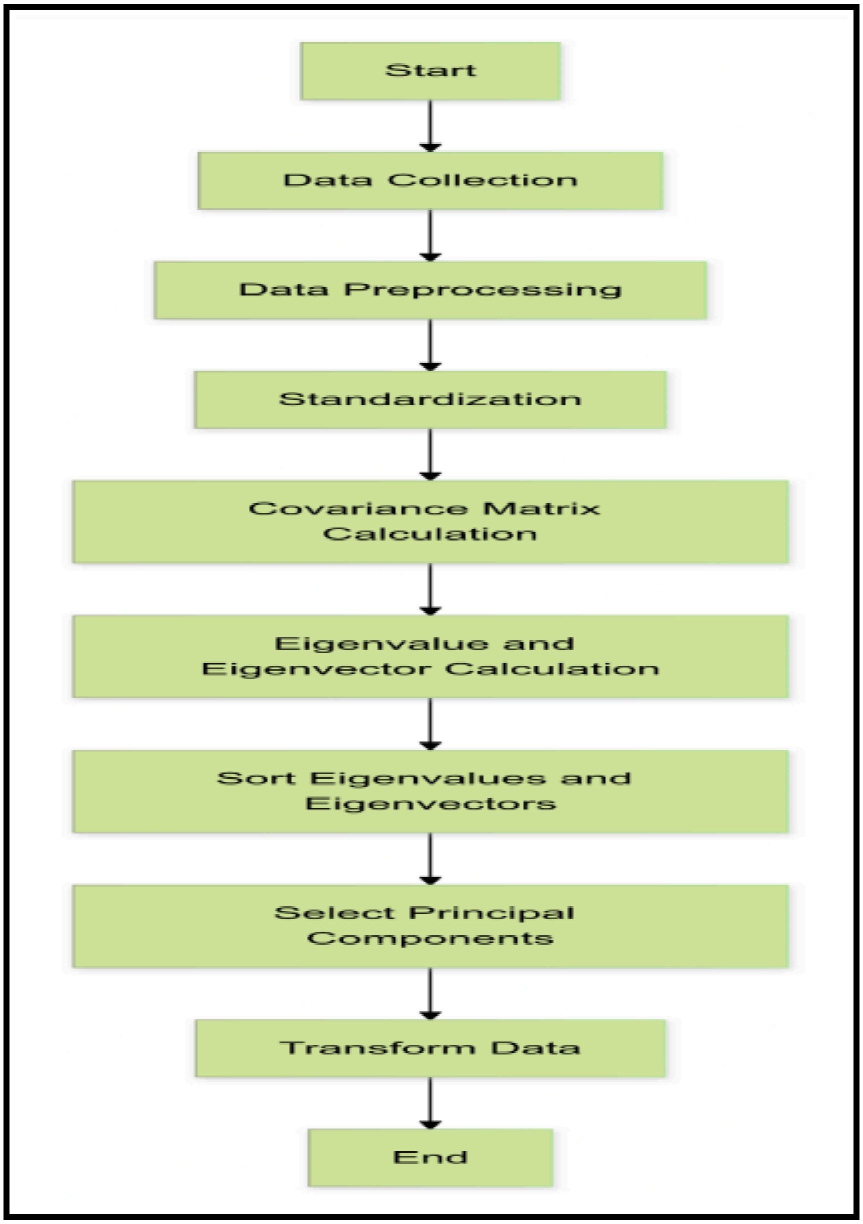
**Hyperparameters**

PCA is a deterministic approach and does not require many hyperparameter tuning in practice. There are, however, a few important parameters that need to be considered.

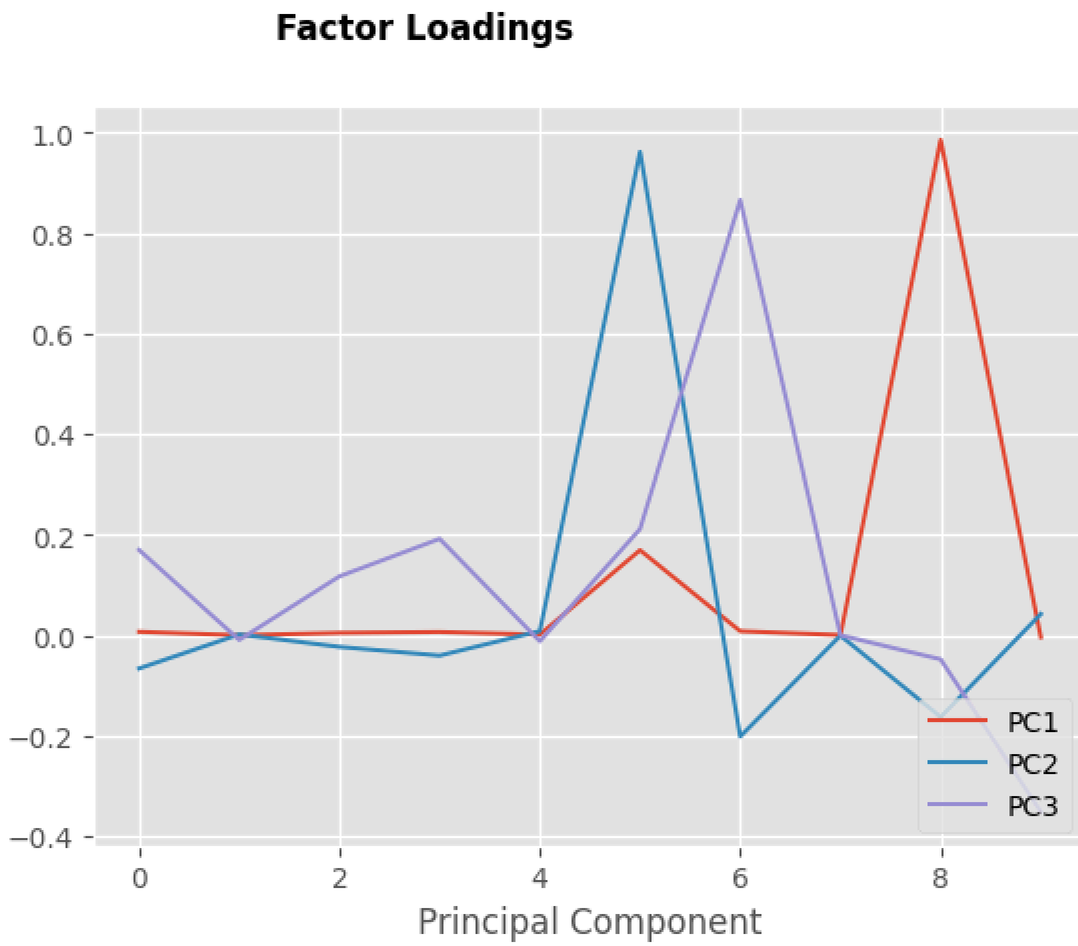
- Number of Principal Components k: The number of components to retain; this needs to be decided based on the desired reduction in dimensionality and amount of variance you want to retain.
- Scaling: Pre-PCA data standardization would be critical in ensuring equal contributions from all the features to improve the robustness of PCA.

**Illustrations:**

PCA Working Flowchart:


Start
Data Collection
Data Preprocessing
Standardization
Covariance Matrix Calculation
Eigenvalue and Eigenvector Calculation
Sort Eigenvalues and Eigenvectors
Select Principal Components
Transform Data
End

FactorLoading Graphs:




**Journal:**


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

## Step 3. Technical section

### Ridge Regression

A hyperparameter is a model parameter that is set before the instantiation of the machine-learning model. It then affects the behaviour of the model. The hyperparameter is not derived from the data itself and has to be chosen manually, or through automated tuning methods. Tuning the hyperparameter involves finding its value or values that lead to the best performance when used in the model. In the case of ridge regression, the hyperparameter is alpha () which controls the extent of shrinkage of feature coefficients. Tuning this hyperparameter involves finding its value that optimizes the balance between bias and variance to prevent the model from overfitting or underfitting the data. Various tuning methods are used.

**Grid search**

This is a tuning method that exhaustively explores the performance of the model using a specifically defined hyperparameter space in order to return the optimized parameter that optimizes model performance. For example, for ridge regression, we define a list of alpha values; grid search trains the model with every combination of the items in the list, evaluates the models' performance, and returns the alpha value that gives the best model performance (such as lowest Mean Squared Error). It is best used in cases of small search spaces.

**Pros:**

- It is simple to implement and understand.
- It is deterministic, with no change in results when the set-up is preserved.
- It evaluates every hyperparameter option

**Cons:**

- Slow and computer-intensive in case of large hyperparameter space.
- It assumes the hyperparameters in the given range are independent which maybe wrong.
- Entirely limited to the given hyperparameter space which may exclude better choices from evaluation.

**Random search**

This tuning method works similarly to grid search, with the difference being that it evaluates a fixed number of randomly chosen values from the hyperparameter space. It does not evaluate every combination, and so is faster than grid search, and can be applied to a wider hyperparameter space. It is best applied in cases of large search spaces.

**Pros:**

- Faster than grid search.

- Handles larger hyperparameter spaces than grid search.

**Cons:**
- It is not exhaustive and may miss the optimal combination through random sampling.
- Performance depends on number of random samples evaluated. Low number may give in optimal combination.
- Results may change with each iteration and a random seed may be required to maintain uniformity of results across runs.

**Bayesian optimization**

This method uses probabilistic models to predict the hyperparameters that are likely to perform well, basing the probabilistic score on past evaluations. Starting with a random sample of hyperparameters, a probabilistic model is used to predict performance across the hyperparameter space. The next hyperparameter is chosen based on its high probability to improve the model performance. It is best applied in cases where many evaluations are expensive, and the models are complex.

**Pros:**
- Requires fewer iterations to find the optimal parameters.
- Works well with both continuos and categorical hyperparameters.
- Learns from previous evaluations to improve next evaluation, increasing efficiency.

**Cons**:
- Higher complexity in implementation and comprehension.
- High computational costs in updating the probabilistic model.

**Cross-validation**

Cross-validation is used to evaluate the results of hyperparameter tuning. In this method, the data is split into k-folds. The model is trained on k-1 folds and validated on the remaining fold. This is repeated k times ensuring that each fold serves as a validation fold once. An average of the performance metric such as MSE is obtained.

**Pros:**
- More robust to overfitting or underfitting because it evaluates multiple data splits.
- Can work with any supervised learning model such as LASSO, Ridge, elastic net.

**Cons:**
- Computationally expensive due to multiple model evaluations.

- If data is not well balanced, the splits may give inaccurate results because of outlier effects.

To resolve its limitation with imbalanced data, repeated k-fold cross-validation can be used, where the splits are made multiple times to increase homogeneity and better representation of the data.

### PCA (Principal Components Analysis)

PCA is one of the popular dimensionality reduction techniques that transforms the original features into a set of uncorrelated variables called principal components. It is generally treated as a parameter-free model, meaning there are very few hyperparameters to adjust. However, two significant aspects can be tuned for better performance: number of principal components (n_components) and scaling.

Scaling: This is ensuring the data standardization, thereby making each feature to have a similar contribution to the principal components. This becomes very essential, especially in cases involving currency data. The value of a currency relative to the dollar may vary; hence scaling aids in consistency in analysis.

Number of Principal Components (n_components): In PCA, this is perhaps the most critical hyperparameter that dictates the dimensionality of transformed data. Typically, selecting the optimal number of components is done through methods of explained variance or scree plot analysis. In this case, we initially chose 5 components, and with the help of the scree plot, we filtered and chose the appropriate number based on the highest variance.

## Step 4. Marketing Alpha

### Ridge Regression

Ridge regression is very applicable to financial modeling and can be applied in optimising trading strategies. The model we used made an accurate forecast of the 25-day forward returns using returns over various time windows.

By handling multicollinearity without eliminating any feature, ridge regression provides a robust option to create accurate predictions while also capturing more market signals than LASSO which loses information through eliminating some features.

When the hyperparameter alpha is optimally tuned, the model can be adapted to changing market conditions quickly, because it involves only tuning the alpha.

Ridge regression is easy to understand in its way of operation making it more trustworthy to investors that want transparency of model operation. It is not a black box model.

Ridge regression can effectively address market noise through regularization. Reducing the noise extends the model's applicability to different datasets and reliable performance over time.

Ridge regression can also easily be combined with other machine learning methods such as LASSO, enabling synergy in the strength of the model.

Ridge regression is relatively cost-effective because it uses less computational power and has less complexity than many other deep machine learning models. Therefore it offers reliable financial predictions without high investment in computational resources.

## Principal Components Analysis (PCA)

Leverage the techniques of machine learning. PCA is an efficient way of handling high-dimensional data so that we can reduce the complexity of the data with maximum retention of key information. In our example, we were working with 10 currency exchange rates against the dollar, reducing the dimension to 5 components and retaining more than 90% of the relevant information. This not only accelerated the computation but also decreased the storage requirement.

PCA reduced our 10 columns of data into a set of 5 principal components (PCs), in which the data was compressed and computation was made more efficient, while retaining most of the variance. We used the most significant PCs while eliminating noise and ensuring that 3 PCs captured above 90% of variance.

As an unsupervised learning algorithm, PCA does not necessitate the use of labeled data and reduces multicollinearity in high-dimensional data sets. It also alleviates the curse of dimensionality and makes computations faster while preserving variance capture, more particularly in the first few PCs.

In our exploratory analysis of forex rates, the biplot indicates that PCs 1, 2, and 3 explain more than 90% of the variance, explaining general trends of different currencies against the dollar. Thus, for example, while the Japanese yen (DEXJPUS) is inversely correlated with PC1, it has been generally weakening against the dollar. The INR, MXN, BZ, and VZ currencies have clustered together, indicating general trends because they are emerging market currencies.

PCA also detected the stability of INR, CAD, and MXN against the USD, volatility and weakening trends of CNY and JPY, as well as stability and strengthening of CHF. By and large, PCA successfully became an efficient tool in understanding the general trends of different currencies, their volatilities, and stabilizations within high-dimensional data.

**Step 5. Learn more**

## Ridge regression


- Benali, Mimoun and Lahboub Karima. "Modelling Stock Prices of Energy Sector using Supervised Machine Learning Techniques." International Journal of Energy Economics and Policy 14.2 (2024): 594-602. <https://doi.org/10.32479/ijeep.15553>.
- Geraldo-Campos, Luis Alberto, Tamara Pando-Ezcurra and Juan J. Soria. "Machine Learning for Credit Risk in the Reactive Peru Program: A Comparison of the Lasso and Ridge Regression Models." Economies 10.8 (2022).
- Liu, Feng and David Pitt. "Application of bivariate negative binomial regression model in analysing insurance count data." Annals of Acturial Science 11.2 (2017): 390-411.
- Patwary, Muhammed J.A., et al. "Stock Price Prediction Using Semi-supervised Ridge Regression." Emerging Technologies in Computing. iCETiC 2022. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Ed. M.H. Miraz, et al. Springer, Cham, 2023. <https://doi-org.kuleuven.e-bronnen.be/10.1007/978-3-031-25161-0_9>.
- Safi, Samir K., et al. "Optimizing Linear Regression Models with Lasso and Ridge Regression: A Study on UAE Financial Behavior during COVID-19." Migration Letters 20.6 (2023): 139-153.
- Yi, Chae-Deug. "Machine Learning Prediction of Economic Effects of Busan's Strategic Industry through Ridge Regression and Lasso Regression." Journal of Korea Port Economic Association 37.1 (2021): 197-215. <https://koreascience.kr/article/JAKO202123261999200.page>.

**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*, vol. 31, 2014, pp. 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.

## References


1. Benali, Mimoun and Lahboub Karima. "Modelling Stock Prices of Energy Sector using Supervised Machine Learning Techniques." *International Journal of Energy Economics and Policy* 14.2 (2024): 594-602. <https://doi.org/10.32479/ijeep.15553>.

2. Brooks, Jake Miller. *How to Code Ridge Regression from Scratch A closed-form approach in Numpy*. 23 November 2020. 10 December 2024. <https://towardsdatascience.com/how-to-code-ridge-regression-from-scratch-4b3176e5837c>.

3. 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.

4. Jaadi, Zakaria. “Principal Component Analysis (PCA) Explained.” Built In, https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

5. “PCA — scikit-learn 1.5.2 Documentation.” Scikit-learn, https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.

6. “Principal Component Analysis.” Wikipedia, https://en.wikipedia.org/wiki/Principal_component_analysis.