Stock Price Prediction using Principal Component Analysis

Ujjwal Raj (ME22B1072)
Aman Gupta (ME22B1071)
Prashant Tyagi (ME22B1069)
Vinay Kumar (ME22B1070)

May 14, 2024

Problem Definition

Understanding Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of data while retaining most of its variance. It does this by transforming the data to a new coordinate system in which the greatest variance lies on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

Our Objective

Our objective in this project is to leverage PCA for predicting stock prices. Predicting stock prices is a crucial task in financial markets and can lead to significant financial gains if done accurately.

Motivation

Significance of Predicting Stock Trends

Even slight insights into stock movement can lead to significant financial gains, making accurate prediction highly desirable. However, predicting stock prices is a complex task due to the high volatility and uncertainty in the stock market.

Why PCA?

PCA is chosen due to its effectiveness in mitigating the curse of dimensionality often encountered in stock data analysis. The curse of dimensionality refers to the exponential increase in volume associated with adding extra dimensions to Euclidean space, which can lead to problems in machine learning and data analysis.

Dealing with Curse of Dimensionality

There are two ways to remove the curse of dimensionality

Feature Selection

It is the process which help us understand the most important feature.

Dimensionality Reduction

Dimensionality reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables. Techniques like PCA are used for this purpose. By reducing the dimensionality, we can improve model performance and can easily understand and visulize the data.

Feature Reduction Example: Housing Dataset

Analogy with Housing Features

Consider buying a house with numerous attributes such as proximity to amenities, number of rooms, size of the house, etc. Despite the abundance of features, the decision often boils down to a few critical factors such as location, price, and size. This is similar to feature reduction.

Application in Stock Prediction

In the context of stock prediction, PCA helps identify key factors amidst a multitude of variables, simplifying the prediction process. For example, instead of considering all the features of a stock such as its previous prices, volume, etc., we can use PCA to find a smaller set of features that capture the most variance in the data.Like PE Ratio, Inflation, Consumer Sentiment etc

Mathematical Method: Eigen Decomposition

Understanding Eigen Decomposition

Eigen Decomposition is a type of matrix decomposition that plays a pivotal role in PCA. It involves decomposing a square matrix into a set of eigenvectors and eigenvalues. The eigenvectors represent the directions of the new space, and the eigenvalues represent the magnitudes or lengths for the corresponding eigenvectors.

Linear Regression Insights

Concepts from linear regression, such as least squares fitting, can seamlessly integrate with PCA, contributing to accurate modeling of stock price trends. In particular, the residuals from a linear regression model can be used to calculate the variance that is captured by each principal component.

Mathematical Steps in PCA

Step 1: Standardize the Data

First, we standardize the data to have a mean of 0 and a standard deviation of 1. This is given by:

$$X_{std} = \frac{X - \mu}{\sigma}$$

where X is the original feature vector, μ is the mean of the feature vector, and σ is the standard deviation of the feature vector.

Step 2: Compute the Covariance Matrix

Next, we compute the covariance matrix of the standardized data. The covariance matrix is given by:

$$\Sigma = \frac{1}{n-1}(X_{std} - \mu)(X_{std} - \mu)^T$$

where n is the number of observations.

Mathematical Steps in PCA

Step 3: Compute the Eigenvalues and Eigenvectors

We then compute the eigenvalues and eigenvectors of the covariance matrix. The eigenvector v and eigenvalue λ of a matrix A satisfy the equation:

$$Av = \lambda v$$

The eigenvectors represent the directions or components for the reduced subspace of X_{std}, while the eigenvalues represent the magnitudes for the directions.

Step 4: Sort Eigenvalues and their Corresponding Eigenvectors

Next, we sort the eigenvalues in decreasing order and choose the first k eigenvectors, which correspond to the k largest eigenvalues, where k is the number of dimensions of the new feature subspace ($k \leq d$).

Mathematical Steps in PCA

Step 5: Transform the Original Dataset

Finally, we transform the original standardized dataset via the selected eigenvectors to obtain a k-dimensional feature subspace.

$$Y = X_{std} \cdot W$$

where Y is the transformed k-dimensional feature subspace and W is the matrix of the top k eigenvectors.

Finding Principal Components with Max Variance

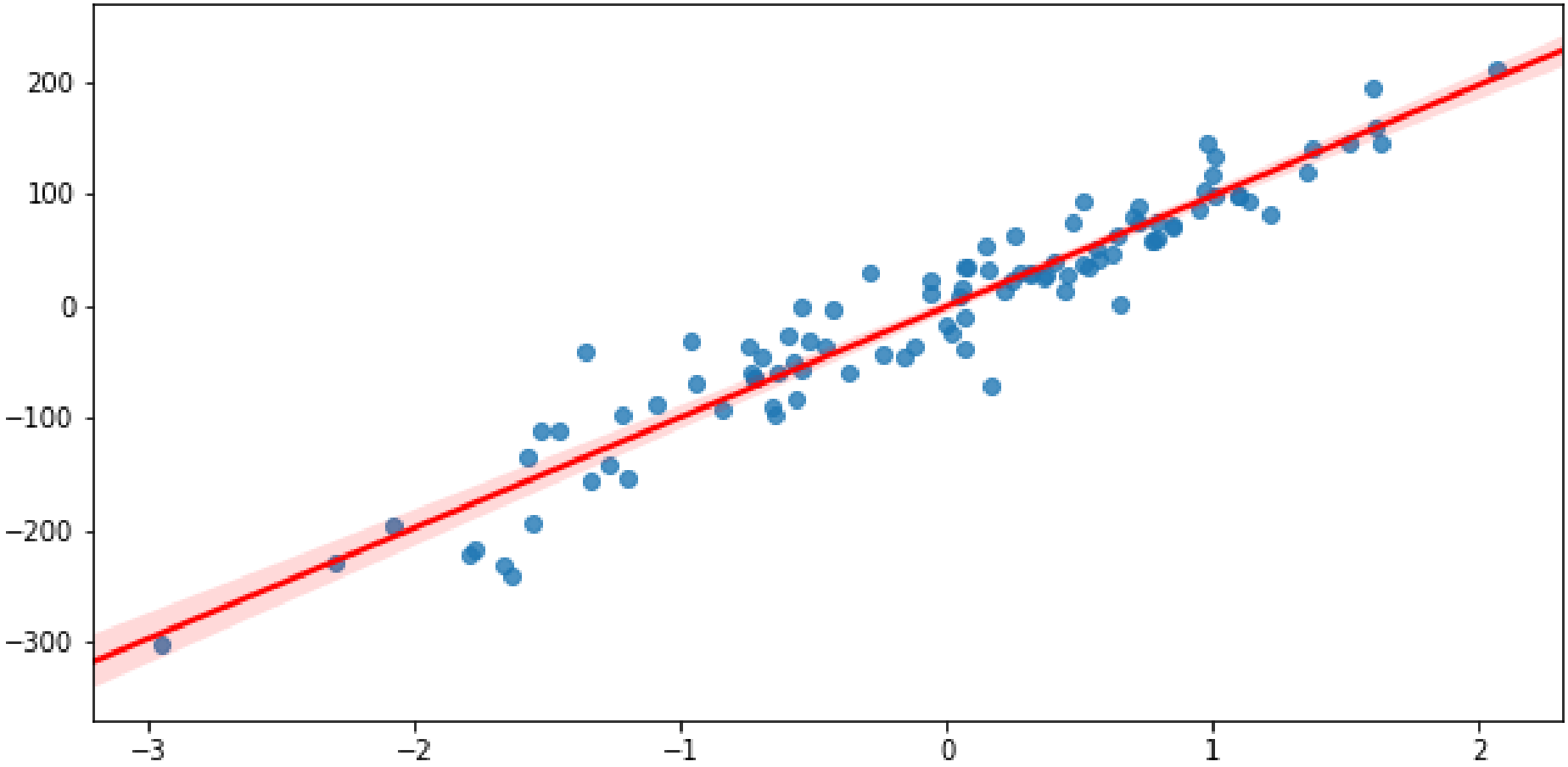
Utilizing Eigenvalues

Eigenvalues aid in identifying principal components with maximum variance, a crucial step in dimensionality reduction. The principal components are the eigenvectors of the covariance matrix of the data, and the magnitude of the corresponding eigenvalue indicates the amount of variance captured by that principal component.

Projection for Dimensionality Reduction

By projecting all components onto those with maximum variance, we effectively reduce dimensions while preserving the essential information. This is done by calculating the dot product of the data points with the eigenvectors.

Linear Regression Diagram



Stock Price Prediction with PCA Code

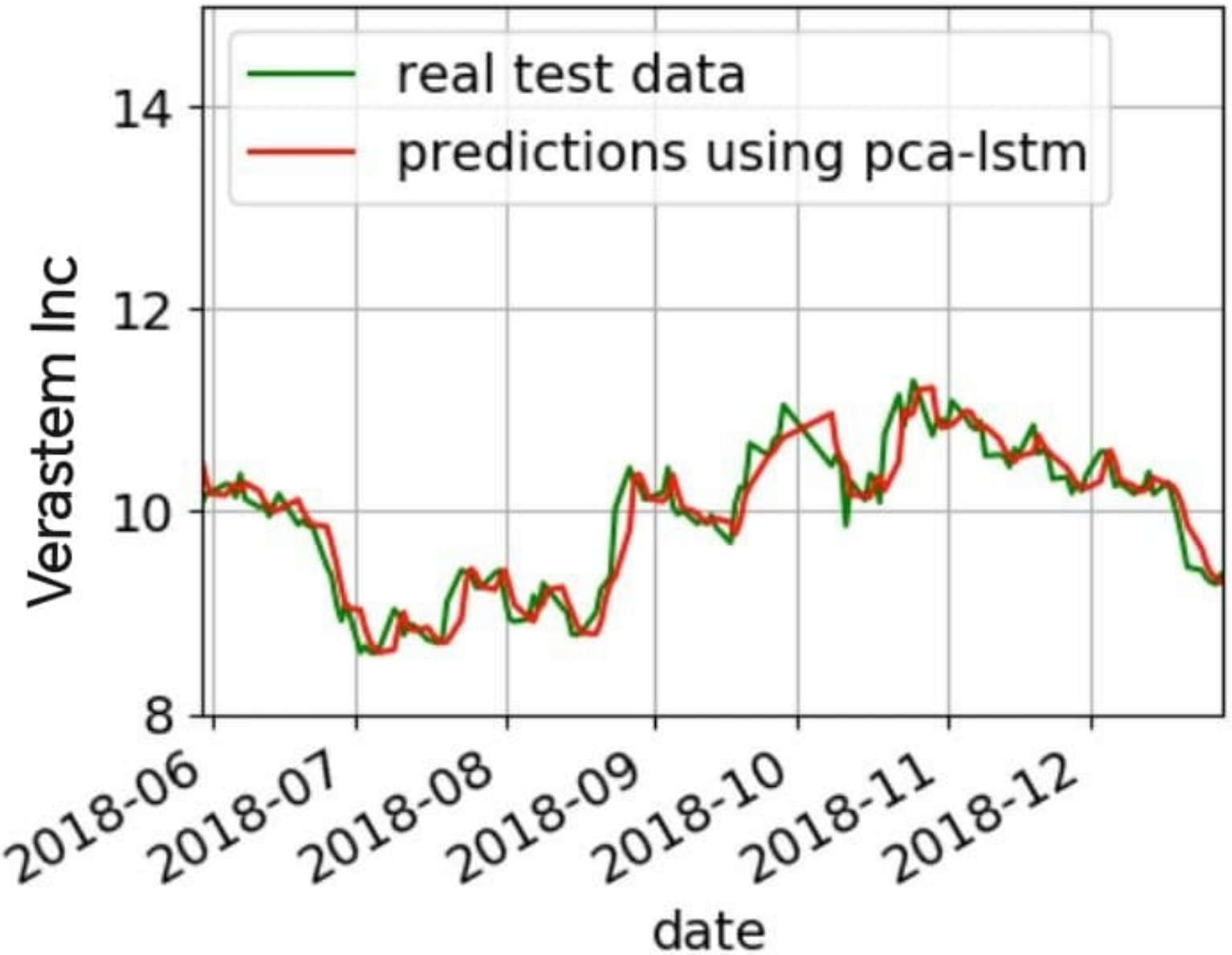
nm.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```python
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import yfinance as yf
import matplotlib.pyplot as plt

# Fetch historical stock data (example: Verastem Inc)
stock_symbol = 'VSTM'  # Change to your desired stock symbol
stock_data = yf.download(stock_symbol, start='2018-06-01', end='2018-12-31')

# Define the number of principal components to keep
n_components = 5

# Preprocessing: Extract relevant features and normalize
features = stock_data[['Open', 'High', 'Low', 'Close', 'Volume']].pct_change().dropna()
features = (features - features.mean()) / features.std()

# Apply PCA
pca = PCA(n_components=n_components)
pca.fit(features)
principal_components = pca.transform(features)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(principal_components, stock_data['Close'].shift(-1).dropna(), test_size=0.2, random_state=42)

# Train a linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
predictions = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, predictions)
print(f'Mean Squared Error: {mse}')

# Optional: Visualize actual vs. predicted values
plt.plot(y_test.index, y_test.values, label='Actual')
plt.plot(y_test.index, predictions, label='Predicted')
plt.xlabel('Date')
plt.ylabel('Stock Price')
plt.title(f'Stock Price Prediction for {stock_symbol} using PCA and Linear Regression')
plt.legend()
plt.show()
```

Stock Price Prediction with PCA Graph



Dataset Explanation

Data Preprocessing and Model Training

The code starts by fetching historical stock data for Verastem Inc(an American pharmaceutical company that develops medicines to treat certain cancers) using the yfinance library. It then preprocesses the data by calculating the percentage change for the 'Open', 'High', 'Low', 'Close', and 'Volume' features and standardizing these features to have a mean of 0 and a standard deviation of 1. The code then applies Principal Component Analysis (PCA) to reduce the dimensionality of the data, keeping only the top 5 principal components. The data is then split into training and testing sets, and a linear regression model is trained on the training set.

Stock Price Prediction and Evaluation

The trained model predicts stock prices. Its performance is evaluated using Mean Squared Error and visualized by plotting actual vs predicted prices.

Conclusion

Summarizing the Project

We've successfully explored and demonstrated the efficacy of PCA in predicting stock prices, highlighting its role in mitigating dimensionality issues. We believe that our work can serve as a foundation for future research in this area.

Thank You

We extend our gratitude for the opportunity to undertake this project and for your attention. We look forward to any feedback or questions you may have.