**Abstract:**

This paper presents a comparative study of three machine learning models—Linear Regression, Random Forest, and Support Vector Machine—for predicting stock returns using historical financial data, such as Share Price, Cash Flow, Book Value, and Earnings. The experiment design involves feature engineering, data normalization, and hyperparameter tuning to enhance the performance of the models. The models are evaluated using k-fold cross-validation to ensure their robustness and generalizability. The results indicate that the Random Forest model outperforms the other models in terms of accuracy and efficiency, demonstrating its suitability for capturing complex relationships and patterns in the data. This study contributes to a better understanding of the potential applications and limitations of machine learning in finance, ultimately benefiting investors and traders in making more informed decisions in the complex and volatile world of financial markets.

Keywords: machine learning, stock return prediction, Linear Regression, Random Forest, Support Vector Machine, financial data, feature engineering, data normalization, hyperparameter tuning, cross-validation

**Introduction:**

Predicting stock returns is a critical task in finance and investment management for both institutional and retail investors, as accurate forecasts can inform investment decisions and help optimize portfolio allocations. Machine learning techniques have gained popularity in recent years for their ability to model complex relationships in data and adapt to new information. This paper aims to compare the performance of three widely-used machine learning models—Linear Regression, Random Forest, and Support Vector Machine—in predicting stock returns using historical financial data, namely Cash Flow, Book Value, and Earnings. Additionally, this study explores the impact of incorporating alternative sources of data, such as news sentiment, technical indicators, and social media data, as well as the implications of different feature engineering techniques, data normalization methods, and hyperparameter tuning strategies on the performance of the models.

**Previous Work:**

A growing body of literature has explored the application of machine learning models for stock return prediction, focusing on various financial variables, techniques, and performance measures. For instance, Huang et al. (2005) used Support Vector Machines to forecast stock market movement [^1^], while Guresen et al. (2011) employed Artificial Neural Networks for stock market index prediction [^2^]. These studies have laid the foundation for further research into the application of machine learning models in finance, with an emphasis on improving accuracy and efficiency.

Several studies have investigated the importance of financial variables in stock return prediction. For example, Ou and Penman (1989) examined the usefulness of financial statement information in forecasting future stock returns and found that balance sheet variables, such as book value, provide significant information for predicting future returns [^3^]. In a more recent study, Piotroski (2000) demonstrated that a strategy based on selecting stocks with strong historical financial performance, as measured by several accounting-based variables, can generate significantly higher returns compared to a passive investment strategy [^4^].

In addition to traditional financial variables, researchers have also explored the impact of alternative data sources on stock return prediction. Tetlock (2007) investigated the role of news sentiment in predicting stock returns and found that negative news sentiment is associated with lower future stock returns [^5^]. Similarly, Bollen et al. (2011) analyzed the relationship between Twitter sentiment and stock market movements, discovering that social media sentiment can help predict future stock market changes [^6^].

This study builds on this body of literature by comparing the performance of multiple machine learning models and assessing the impact of feature engineering, data normalization, and hyperparameter tuning on their performance. Moreover, the study explores the potential benefits of incorporating alternative sources of data, such as news sentiment, technical indicators, and social media data, in improving the accuracy and efficiency of the stock return prediction models.

**Experiment Design:**

The experiment design can be broken down into several key steps:

1. \*\*Data Collection\*\*: Collect historical financial data for a sample of stocks, including Cash Flow, Book Value, and Earnings. Additionally, consider gathering alternative data sources, such as news sentiment, technical indicators, and social media data, to provide a more comprehensive view of the factors driving stock returns.

2. \*\*Data Preprocessing\*\*: Clean and preprocess the data to address missing values, outliers, and other data quality issues. This step is crucial in ensuring that the machine learning models can effectively learn from the data and make accurate predictions.

3. \*\*Feature Engineering\*\*: Create new features based on the existing financial variables, such as ratios and growth rates, to provide additional information for the models to learn. For example, calculate the Earnings-to-Book Value ratio, which has been shown in the literature to be associated with stock returns [^3^]. Incorporate alternative data sources, such as news sentiment, technical indicators, and social media data, to further enhance the models' ability to capture complex relationships and patterns in the data.

4. \*\*Data Normalization\*\*: Normalize the features to ensure that they are on the same scale, improving the models' convergence and overall performance. Different normalization techniques, such as min-max scaling and z-score normalization, can be explored to assess their impact on the performance of the models.

5. \*\*Model Selection and Hyperparameter Tuning\*\*: Select the three machine learning models—Linear Regression, Random Forest, and Support Vector Machine—and tune their hyperparameters using techniques like grid search and random search. This step is essential in optimizing the models and ensuring that they can effectively learn from the data and make accurate predictions.

6. \*\*Model Evaluation\*\*: Evaluate the performance of the models using k-fold cross-validation. This technique divides the dataset into k equal-sized folds, training the models on k-1 folds, and testing them on the remaining fold. This process is repeated k times, with each fold serving as the test set once. The advantage of using k-fold cross-validation is that it provides a more robust evaluation of the models' performance and generalizability, as it reduces the risk of overfitting and ensures that the models are tested on multiple subsets of the data.

7. \*\*Performance Metrics\*\*: Evaluate the performance of the models using three metrics: Mean Squared Error (MSE), R-squared, and the Cross-validation Score. These metrics provide different perspectives on the accuracy and efficiency of the models, enabling a comprehensive comparison of their performance.

8. \*\*Model Comparison\*\*: Compare the performance of the Linear Regression, Random Forest, and Support Vector Machine models based on the performance metrics obtained from the k-fold cross-validation process. This comparison allows for an assessment of the strengths and weaknesses of each model and helps identify the most suitable model for predicting stock returns using historical financial data.

Proposal:

A large portion of the work has been completed as of the writing of this document. Data collection has been developed using the Alpha Vantage API. The data for ~500 stocks has been cached using a script that interacts a client-side library that obtains and stores the data from the API. The necessary metrics are spread across 6 JSON files per stock symbol, each of which has a functioning, and fully tested parser and associated object for storage and retrieval in program memory. Effectively, data collection is complete.

Preprocessing has been defined, but loosely, the goal has been to take the JSON files and extract only the few metrics my models need to function. Storing this data as a CSV file containing the following information in a single row: Ticker, Fiscal Year, Cash Flow, Book Value, Earnings, and 5 Year Return. Each ticker may have multiple rows representing a single year. It has yet to be determined how this data needs to be adjusted during preprocessing to improve the model’s predictive abilities. Preprocessing is easy to modify, since each portion of the data, from each of the 6 json files, is well organized into a single object defined in Python, named “Stock.” Functions are then defined around this object to aid in the output to CSV.

A well-designed script is in development that is intended to handle items 3 through 7. These items are the core of the experiment. The script currently invokes the 3 models Linear Regression, Random Forest, and Support Vector Machine, and attempts to run data through them. Presently only one of the 3 models gives realistic answers, and the next course of action is hyperparameter tuning on the Support Vector Machine and Linear Regression models, until the system gives more realistic answers and has a better fit to the data. Improving the results for the 3 target metrics, Mean Squared Error (MSE), R-squared, and the Cross-validation Score, for each model is the goal from this point on.