**Project Report: Stock Market Prediction using Sentiment Analysis and Machine Learning**

**1. Introduction**

The goal of this project is to predict stock market movements based on sentiment analysis of financial discussions. The analysis utilizes natural language processing (NLP) to gauge sentiment from the messages, which are then used as features in a machine learning model to predict whether the stock will move up (positive movement) or down (negative movement). The project explores various methods for handling imbalanced data and overfitting, while evaluating the performance of the models using various metrics.

**2. Problem Statement**

Develop a machine learning model that predicts stock movements by scraping data from social media platforms like Twitter, Reddit, or Telegram. The model should extract insights from user-generated content, such as stock discussions, predictions, or sentiment analysis, and accurately forecast stock price trends.

**3. Data Description**

The dataset used in this project is discussion\_stock\_market\_with\_sentiment.csv, containing the following key columns:

* **Message**: The textual content of the message.
* **Sentiment**: The sentiment score associated with the message, ranging from negative to positive.
* **Stock\_Movement**: A binary target variable, where 1 represents positive movement and 0 represents negative movement in the stock price.

**Data Preprocessing**:

* Missing values in the **Sentiment** column were filled with neutral sentiment (0).
* The **Stock\_Movement** variable was generated based on sentiment thresholds (e.g., positive sentiment greater than 0.1 indicates upward movement, and negative sentiment less than -0.1 indicates downward movement).
* Neutral sentiment data was excluded, leaving only instances where sentiment had a clear impact on stock movement.

**4. Data Preprocessing and Feature Engineering**

* **Sentiment Handling**: The sentiment scores were adjusted based on predefined thresholds to categorize stock movements into positive and negative.
* **Message Length**: An additional feature was created to capture the length of the message, which could provide insight into the message's complexity or relevance.
* **Feature Set**: The features selected for the model were:
  + **Sentiment**: The sentiment score of the message.
  + **Message\_Length**: The length of the message.

**5. Model Selection and Evaluation**

**Model 1: Logistic Regression**

* Initially, a simpler model such as Logistic Regression was tested to provide a baseline performance.
* **Accuracy**: 0.50
* **Precision**: 0.50
* **Recall**: 1.00
* **F1 Score**: 0.67

**Model 2: Random Forest Classifier**

* Random Forests were tested with various hyperparameters to handle the imbalance and to improve generalization. However, issues with overfitting were observed, especially when handling the class imbalance.
* Performance metrics such as accuracy, precision, recall, and F1 score were used to evaluate performance.
* **Confusion Matrix**:

[[0 2]

[0 2]]

**Model 3: K-Nearest Neighbors (KNN)**

* Due to the small number of training samples after resampling, KNN models with a high number of neighbors led to errors (n\_neighbors = 6 exceeded the number of available samples).
* **Solution**: Reduced the number of neighbors to handle smaller datasets.

**Data Resampling**:

* **SMOTE (Synthetic Minority Over-sampling Technique)** was applied to balance the dataset, addressing the class imbalance.
* **Accuracy**: 0.50
* **Precision**: 0.50
* **Recall**: 1.00
* **F1 Score**: 0.67

Despite the improvement in balancing the data, overfitting remained an issue with more complex models.

**6. Overfitting and Underfitting Solutions**

To mitigate overfitting and underfitting issues, several strategies were tested:

* **Regularization**: Using models such as Logistic Regression with C parameter tuning to manage overfitting.
* **Cross-validation**: Implementing k-fold cross-validation to evaluate the models more effectively.
* **Hyperparameter Tuning**: Fine-tuning the parameters of Random Forest and other models to achieve a better balance between bias and variance.
* **Simpler Models**: Instead of overly complex models like Random Forest, simpler models such as Logistic Regression were preferred in some instances.

**7. Results and Discussion**

The model showed some challenges with overfitting, especially when balancing the dataset using resampling methods like SMOTE. The precision and recall scores were reasonably good, with a high recall indicating that the model was successfully identifying the positive stock movements but failing to predict negative movements accurately. This highlights a potential bias toward predicting positive movements.

**Key findings**:

* The **class imbalance** was one of the main issues leading to model overfitting. Resampling techniques like SMOTE helped balance the data but didn’t fully resolve the overfitting.
* The simpler models provided a baseline performance, which helped identify areas where more complex models were not generalizing well.

**8. Future Improvements**

* **Model Complexity**: Experiment with different models such as **Gradient Boosting** or **XGBoost**, which may perform better in handling the imbalance and overfitting.
* **Hyperparameter Optimization**: Apply a more rigorous hyperparameter tuning strategy using techniques like GridSearchCV or RandomizedSearchCV to identify the best parameters for the model.
* **Sentiment Analysis Refinements**: Enhance sentiment analysis methods by using more sophisticated models such as **BERT** or **Transformers** that could offer deeper insights into the textual data.
* **Incorporate More Features**: Include additional features such as past stock prices, volume, or other financial indicators, which could improve the prediction of stock movement.

**9. Conclusion**

This project successfully explored using sentiment analysis to predict stock movements, leveraging machine learning techniques to build a predictive model. While initial models provided decent results, challenges like class imbalance and overfitting were encountered. Further work in feature engineering, model selection, and hyperparameter optimization could improve the model's performance.

Despite the challenges, this project provides a foundation for using sentiment data in stock market prediction and can be expanded by incorporating more sophisticated models and features in future iterations.

**10. References**

* Scikit-learn documentation for machine learning algorithms and metrics.
* Imbalanced-learn documentation for resampling techniques like SMOTE.