Undergraduate Student Project Proposal:
Machine Learning for European Option Pricing
Project Overview

The goal of this project is to develop a machine learning-based system for pricing European options and estimating implied volatility using open-source data from OptionDX.com. The project will leverage state-of-the-art algorithms, including Random Forest, XGBoost, and Neural Networks, to analyze historical option data and predict market behavior.

Key Steps

1. **Data Collection**: Gather historical European option data from OptionDX.com (free), including strike price, expiration date, underlying asset price, and other relevant variables. SPX is a European style option instrument on the S&P500 Index.
2. **Model Development**: Implement three machine learning models:
    - Random Forest – tree-based model, useful for feature importance analysis.
    - XGBoost – ensembled model for efficient gradient boosting.
    - Neural Networks (e.g., Multi-Layer Perceptron) - for capturing complex non-linear patterns in option pricing.
3. **Model Evaluation**: Compare the performance of the models using metrics such as RMSE, R-squared, and computational time efficiency.
4. **Visualization**: Create visualizations to demonstrate model predictions versus actual implied volatility surfaces.
5. **Documentation & Deployment**: Document findings and deploy a user-friendly interface for real-world application.

Expected Outcomes

- A robust machine learning framework for European option pricing constructed in Python (Google Colab link).
- Insights into the relative performance of different algorithms for financial data analysis.
- A deployable tool for traders and risk managers to estimate implied volatility quickly.

Stretch Goals

- Compare the machine learning approaches to numerical methods for calculating implied volatility (Black-Scholes model). Is there a time advantage to using machine learning trained on this data compared to the numerical method? Is there a loss of accuracy?
- Compare the machine learning models trained on different sizes of training set. What is the loss or increase in accuracy due to training set size? What is the trade-off due to additional training time?