Initial Models Report

Team Lead: Hunter Huberdeau

Recorder: Amita Patil

Spokesperson: Eliel Polanco, Hayden Realmuto

**Recap Background & Question :**

Understanding and anticipating the investment decisions of influential figures has long been a subject of interest in financial markets. Warren Buffett, renowned for his disciplined value investing approach, focuses on identifying fundamentally strong yet undervalued companies (Hagstrom, 2013). Investors and financial professionals closely track his moves through SEC filings and media reports, but these sources provide insights only in retrospect. Team Lambda aspires to bridge that gap by leveraging machine learning to predict which stocks Buffett is likely to buy next, based on historical data and company-level financial indicators. By deploying predictive modeling and analysis, we aim to create a decision-support tool that empowers individual investors with insights traditionally available only after the fact.

The intersection of behavioral finance and predictive analytics presents a compelling opportunity to model Buffett’s investment patterns. While previous research has explored stock price prediction and portfolio performance modeling, few studies have specifically attempted to emulate the buy decision-making process of a known value investor using explainable machine learning models (Fischer & Krauss, 2018). Our approach is novel in that it does not merely assess market trends but seeks to identify the financial characteristics that make a stock appealing to Buffett before he purchases.

We hypothesize that historical financial metrics such as a low price-to-earnings (P/E) ratio, strong revenue growth, and high dividend yield are significant predictors of his stock selections. This is because Buffett prioritizes companies that exhibit financial strength, are undervalued relative to their intrinsic worth, and demonstrate steady, long-term growth potential (Buffett & Cunningham, 2020). If successful, we predict this work could enhance traditional financial analysis by providing predictive insights into high-profile investment strategies, offering a practical tool for investors who seek to align their decisions with Buffett’s time-tested approach.

**Methods:**

**Initial Model Selection:**

In the previous weeks, after conducting exploratory data analysis (EDA) and performing feature engineering, it became clear that class imbalance was a key challenge in this dataset. The **ElasticNet Logistic Regression** was chosen as the first model due to its effectiveness in handling both regularization and class imbalance. It offers a flexible combination of L1 and L2 regularization, which helps in preventing overfitting and ensuring that only the most relevant features are used. Additionally, **XGBoost** was selected as a powerful alternative due to its strong performance on imbalanced datasets and its capability to handle complex relationships in the data.

**Methods Used:**

* **Train/Test Split:** The data was split into training and testing sets using an 80/20 ratio, with stratification to ensure that the class distribution is preserved in both subsets.
* **SMOTE for Class Imbalance:** To handle the imbalance in the target variable, **SMOTE** was applied to the training data, which generated synthetic examples for the minority class.
* **Cross-Validation:** We used 5-fold cross-validation for model evaluation, ensuring that our model's performance is robust and not overly reliant on any single split of the data.
* **Logistic Regression and XGBoost:** We trained both ElasticNet Logistic Regression and XGBoost classifiers, with ElasticNet Logistic Regression using a balanced class weight and XGBoost incorporating scale\_pos\_weight to correct for the class imbalance.

**Model Assumptions:**

* **Logistic Regression Assumptions:** Logistic regression assumes that the relationship between the predictors and the log-odds of the target is linear. It also assumes that the predictors are independent of each other (no multicollinearity).
* **XGBoost Assumptions:** XGBoost, being a tree-based model, does not assume a linear relationship and can capture complex interactions between features. However, it assumes that the data is properly preprocessed .We did not explicitly test assumptions like linearity or independence for logistic regression, but the model was regularized (ElasticNet) to reduce potential issues from multicollinearity. For both models, no tests like Durbin-Watson for autocorrelation were applied, as the dataset does not involve time-series analysis.
* **Random Forest Assumptions:** Random Forest is a non-parametric, tree-based model that does not assume linearity or feature independence and is robust to multicollinearity, outliers, and non-linear relationships.

**Results:**

**1. Fitted Model:**

* **ElasticNet Logistic Regression Results:**
  + The **classification report** indicated that while the precision for Class 0 was very high (0.99), recall for Class 1 was extremely low (0.04). This reflects the model's bias toward predicting the majority class (Class 0), with minimal detection of Class 1.
  + **ROC AUC Score:** 0.48, which suggests poor discriminative ability between the two classes.
  + **PR AUC Score:** 0.01, indicating a very weak ability to detect the minority class.
* **XGBoost Results:**
  + **Classification Report** shows much better performance in detecting Class 1, with a **recall of 0.36** for Class 1 and a **precision of 0.41**. The model's performance on Class 0 is very high, with precision and recall near 1.
  + **ROC AUC Score:** 0.86, indicating a much better ability to distinguish between the two classes compared to logistic regression.
  + **PR AUC Score:** 0.27, which is also an improvement over logistic regression, reflecting the model's ability to better detect Class 1.

• **Random Forest Results:**

* **Precision** for Class 0: 0.9874, **Recall** for Class 0: 0.99832.
* **Recall** for Class 1 was still low at 0.085, **Precision** for Class 1 improved to 0.08900.
* **ROC AUC**: 0.86 (similar to XGBoost, showing good overall ranking capability).
* **PR AUC**: 0.1731 (still low, suggesting the model struggles with Class 1 detection).

**2. Assumptions :**

* **Logistic Regression:** While we didn’t explicitly check for multicollinearity or normality of residuals, we utilized ElasticNet regularization to mitigate potential issues like multicollinearity and overfitting.
* **XGBoost:** XGBoost does not have the same assumptions as logistic regression (e.g., linearity), and thus, was more adaptable to the data. The use of **scale\_pos\_weight** helped manage the class imbalance effectively.
* **Random Forest** is a non-parametric tree-based model, robust to multicollinearity and non-linearities.

**3. Overfitting:**

* Both models were evaluated on a **train/test split** to check for overfitting. The **5-fold cross-validation** for logistic regression also helped assess the model's generalizability.
* **ElasticNet Logistic Regression:** The model showed signs of overfitting due to its poor recall for the minority class. However, regularization helped limit overfitting in the training set.
* **XGBoost:** While the model performed well overall, it still showed some room for improvement in detecting Class 1. Further hyperparameter tuning could improve its ability to capture the minority class without overfitting.

**Visualizations :**

A graph of a bar

AI-generated content may be incorrect.

Figure 1:Class Imbalance

A graph of a class distribution

AI-generated content may be incorrect.

Figure 2:Balance the Training Data with SMOTE

* **Logistic Regression**

A diagram of a confusion matrix

AI-generated content may be incorrect.

Figure 3:Logistic Regression Confusion Matrix

* 1. ROC curve   
     A graph of a positive rate

     AI-generated content may be incorrect.

Figure 4: Logistic Regression ROC curve

* **XGBoost**

A graph of a diagram

AI-generated content may be incorrect.

Figure 5: XG Boost Confusion Matrix

A graph of a positive rate

AI-generated content may be incorrect.

Figure 6: XGBoost ROC Curve

Random forest with Smote  
A screenshot of a computer

AI-generated content may be incorrect.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Random Forest Model Results (After SMOTE Oversampling):**

The confusion matrix indicates that the model correctly classified 7,732 of the "No Buy" cases and 8 of the "Buy" cases. However, it misclassified 86 actual "Buy" cases as "No Buy," and 13 "No Buy" cases were predicted as "Buy."

**Performance metrics:**

* **Accuracy:** 0.9874
* **95% Confidence Interval:** (0.9846, 0.9897)
* **No Information Rate:** 0.988
* **P-Value [Acc > NIR]:** 0.7197
* **Kappa:** 0.1353
* **McNemar’s Test P-Value:** 4.612e-13
* **Sensitivity (Recall):** 0.99832
* **Specificity:** 0.08511
* **Precision (PPV):** 0.08900
* **Negative Predictive Value:** 0.38095
* **Balanced Accuracy:** 0.54171

Despite high overall accuracy, the model still struggles with correctly classifying the minority "Buy" class. The low specificity and precision suggest that many actual "Buy" cases are still being missed, though there is improvement over earlier models. The Balanced Accuracy (54.17%) and Kappa (13.5%) reflect this moderate enhancement.

A graph of a curve

AI-generated content may be incorrect.

Figure 7:Random forest ROC Curve

**ROC AUC: 0.8698**

* **Definition:** ROC AUC (Receiver Operating Characteristic - Area Under the Curve) measures a model's ability to distinguish between classes.
* **Interpretation:** A value of 0.8698 means the model correctly distinguishes between the two classes (e.g., "buy" vs. "not buy") nearly 87% of the time.
* **Scale:**
  + 0.5 = No discriminative power (random guessing)
  + 1.0 = Perfect separation
* **Assessment:**
  + This is a **very good result**, indicating strong overall classification performance across all thresholds.

**2. PR AUC: 0.1731**

* **Definition:** Precision-Recall AUC focuses specifically on the model's performance in identifying the positive class, particularly useful for imbalanced datasets.
* **Interpretation:** A score of 0.1731 indicates the model has difficulty identifying positive cases (likely "buys"), which may lead to many false positives or missed true positives.
* **Scale:**
  + 0 = Worst
  + 1 = Perfect prediction of positives
* **Assessment:**
  + This is a **low score**, suggesting that despite good ROC AUC, the model is **not reliable** for predicting the minority class (positives) accurately.

**Random Forest Results Summary (After SMOTE Oversampling)**

**Confusion Matrix**

A blue and red squares with white text

AI-generated content may be incorrect.

Figure 8:Random forest -confusion Matrix

A graph and chart with text

AI-generated content may be incorrect.

Figure 9:Random forest Feature Importance

A graph with lines and text

AI-generated content may be incorrect.

Figure 10:Partial Dependence Plot – ROE (Return on Equity)

This plot shows a **Partial Dependence Plot (PDP)** for the variable **ROE (Return on Equity)**, illustrating how ROE affects the predicted outcome (yhat) of your model — in this case, likely from a Random Forest.

**Interpretation of the Plot:**

* **X-axis (ROE):** Represents the range of values for Return on Equity in your dataset.
* **Y-axis (yhat):** Represents the predicted probability (or model prediction) for the target class, averaged over all other features.
* **Pattern Insight:**
  + The plot shows a **sharp spike around ROE ≈ 0**, where predictions increase significantly.
  + For very **negative ROE values**, predictions remain low, which suggests these companies are less likely to be classified as a "buy."
  + **Higher ROE** past that spike sees a drop and stabilization in predictions, which may indicate diminishing returns in model influence after a threshold.

**What This Means:**

* The model assigns **higher probability of a buy classification** when ROE is slightly above zero.
* **Extremely negative or very high ROE values** do not strongly drive predictions — they are either penalized or plateaued in influence.

**Key Takeaways**

* **SMOTE** helped balance the training set but was not sufficient alone for minority class detection.
* **XGBoost** outperformed ElasticNet Logistic Regression significantly in detecting potential “Buy” candidates.
* **Random Forest** showed solid ROC AUC (0.87) but still lacked practical utility due to very low recall and precision for Class 1.
* Logistic Regression struggles most with imbalanced class predictions.
* Feature importance (Random Forest/XGBoost) and PDPs (e.g., ROE) offer insights into the traits influencing Buffett-style investments.
* Further **threshold tuning**, **ensemble methods**, and **cost-sensitive learning** may be beneficial.

**Next Steps**

* **Hyperparameter tuning** for XGBoost and Random Forest to improve Class 1 detection.
* **Threshold tuning** to find a better balance between precision and recall for the minority class.
* **SHAP analysis** for deeper explainability—especially to highlight what drives predictions for Class 1 (likely "Buys").