Insurance net 2024

Lecturer: Dr. Chen Hajaj

Team members:

Tomer Sabag:318814977

Itamar Melnik:207312307

Repository GitHub link:

https://github.com/Itamar-Melnik/ML2

Table of contents:

Introduction

The focus of this project is on addressing a critical challenge in the financial industry: determining which among the numerous insurance funds available are likely to succeed or fail. With the growing diversity and volume of financial data, it becomes increasingly difficult for stakeholders to identify high-performing funds. To tackle this, the project employs machine learning techniques, specifically classification and clustering, to analyze and evaluate the performance of insurance funds. The project aims to provide a data-driven solution to predict the success of insurance funds by leveraging their financial and performance data. This enables stakeholders to make informed investment decisions and optimize resource allocation. This is achieved by employing supervised learning for classification and unsupervised learning for clustering. The primary objective was to classify funds based on the Alpha measure—a metric for evaluating fund performance—to distinguish high-performing from underperforming funds. Additionally, the project explores clustering to uncover hidden patterns and groupings within the data, providing a deeper understanding of fund characteristics. This analysis helps uncover insights into fund behavior and provides actionable knowledge for stakeholders to make informed decisions.

Formula for measure Alpha: $Alpha = Ri - (Rf + \beta \cdot (Rm - Rf))$

Explanation:

- Ri: Actual return of the insurance fund.
- Rf: Risk-free rate of return (e.g., return on government bonds).
- β: Beta coefficient, representing the fund's sensitivity to overall market movements.
- Rm: Return of the market benchmark (stock market index).

Dataset&features

The dataset was sourced from the governmental database, providing comprehensive financial and performance metrics of insurance funds in 2024. It includes features such as yield metrics, management fees, market exposures, and liquidity ratios, offering a full-bodied foundation for analysis. The dataset was prepared through extensive preprocessing steps tailored separately for classification and clustering. Below, we outline the specific steps and reasoning for feature selection based on column analysis.

Preprocessing Steps:

The preprocessing steps for classification and clustering shared several similarities but were tailored to the unique requirements of each task. The description below unifies both the commonalities and the differences.

Feature Selection:

We used Predictive Power Score (PPS) and Spearman correlation matrices to identify multicollinearity and refine feature selection. PPS helped detect predictive relationships and highlighted redundant features, while Spearman correlation focused on linear dependencies to remove multicollinearity. Spearman correlation, on the other hand, highlighted linear relationships and helped in removing redundant or highly correlated features.

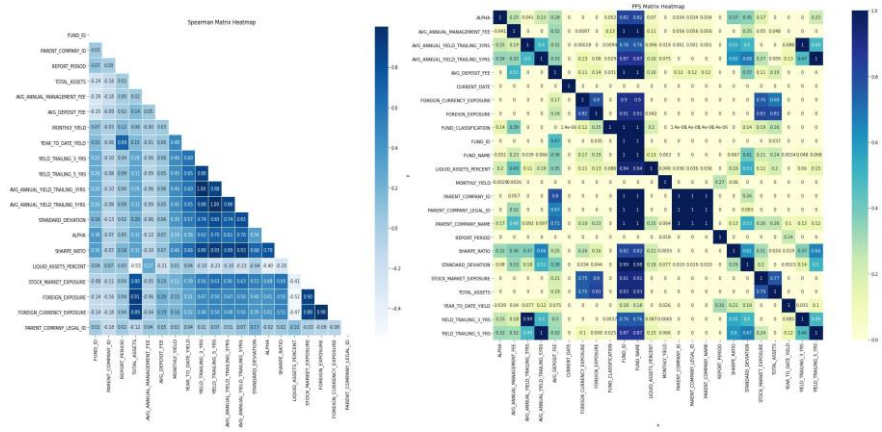
FUND_NAME, PARENT_COMPANY_ID, PARENT_COMPANY_NAME, FUND_CLASSIFICATION, and *PARENT_COMPANY_LEGAL_ID*: These columns had perfect multicollinearity with FUND_ID and were therefore redundant. (PPS)

AVG_ANNUAL_YIELD_TRAILING_3YRS and *AVG_ANNUAL_YIELD_TRAILING_5YRS*: These were perfectly explained by *YIELD_TRAILING_3_YRS* and *YIELD_TRAILING_5_YRS* respectively, and thus removed. (PPS)

FOREIGN_EXPOSURE: This column was highly correlated (98%) with *FOREIGN_CURRENCY_EXPOSURE* and removed to reduce redundancy. (Spearman)

FUND_ID: Removed after imputing missing values to prevent classification models from biasing predictions based on company identifiers instead of financial data.

For clustering, the ALPHA column was retained as a feature, while for classification, it was used as the target variable and rows with missing values in ALPHA were removed.



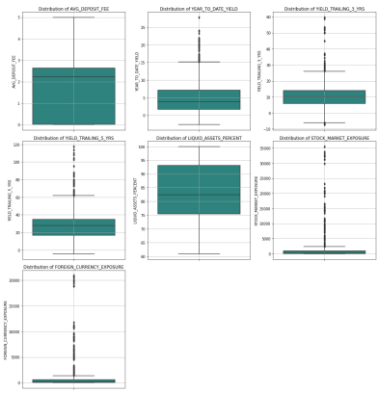
Handling Missing Data:

To handle missing values, we first filled them using the mean of each company (*FUND_ID*) to preserve company-specific trends. Afterward, for the remaining missing values, we applied imputation based on the data's distribution:

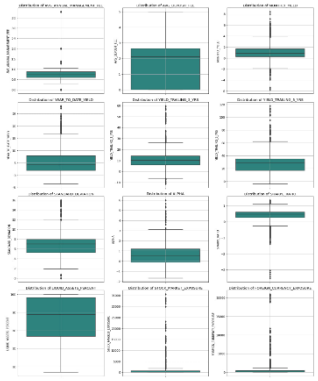
Median Imputation: Applied to columns with outliers or asymmetrical distributions, such as yield metrics and *SHARPE_RATIO*.

Mean Imputation: Used for symmetrically distributed columns like *LIQUID_ASSETS_PERCENT* and *AVG_ANNUAL_MANAGEMENT_FEE*.

Classification:



Clustering:



Scaling and Encoding:

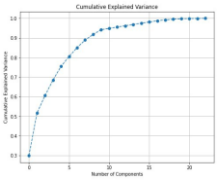
Scaling and encoding were key preprocessing steps in this project:

Scaling: We applied StandardScaler in both tasks to standardize numerical features, ensuring that all variables were adjusted to a common scale suitable for effective analysis.

One-hot Encoding: Categorical variables were converted into binary vectors using one-hot encoding to ensure compatibility with machine learning models, while avoiding any implicit ordinal assumptions.

Dimensionality Reduction (Clustering Only):

Principal Component Analysis (PCA) was applied during clustering to simplify the dataset by retaining 7 components, which captured 85% of the total variance. This step reduced noise, improved clarity, and enhanced clustering efficiency while maintaining key patterns in the data.



Methodology

Classification:

This phase aimed to predict the Alpha class using supervised learning models:

1. Random Forest:

- Selected for its ensemble approach, which combines multiple decision trees to improve accuracy and robustness against overfitting.

- Provides insights into feature importance, making it valuable for understanding which variables most influence predictions.

2. Gradient Boosting:

- Chosen for its iterative refinement process, where each tree corrects the errors of the previous one, leading to high predictive accuracy.
- Effective for handling complex patterns in structured data.

3. XGBoost:

- Utilized for its advanced boosting capabilities, including optimized memory usage and faster computation compared to other boosting models.

4. SVM (Support Vector Machine):

We used SVM with an RBF kernel to model non-linear relationships and maximize the margin between classes. This approach was chosen for its effectiveness in handling complex datasets with overlapping features and its strength against noise, making it suitable for identifying patterns in challenging data.

5. k-Nearest Neighbors (kNN)

- Included for its simplicity and effectiveness in capturing local patterns by considering the nearest data points for classification.
- Non-parametric, making it flexible for datasets without strong assumptions about underlying distributions.

Clustering:

Unsupervised learning techniques were employed to explore natural groupings. Through extensive testing, we identified that these models offered the most meaningful clusters, as confirmed by visualizations and metrics. Each model was chosen for its specific advantages:

K-Means Clustering:

- K-Means partitions the dataset into a predefined number of clusters (k) by assigning each data point to the nearest centroid and iteratively optimizing these centroids to reduce intra-cluster variance.
- Chosen for its simplicity and efficiency in partitioning data into distinct clusters.

Hierarchical Clustering (Agglomerative):

- Hierarchical clustering constructs a tree-like structure (dendrogram) to represent nested groupings within the data. This method provided a different perspective from K-Means. Hierarchical clustering builds a dendrogram to represent nested groupings within the data, offering a detailed perspective on relationships. Despite their differing methodologies, both algorithms arrived at similar cluster groupings, which strongly supports the validity of the results and enhances confidence in the identified clusters.

DBSCAN:

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) groups data points based on their density, identifying clusters of varying shapes and densities without requiring a predefined number of clusters.

Despite extensive attempts using various parameters and visualizations, DBSCAN failed to classify the data into meaningful clusters. However, it excelled at identifying outliers.

Experiments and Results

Classification Results:

Before presenting the results, parameter tuning was performed using grid search to identify the optimal hyperparameters for each model. Following this, cross-validation was applied to evaluate model stability and minimize overfitting.

The models were evaluated using the following metrics, tailored specifically to the nature of our dataset and objectives:

Accuracy: Chosen as the primary metric because our dataset was balanced, and overall correctness was critical for reliably classifying all funds.

Precision: Evaluated to understand false positives, which had minimal negative impact in our dataset. However, it provided valuable insights into the model's ability to identify truly high-performing funds.

Recall: Assessed to minimize false negatives, ensuring we captured all high-performing funds, which was important but secondary to overall accuracy.

F1-Score: Used to balance precision and recall, offering insights into overall model performance when both metrics are moderately significant.

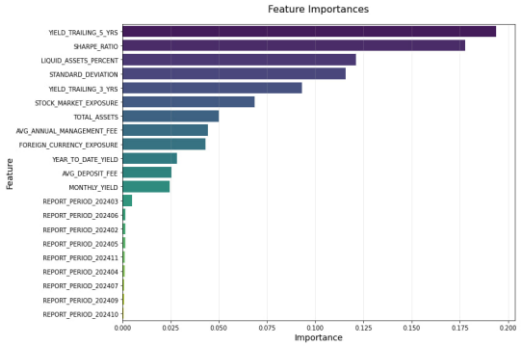
Accuracy was chosen as the primary metric to ensure reliable classification of the Alpha measure across all data points. With a balanced dataset and relatively few rows, it provided an effective measure of overall correctness. Precision and recall were secondary since specific error types were less critical in this context.

Random Forest delivered the highest accuracy (96.6%) and F1-score (97.6%), outperforming other models. XGBoost (95.5% accuracy, 96.8% F1-score) and Gradient Boosting (95.1% accuracy, 96.5% F1-score) closely followed, showing strong reliability. Random Forest emerged as the most effective classifier for this task.

	Model	Accuracy	Precision	Recall	F1-Score
0	Random Forest	0.966292	0.968254	0.983871	0.976000
1	Gradient Boosting	0.951311	0.957672	0.973118	0.965333
2	XGBoost	0.955056	0.957895	0.978495	0.968085
3	SVM	0.943820	0.952381	0.967742	0.960000
4	KNN	0.932584	0.937500	0.967742	0.952381

Top 3 Most Important Features:(Random forest)

YIELD_TRAILING_5_YRS (19.4%),SHARPE_RATIO (17.8%),LIQUID_ASSETS_PERCENT (12.1%)



	Feature	Importance
6	YIELD_TRAILING_5_YRS	0.193891
8	SHARPE_RATIO	0.177706
9	LIQUID_ASSETS_PERCENT	0.121093
7	STANDARD_DEVIATION	0.115679
5	YIELD_TRAILING_3_YRS	0.093174
10	STOCK_MARKET_EXPOSURE	0.068670
0	TOTAL_ASSETS	0.049874
1	AVG_ANNUAL_MANAGEMENT_FEE	0.044453
11	FOREIGN_CURRENCY_EXPOSURE	0.043154
4	YEAR_TO_DATE_YIELD	0.028269
2	AVG_DEPOSIT_FEE	0.025416
3	MONTHLY_YIELD	0.024506

Model Performance Interpretation
An interesting finding in the classification was that the 5-year yield (YIELD_TRAILING_5_YRS) had a stronger influence on the Alpha measure compared to shorter-term yields, such as monthly or yearly yields. This highlights the significance of long-term performance in evaluating fund success.

Clustering Results:

Parameter Tuning for Clustering Models:

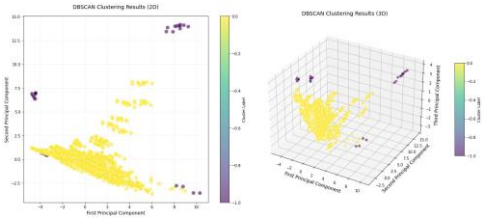
K-Means: The Elbow Method, applied on the inertia (sum of squared distances), was used to determine the optimal number of clusters.

Hierarchical Clustering: The optimal number of clusters was determined using a range of *n_clusters* and selecting the configuration with the highest Silhouette Score.

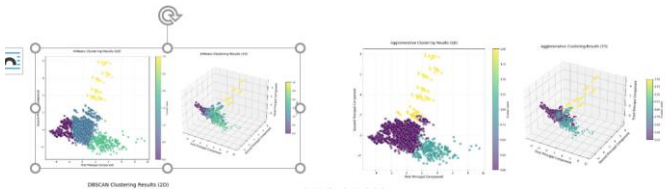
DBSCAN: The Parameters 'eps' and 'min samples' were fine-tuned using the K-Distance graph and the highest Silhouette Score.

Interpretation of the Clustering Models:

DBSCAN: The model struggled to classify the data into meaningful clusters but successfully identified outliers in the dataset.



Agglomerative Clustering & K-Means: These models produced clusters that were slightly "stretched" or less distinct due to their sensitivity to outliers. Based on these observations, we decided to remove the outliers detected by DBSCAN and reapply the Agglomerative Clustering and K-Means models without the influence of the outliers.



Clustering Results After Removing Outliers:

After removing the outliers detected by DBSCAN, we re-evaluated the data using Agglomerative Clustering and K-Means:

Both algorithms produced relatively similar clusters in terms of data distribution.
However, K-Means created 4 clusters, while Agglomerative Clustering formed 3 clusters.

To ensure forceful classification, we decided to retain only the data points where both algorithms agreed on their clustering. This approach provides a consistent and reliable classification of the data.

Cluster Observations:

Cluster 2: High values for *TOTAL_ASSETS, STOCK_MARKET_EXPOSURE, and FOREIGN_CURRENCY_EXPOSURE*, suggesting substantial resources and high diversification.

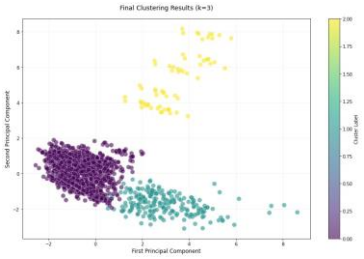
Cluster 1: Largest Standard Deviation, indicating high variability and risk, with relatively higher yields (*YIELD_TRAILING_3_YRS and YIELD_TRAILING_5_YRS*).

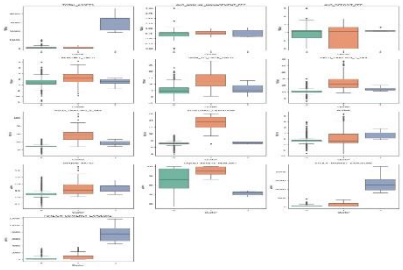
Cluster 0: Poorer performance, representing conservative or underperforming funds with lower asset allocation and yields.

An interesting finding was that Cluster 2, despite having the highest TOTAL_ASSETS, did not achieve the highest yields across all time periods, which was unexpected given its substantial resources and diversification.

Limitations

K-Means Over-Classification: Based on visualizations, K-Means may have classified the data into too many clusters, potentially reducing clarity in cluster separation.





Conclusion and Discussion

Summary of Contributions:

This project utilized supervised and unsupervised machine learning techniques to analyze the performance of insurance funds. Key contributions include:

Classification: We evaluated five models for classification, with Random Forest emerging as the top performer. It achieved the highest accuracy (96.6%) and F1-score (97.6%) for predicting Alpha. The top features influencing this performance were *YIELD_TRAILING_5_YRS, SHARPE_RATIO, and LIQUID_ASSETS_PERCENT*.

Clustering: Using DBSCAN, we removed outliers from the dataset. By combining K-Means and Agglomerative Clustering, we formed a final clustering solution. Cluster 2 holds substantial resources, enabling them to diversify heavily into foreign investments and stock market exposure. Cluster 1 contained high-risk, high-variability funds, and Cluster 0 represented underperforming funds with lower yields. From Cluster 1, we observed that high risk is often associated with higher potential yields, reflecting the trade-off between risk and return.

Roles and Contributions:

we worked together on all aspects of the project. the code was written and executed on Itamar's computer, while the report and presentation were prepared on Tomer's computer.

Future Direction:

Expanding the dataset to include funds from different regions or countries would allow for a comprehensive analysis of regional differences in fund performance and clustering patterns. This could uncover how local market conditions, regulations, and investment strategies influence fund behavior, providing more globally relevant insights.