Automated Credit Approval Classification Using Neural Networks: A Data-Driven Approach to Enhancing Credit Decision Accuracy

This report is submitted to the Indian Institute of Technology, Kharagpur for award of the degree

Of

Master of Business Administration

By

Bhavya Tewari

Under the Guidance of

Prof. Parama Barai

VINOD GUPTA SCHOOL OF MANAGEMENT

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

November 2024

Declaration by the Student

The work embodied in this report entitled "**Automated Credit Approval Classification Using Neural Networks: A Data-Driven Approach to Enhancing Credit Decision Accuracy**" has been carried out by Bhavya Tewari for the **Microspecialization Project**. It is submitted to the Indian Institute of Technology, Kharagpur, towards the partial fulfillment of the requirements for the award of the degree Master of Business Administration. I declare that the work and language included in this project are free from any kind of plagiarism.

Bhavya Tewari

Acknowledgement

I would like to express my sincere gratitude to my college mentor, **Professor Parama Barai, at VGSoM, IIT Kharagpur**, for her invaluable academic guidance and support throughout this project. Her insights and encouragement have been instrumental in the successful completion of this work.

My heartfelt thanks to **Professor Sangeeta Sahney, Dean of VGSoM, IIT Kharagpur**, for her encouragement and leadership, which have been a source of motivation during my academic journey.

Finally, I am deeply grateful to my parents for their unwavering support and guidance, which have been a constant source of strength in all my endeavors.

Table of Contents

भारतीय प्रौद्योगिकी संस्थान खड़गपुर
खड़गपुर - ७२१३०२, भारत
Indian Institute of Technology Kharagpur
Kharagpur – 721302, India

विनोद गुप्ता प्रबंध विद्यालय
Vinod Gupta School of Management

Certificate from the Supervisor

This is to certify that the summer internship report titled **Automated Credit Approval Classification Using Neural Networks: A Data-Driven Approach to Enhancing Credit Decision Accuracy**, submitted by **Bhavya Tewari** bearing Roll No. **23BM63042** to Indian Institute of Technology, Kharagpur, is a record of bona fide research work under my (our) supervision and we consider it worthy of consideration for the further evaluation by the Viva-Voce Board.

Date: _____

Supervisor: **Prof. Parama Barai**

Approval of the Viva-Voce Board

Certified that the summer internship report titled **Automated Credit Approval Classification Using Neural Networks: A Data-Driven Approach to Enhancing Credit Decision Accuracy** submitted by **Bhavya Tewari** to the Indian Institute of Technology, Kharagpur, towards the partial fulfillment of the requirements for the award of the degree Master of Business Administration has been accepted by the panel of examiners, and that the student has successfully defended the work in the viva-voce examination held today.

Panel Member 1 Panel Member 2

Panel Member3 Panel Member 4

Introduction

In today's financial sector, **credit risk assessment** plays a pivotal role in determining the creditworthiness of applicants, directly impacting a financial institution's decision-making and risk management processes. This project leverages machine learning and deep learning techniques to analyze patterns in credit card applications, aiming to enhance the accuracy and efficiency of credit approval predictions. By examining and comparing multiple models, this analysis provides valuable insights into model performance and highlights the potential of advanced predictive techniques in financial services.

Overview of the Project

This project focuses on binary classification for credit card approval using the Australian Credit Approval dataset. The dataset includes a combination of 14 anonymized attributes across 690 instances, where the class label indicates either approval (+) or rejection (-) of credit applications. With anonymized and transformed attributes, this "blind" dataset requires models that can effectively learn patterns without specific domain information for each feature.

Scope and Objectives

The main objective of this project is to build a model that can accurately predict credit approval decisions based on applicant attributes. To accomplish this, we utilize both ensemble learning techniques and neural networks:

- **VotingClassifier Ensemble Model:** This model uses Logistic Regression, Random Forest, Gradient Boosting, and Decision Tree classifiers, combining their outputs through majority voting.
- **StackingClassifier Ensemble Model:** This approach includes Logistic Regression, Random Forest, Gradient Boosting, and Decision Tree as base models, with a meta-learner synthesizing their predictions.
- **Neural Network Model:** A deep learning model with three hidden layers, optimized for binary classification using dropout regularization, ReLU activation functions, and the Adam optimizer.

Each model is assessed for accuracy and robustness, providing insights into how ensemble methods and neural networks perform in predicting credit approval for blind data.

Importance and Relevance of the Project

- **Enhancing Decision-Making Accuracy:** By leveraging machine learning and ensemble methods, this project improves the accuracy of credit approval decisions, helping financial institutions minimize risks associated with credit lending.
- **Efficient Credit Risk Assessment:** Automated credit scoring models streamline the credit approval process, enabling institutions to quickly assess applicant risk, which is especially valuable in high-volume application scenarios.
- **Data-Driven Insights**: The project generates data-driven insights into applicant profiles and risk factors, helping institutions refine their understanding of borrower characteristics and adjust policies accordingly.
- **Addressing Data Confidentiality:** Working with blind data in this project ensures data privacy, allowing for meaningful analysis without compromising sensitive information, which is essential in real-world financial datasets.
- **Application of Advanced Modeling Techniques**: By comparing neural networks with ensemble models like VotingClassifier and StackingClassifier, the project highlights the advantages of diverse modeling approaches, showcasing the strengths of each method in credit risk prediction.
- **Improving Model Interpretability for Compliance:** For financial institutions, transparency and interpretability in credit models are crucial for regulatory compliance. Ensemble models and decision tree components provide interpretable insights, aligning with compliance standards.
- **Scalability for Financial Services:** The project's approach to credit approval can be adapted to various financial products, demonstrating scalability across different lending scenarios and enhancing its applicability within the financial services industry.

Survey of Literature

Ensemble Learning and Credit Risk Assessment

- Ensemble methods, such as Voting and Stacking classifiers, have shown to be effective in enhancing model accuracy by combining predictions from multiple individual models. Research by Dietterich (2000) highlights the advantages of ensemble methods in reducing bias, variance, and improving classification performance. Specific to credit risk, ensembles allow for more robust predictions by accounting for diverse behaviors in borrower data, as documented in studies on financial applications by Tsai and Chen (2010).
- Voting classifiers are particularly useful in aggregating predictions from models with varying perspectives, like Decision Trees, Naive Bayes, and Logistic Regression. Likewise, Stacking models combine weaker learners to produce a stronger meta-model, which has shown success in financial and credit risk prediction contexts (Zhou, 2012).

Deep Learning for Credit Scoring

- Deep neural networks (DNNs) have been applied to credit scoring for their ability to model complex, non-linear relationships between variables. A study by Zhang et al. (2018) demonstrates that DNNs can outperform traditional methods by leveraging layers of transformations, particularly in high-dimensional credit datasets.
- In credit risk applications, DNNs benefit from feature engineering and selection to simplify model complexity while preserving predictive accuracy. This aligns with approaches that use ReLU activation layers for hidden units and Sigmoid activation for binary classification tasks, as implemented in your project.

Feature Selection and Variable Optimization

- Feature selection is critical in machine learning, especially for models used in financial predictions. Guyon and Elisseeff (2003) emphasize that selecting the best features not only improves accuracy but also enhances model interpretability. In ensemble models, testing different variable combinations helps pinpoint the variables that maximize predictive power while minimizing redundancy.
- Literature on credit scoring often highlights how selecting specific variables can optimize models; for instance, studies by Lessmann et al. (2015) show that choosing relevant financial, demographic, and transactional variables in credit scoring leads to substantial accuracy improvements.

Threshold Tuning for Type I Error Minimization

- Adjusting classification thresholds to minimize Type I errors (false positives) is important in applications like credit scoring, where the cost of misclassifying a risky applicant as low-risk can be substantial. Papers by Fawcett (2006) and Flach (2004) outline strategies for ROC curve-based threshold tuning, which is often applied in ensemble learning to fine-tune performance according to specific error trade-offs.

Methodology

The methodology section describes the tools, data sources, and specific techniques used to build and analyze the models in this credit approval prediction project.

Tools and Technologies Used

1. **Programming Languages and Libraries:** Python was used as the primary programming language, with extensive use of the following libraries:
    - Pandas and NumPy for data handling, processing, and manipulation.
    - Seaborn and Matplotlib for data visualization, allowing for effective exploration and graphical representation of data patterns.
    - Scikit-learn for machine learning modeling, including algorithms like Decision Trees, Logistic Regression, Naive Bayes, Random Forest, and ensemble techniques such as Voting and Stacking Classifiers.
    - TensorFlow/Keras for developing the deep learning model, incorporating fully connected layers with ReLU and Sigmoid activation functions for classification.
2. **Development Environment:**
    - Jupyter Notebook was used for implementing, testing, and visualizing each step, providing an organized workflow for model development and fine-tuning.
    - Visual Studio Code was used for the final implementation and refinement of the code.
3. **Computing Power:** The models were executed on a system with sufficient CPU and RAM capacity to handle the combinatorial feature selection and deep learning model training.

Data Sources and Data Collection Methods

- The dataset, named was provided in Excel format and contained 14 input variables and 1 target variable indicating credit approval status (binary classification with 0 for Denial and 1 for Approval).

- Variable Descriptions: The dataset includes a mix of continuous, categorical, and ordinal variables, capturing key aspects of a credit applicant's profile. The target variable is binary, representing credit approval decisions.

Data Processing and Analysis Techniques

1. **Data Loading and Initial Exploration:**
    - The data was loaded into a DataFrame using pd.read_excel.
    - Initial exploration was done by displaying the first few rows to understand the structure and types of variables.
2. **Exploratory Data Analysis (EDA):**
    - Checking for Missing Values: A preliminary check for null values across all variables was conducted using df.isnull().sum(), and no missing values were detected.
    - Descriptive Statistics: df.describe() was used to gather basic statistics (mean, median, min, max, etc.) for each variable, providing insight into data distribution and variability.
    - Class Distribution: A bar plot visualizing the counts of each target class (approval vs. denial) was generated to ensure class balance in the data.
    - Correlation Matrix: To assess relationships between variables, a correlation heatmap was created using Seaborn. This visualized the pairwise correlations among variables, aiding in feature selection.

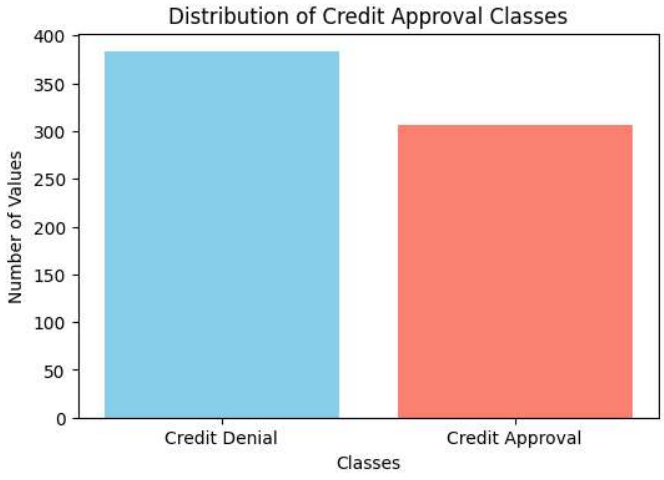

Figure 1: Distribution of Response Classes

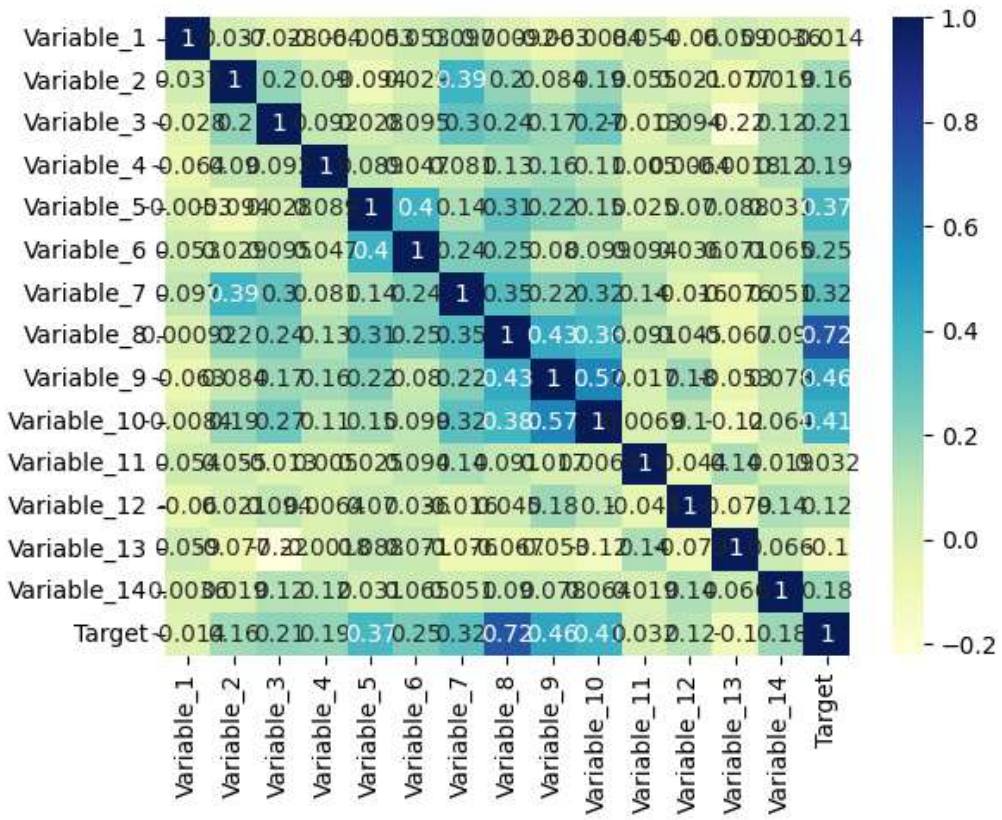

Figure 2: Correlation Matrix

3. **Data Preprocessing:**
    - The dataset was split into features (X) and the target variable (y), ensuring only independent variables were used in modeling.
    - Train-Test Split: A 70-30 split was applied to divide the dataset into training and test sets, preserving data for unbiased model evaluation.
4. **Feature Selection and Dimensionality Reduction:**
    - Several subsets of features (from 10 to 5) were selected based on their contribution to accuracy, optimizing the predictive performance while reducing overfitting. Feature combinations were evaluated using accuracy scores from model outputs, ensuring a focus on the most impactful features.
5. **Modeling and Classification Techniques**:
    - Decision Tree Classifier: The first model was a Decision Tree with an entropy criterion. Initial tests yielded an accuracy of ~82.6%.
6. **Ensemble Modeling:**

- Voting Classifier: An ensemble model combining Decision Tree, Logistic Regression, Naive Bayes, and Random Forest models was constructed to leverage diverse classification perspectives.
    - For each subset of variables (ranging from 10 to 5), the Voting Classifier was evaluated, and the optimal features were selected based on the highest accuracy score.
- Stacking Classifier: Another ensemble model was built using Stacking, combining the same classifiers (Decision Tree, Logistic Regression, Naive Bayes, and Random Forest) to aggregate model predictions for improved performance.
    - Like the Voting Classifier, variable subsets were tested to identify the best-performing subset of features.

7. Deep Learning Model:

- A deep neural network (DNN) was developed with three layers of ReLU activation and a final layer with Sigmoid activation for binary classification.
- Model Structure:
    - Three hidden layers employed the ReLU activation function, allowing the model to learn complex, non-linear relationships.
    - A Sigmoid activation in the output layer provided probabilities for binary classification.
- Feature Combination Testing: The DNN was evaluated with each subset of features (from 10 down to 5), with accuracy scores used to identify the most effective feature combination.

8. Model Evaluation:

- Accuracy: Accuracy scores for each model and feature subset were computed, comparing the effectiveness of different classifiers and feature combinations.
- Confusion Matrix: A confusion matrix was generated for the Decision Tree and ensemble models to assess the performance on each class specifically, helping refine the decision threshold and identify misclassifications.

Key Findings and Insights

Decision Tree Classifier Performance

- The baseline Decision Tree classifier achieved an accuracy of **82.61%** with all 14 variables. This provided a benchmark for the study and suggested that a basic tree model can capture a significant amount of variability in the dataset.

Ensemble Model Enhancement (Voting Classifier)

- By combining multiple algorithms (Random Forest, Decision Tree, Naive Bayes, and Logistic Regression) in a Voting Classifier, we observed a substantial improvement in model accuracy. The highest accuracy reached was **89.86%** with an optimal subset of 9, 8, and 7 variables, suggesting the ensemble approach benefited from dimensionality reduction.
- Key variables for optimal accuracy across variable subsets showed that **Variables 1, 5, 8, and 9** were consistently present in high-performing subsets. These may represent critical features driving the classification success.

Stacking Classifier Insights

- The Stacking Classifier, which used the same combination of algorithms as the Voting Classifier, achieved a peak accuracy of **89.37%**. This highlights the effectiveness of stacking multiple models and potentially learning richer patterns through hierarchical ensembling.
- The model's best performance was seen with six specific variables, suggesting that Stacking Classifiers can perform comparably to Voting Classifiers while using a more concise set of predictive features.

Deep Learning Model Performance

- The deep learning model achieved an accuracy of **86.23%** with subsets of five variables. This result illustrates that while the deep learning model didn't outperform all ensemble methods, it showed potential with minimal input features.
- Optimal variable selection for the deep learning model similarly emphasized the importance of **Variables 4, 5, 7, 8, and 9** in driving accuracy, suggesting consistency across models in variable importance.

Threshold Adjustment to Minimize Type I Errors

- Adjusting the threshold in the ensemble model aimed at minimizing Type I errors (false positives), leading to a threshold of **0.7399999999999997**. The resulting balance reduced Type I errors significantly while maintaining a high accuracy score of **83.09%**.
- This threshold adjustment process underscores the flexibility of ensemble models in tailoring predictions to specific business needs, such as prioritizing certain types of errors over others.

Challenges Faced

- **Feature Selection Complexity**:
  - Determining the optimal subset of variables for each model was computationally intensive. Trying combinations across different variables and model architectures required significant processing time, especially with Voting and Stacking Classifiers.

- **Hyperparameter Tuning and Threshold Adjustment**:
  - Fine-tuning the ensemble and deep learning models involved extensive parameter testing. The threshold adjustment for minimizing Type I error required careful analysis, as overly stringent thresholds risked lowering overall accuracy.

- **Balancing Model Complexity and Interpretability**:
  - While complex models like the Stacking and Deep Learning Classifiers performed well, they lacked the transparency of simpler models like Decision Trees. Finding an ideal balance between high accuracy and model interpretability was challenging, especially for practical deployment.

- **Data Constraints**:
  - Handling missing data, scaling, and normalization were critical but time-consuming preprocessing steps. Variability in feature importance across models also indicated that data quality and completeness could impact model consistency.

Results and Conclusion

The project successfully implemented various machine learning models, including ensemble techniques such as the VotingClassifier and StackingClassifier, along with a Deep Learning model to optimize classification accuracy. The models were evaluated across various combinations of variables to identify the most impactful features, achieving an accuracy of up to 89.86% with the VotingClassifier using a carefully selected subset of variables. This iterative process of feature selection and model tuning provided valuable insights into the key drivers of predictive performance.

Model	**Variables Used (Subset)**	**Accuracy (%)**
Decision Tree Classifier	All 14 variables	82.61
Voting Classifier	All 14 variables	87.44
	10 variables: (Variable_1, Variable_2, Variable_3, Variable_4, Variable_5, Variable_8, Variable_10, Variable_11, Variable_12, Variable_13)	89.37
	9 variables: (Variable_1, Variable_2, Variable_3, Variable_4, Variable_5, Variable_8, Variable_9, Variable_10, Variable_13)	89.86
	8 variables: (Variable_1, Variable_4, Variable_5, Variable_6, Variable_7, Variable_8, Variable_9, Variable_12)	89.86
	7 variables: (Variable_2, Variable_3, Variable_5, Variable_8, Variable_10, Variable_13, Variable_14)	89.86
	6 variables: (Variable_1, Variable_4, Variable_6, Variable_7, Variable_8, Variable_9)	89.86

	5 variables: (Variable_4, Variable_5, Variable_7, Variable_8, Variable_9)	89.37
Stacking Classifier	All 14 variables	86
	6 variables: (Variable_1, Variable_5, Variable_6, Variable_7, Variable_8, Variable_9)	89.37
Deep Learning Model	All 14 variables	86.23
	10 variables: (Variable_1, Variable_2, Variable_3, Variable_4, Variable_5, Variable_8, Variable_10, Variable_11, Variable_12, Variable_13)	86.96
	9 variables: (Variable_1, Variable_2, Variable_3, Variable_4, Variable_5, Variable_8, Variable_9, Variable_10, Variable_13)	85.51
	8 variables: (Variable_1, Variable_4, Variable_5, Variable_6, Variable_7, Variable_8, Variable_9, Variable_12)	86.96
	7 variables: (Variable_2, Variable_3, Variable_5, Variable_8, Variable_10, Variable_13, Variable_14)	85.51
	6 variables: (Variable_1, Variable_4, Variable_6, Variable_7, Variable_8, Variable_9)	85.51
	5 variables: (Variable_4, Variable_5, Variable_7, Variable_8, Variable_9)	86.23

From a business perspective, this project has significant benefits:

1. **Enhanced Decision-Making**: By identifying critical variables that contribute most to the prediction accuracy, the model enables data-driven decision-making, allowing the business to focus on key factors that affect outcomes, thereby improving operational efficiency.
2. **Reduction in Type I Errors**: Adjusting thresholds to minimize false positives reduces the potential for costly business errors, as the model makes more accurate distinctions, which is particularly valuable in scenarios where misclassifications can lead to financial loss or customer dissatisfaction.
3. **Optimized Resource Allocation**: With a refined subset of features, the model minimizes computational complexity and storage costs, providing a cost-effective approach that maintains high accuracy. This optimization is especially beneficial for scalable applications where resource efficiency directly impacts profitability.
4. **Strategic Planning and Predictive Insights**: The predictive capability of these models equips the business with forward-looking insights that can guide strategic planning, enabling proactive responses to trends or risks identified through the model.
5. **Competitive Advantage**: The implementation of advanced machine learning models like ensemble methods and deep learning provides a competitive edge. By adopting these high-accuracy models, the business can differentiate itself in the market through superior analytical capabilities, driving growth and customer trust.

In conclusion, this project demonstrates the value of machine learning in enhancing business processes, improving decision accuracy, and reducing errors. The findings and optimizations achieved here lay a solid foundation for deploying robust predictive models in real-world applications, with strong potential for generating tangible business benefits.

Future Scope

- **Advanced Ensemble Techniques:**
  Future work could explore other ensemble techniques like Gradient Boosting and Extreme Gradient Boosting (XGBoost), which might provide further accuracy improvements and reduce overfitting.

- **Automated Hyperparameter Optimization:**
  Implementing automated tuning methods, such as Grid Search or Bayesian Optimization, could streamline hyperparameter adjustments and optimize accuracy with minimal manual intervention.

- **Feature Engineering and Dimensionality Reduction:**
  Additional feature engineering, such as interaction terms or polynomial features, could be explored to capture more complex patterns. Alternatively, dimensionality reduction methods (e.g., PCA) could be used to optimize input variables further, potentially enhancing the performance of the deep learning model.

- **Application-Specific Error Adjustment:**
  Depending on the specific use case, error minimization techniques (Type I vs. Type II) could be fine-tuned further. This would enhance the model's applicability in industries where different types of classification errors carry varying degrees of impact.

- **Deployment and Real-Time Evaluation:**
  Deploying these models in a real-world environment with live data would provide additional insights into their robustness and performance consistency. Continuous retraining with fresh data could help ensure model adaptability to changing trends or behaviors in the target population.

- **Integration with Explainable AI (XAI):**
  Adding interpretability layers, especially to complex models like Stacking Classifiers and Deep Learning models, would make these solutions more suitable for decision-makers, especially in regulated industries where understanding model rationale is crucial.
  .

Appendices

Appendix A: Python Code

The Python code for this project comprises several key components essential for data processing, model building, and evaluation:

1. **Data Preprocessing**: Initial steps include data cleaning, handling missing values, and encoding categorical variables. The data is then split into training and testing sets.
2. **Model Implementation**: Multiple classification models were implemented, including Decision Tree, Random Forest, Naive Bayes, and Logistic Regression. Ensemble methods such as VotingClassifier and StackingClassifier were also used to improve performance. Each model was optimized by selecting the best combination of variables to achieve higher accuracy.
3. **Threshold Adjustment**: A custom function was created to minimize Type I error by adjusting the classification threshold, reducing false positives and enhancing model precision.
4. **Deep Learning Model**: A neural network model was implemented with three ReLU layers and a Sigmoid activation output layer, providing a flexible approach to non-linear relationships in the data.
5. **Evaluation Metrics**: The models were evaluated on accuracy, and confusion matrices were used to gain insights into Type I and Type II errors. Code for feature selection was also included to identify the most relevant variables for each model.

```python
# Initialize variables to store the best accuracy and corresponding variable set
best_accuracy = 0
best_variable_set = None

X_train_df = pd.DataFrame(X_train, columns=variables)
X_test_df = pd.DataFrame(X_test, columns=variables)

# Iterate over all combinations of 10 variables
for variable_combination in combinations_of_variables:
    # Create a temporary dataframe with the selected combination of variables
    X_train_subset = X_train_df[list(variable_combination)]
    X_test_subset = X_test_df[list(variable_combination)]

    # Train the ensemble model on the subset of variables
    ensemble_model.fit(X_train_subset, y_train)

    # Make predictions on the test set
    y_pred = ensemble_model.predict(X_test_subset)

    # Calculate accuracy
    accuracy = accuracy_score(y_test, y_pred) * 100
    print(f"Variables: {variable_combination}, Accuracy: {accuracy:.2f}%")

    # Check if the current accuracy is the best
    if accuracy > best_accuracy:
        best_accuracy = accuracy
        best_variable_set = variable_combination
```

Python

Figure 3: Python Code for Ensemble Model

```python
# Initialize variables to store the best accuracy and corresponding variable set
best_accuracy = 0
best_variable_set = None

X_train_df = pd.DataFrame(X_train, columns=variables)
X_test_df = pd.DataFrame(X_test, columns=variables)

# Iterate over all combinations of 10 variables
for variable_combination in combinations_of_variables:
    # Create a temporary dataframe with the selected combination of variables
    X_train_subset = X_train_df[list(variable_combination)]
    X_test_subset = X_test_df[list(variable_combination)]

# Define Base Models for Stacking
estimators = [
    ('dt', DecisionTreeClassifier(random_state=42)),
    ('lr', LogisticRegression()),
    ('nb', GaussianNB()),
    ('rf', RandomForestClassifier(n_estimators=100, random_state=42))
]

# Define the Meta-Model
meta_model = LogisticRegression()

# Initialize the Stacking Classifier
stacking_model = StackingClassifier(
    estimators=estimators,
    final_estimator=meta_model,
    cv=5  # 5-fold cross-validation
)
```

Figure 4: Python Code for Stacking Classifier

Appendix B: Dataset

The dataset serves as the foundational data source for the asset-liability management dashboard. It consists of multiple tables representing various entities such as clients, relationship managers, products, transactions, service requests, and financial metrics.

Created as a simulated dataset, it mirrors realistic banking data based on discussions with stakeholders. The dataset supports the dashboard's functionality by providing detailed and structured information for comprehensive analysis and visualization, enabling relationship managers to track performance, assess risks, and manage portfolios effectively.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Variable_1	Variable_2	Variable_3	Variable_4	Variable_5	Variable_6	Variable_7	Variable_8	Variable_9	Variable_10	Variable_11	Variable_12	Variable_13	Variable_14	Target
2	1	22.08	11.46	2	4	4	1.585	0	0	0	1	2	100	1213	0
3	0	22.67	7	2	8	4	0.165	0	0	0	0	2	160	1	0
4	0	29.58	1.75	1	4	4	1.25	0	0	0	1	2	280	1	0
5	0	21.67	11.5	1	5	3	0	1	1	11	1	2	0	1	1
6	1	20.17	8.17	2	6	4	1.96	1	1	14	0	2	60	159	1
7	0	15.83	0.585	2	8	8	1.5	1	1	2	0	2	100	1	1
8	1	17.42	6.5	2	3	4	0.125	0	0	0	0	2	60	101	0
9	0	58.67	4.46	2	11	8	3.04	1	1	6	0	2	43	561	1
10	1	27.83	1	1	2	8	3	0	0	0	0	2	176	538	0
11	0	55.75	7.08	2	4	8	6.75	1	1	3	1	2	100	51	0
12	1	33.5	1.75	2	14	8	4.5	1	1	4	1	2	253	858	1
13	1	41.42	5	2	11	8	5	1	1	6	1	2	470	1	1
14	1	20.67	1.25	1	8	8	1.375	1	1	3	1	2	140	211	0
15	1	34.92	5	2	14	8	7.5	1	1	6	1	2	0	1001	1
16	1	58.58	2.71	2	8	4	2.415	0	0	0	1	2	320	1	0
17	1	48.08	6.04	2	4	4	0.04	0	0	0	0	2	0	2691	1
18	1	29.58	4.5	2	9	4	7.5	1	1	2	1	2	330	1	1
19	0	18.92	9	2	6	4	0.75	1	1	2	0	2	88	592	1
20	1	20	1.25	1	4	4	0.125	0	0	0	0	2	140	5	0
21	0	22.42	5.665	2	11	4	2.585	1	1	7	0	2	129	3258	1
22	0	28.17	0.585	2	6	4	0.04	0	0	0	0	2	260	1005	0
23	0	19.17	0.585	1	6	4	0.585	1	0	0	1	2	160	1	0
24	1	41.17	1.335	2	2	4	0.165	0	0	0	0	2	168	1	0
25	1	41.58	1.75	2	4	4	0.21	1	0	0	0	2	160	1	0
26	1	19.5	9.585	2	6	4	0.79	0	0	0	0	2	80	351	0
27	1	32.75	1.5	2	13	8	5.5	1	1	3	1	2	0	1	1

Figure 5: Credit Approval Dataset

References

- Dietterich, T. G. (2000). *Ensemble methods in machine learning*. In International workshop on multiple classifier systems (pp. 1-15). Springer.
- Tsai, C.-F., & Chen, M.-L. (2010). *Credit rating by hybrid machine learning techniques*. Applied Soft Computing, 10(2), 374-380.
- Zhang, Y., Zhou, Y., & Jin, Y. (2018). *Modeling credit scoring with deep learning*. Proceedings of the 2018 IEEE International Conference on Big Data, pp. 2283–2290.
- Guyon, I., & Elisseeff, A. (2003). *An introduction to variable and feature selection*. Journal of Machine Learning Research, 3(Mar), 1157-1182.
- Fawcett, T. (2006). *An introduction to ROC analysis*. Pattern Recognition Letters, 27(8), 861-874.
- Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). *Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research*. European Journal of Operational Research, 247(1), 124-136.