جامعة الملك سعود
King Saud University
كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

King Saud University
College of Computer and Information Sciences
Department of Information Technology

IT 461 Practical Machine Learning

Bank churn prediction

Section #	Student name	Student ID
56702	Aisha Alsaggaf	443203061
56702	Athbah Alaliwei	443200628
56702	Ghaina Alhassnan	443200495
56702	Shaden Alturki	443203057
56702	Dimah Alharbi	443200784

Table of Contents

Table of Tables

Table of Figures

Abstract. This research paper undertakes an extensive and detailed examination of three distinct machine learning models, specifically Linear Regression, Support Vector Machine (SVM), and Dense Neural Network (DNN). to predict **customer churn** (exit likelihood) in a business application. Python serves as the fundamental tool in facilitating a comprehensive and thorough evaluation of their efficacy in predicting trends in the financial market. In the realm of methodology, this study encompasses a multifaceted approach. Python forms the cornerstone for data analysis, model development, and testing. The dataset, sourced from Kaggle, includes customer data, such as demographic details, transaction history, and account activity. With focus on addressing the challenge of **class imbalance**, where non-exited (retained) customers significantly outnumber exited (churned) customers, the core of the study lies in the comparative analysis of the three machine learning models. Each model is rigorously tested against the dataset to evaluate its ability to predict customer exits accurately, enabling a nuanced understanding of their strengths and weaknesses in handling **class imbalance**. The performance of each model is assessed using key metrics such as **accuracy**, **precision**, **recall**, and **F1-score**, with a particular emphasis on **recall**, as minimizing false negatives (i.e., missed exits) is the main priority. While the **Deep Neural Networks (DNN)** provide high accuracy, **Linear Regression** emerges as the most reliable performer in terms of **predictive accuracy** and **consistency**, proving to be a valuable model for churn prediction. This study makes a significant contribution by advancing the application of machine learning techniques to **customer retention**. It offers important insights into the comparative performance of different models, providing a foundation for future research into refining churn prediction systems. The findings stress the importance of **model interpretability** and **recall prioritization** in domains where identifying at-risk customers is crucial for timely intervention, ultimately enhancing customer retention strategies.
Keywords: Linear Regression, SVM, DNN.

1. Introduction

1.1. Addressing Customer Churn and Its Impact

This project focuses on predicting **customer churn** in the banking industry, which occurs when customers leave the bank. Churn directly impacts profitability, as retaining customers is more cost-effective than acquiring new ones. Research shows that acquiring a new customer can cost five to seven times more than retaining an existing one [1]. Additionally, increasing customer retention by just 5% can increase profits by 25% to 95% [2]. Predicting churn helps banks implement **targeted retention** strategies, reduce customer loss, and minimize revenue decline. Given the **competitive nature** of the banking industry, churn prediction is essential to maintain a strong customer base and avoid the costs associated with replacing lost customers.

1.2. Background

Customer churn refers to the loss of customers over a given period and serves as a key indicator of customer satisfaction and loyalty. In banking, understanding churn helps identify customers likely to leave, enabling the development of targeted retention strategies. Reducing churn is vital as

acquiring new customers is significantly more expensive than retaining existing ones—replacing one lost customer often requires acquiring three new ones. Predictive modeling, which uses AI and machine learning to analyze historical data, offers a powerful way to address this issue. By classifying customers based on their likelihood to churn, banks can take proactive steps to minimize losses and improve customer retention.

1.3.Intended Task

The project aims to develop a machine learning model using a **classifier** that predicts whether a customer will churn based on their historical data. The classifier will analyze customer attributes such as age, balance, and product usage to classify customers as likely or unlikely to churn.

1.4.Evaluation Metrics:

The model's performance will be assessed using multiple metrics: **accuracy** for overall correctness, **precision** to measure the avoidance of false positives, **recall** for identifying true positives, and the **F1-score** for a balance between precision and recall. Additionally, **ROC-AUC** will evaluate the model's ability to distinguish between classes, ensuring a comprehensive understanding of its performance.

- **Input**: Customer data (e.g., age, gender, tenure, balance, etc.).
- **Output**: Prediction (Churn = 1, No Churn = 0).

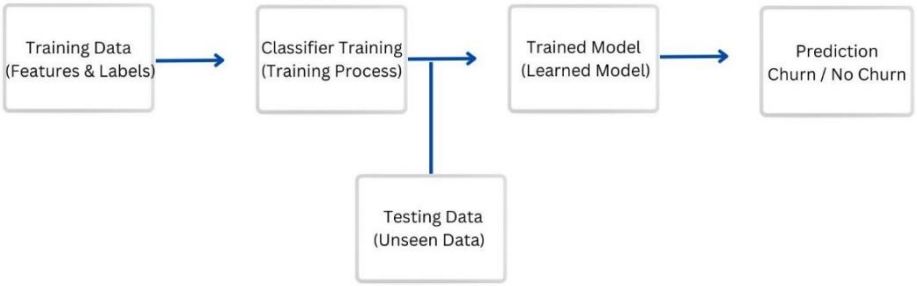


Figure 1:Customer Churn Prediction Process Diagram

This represents the flow of data in the customer churn prediction process, showing how customer data is input into a machine learning model to generate a prediction of whether a customer will churn or not.

2. Related Works:

Customer churn prediction has been successfully applied in various industries, often using machine learning models to detect patterns in customer behavior. For instance, in the telecom industry, a study utilizing the *Telco Customer Churn Dataset* [3] employed Decision Trees and Random Forests to predict customer churn. The data included features like how long the customer has been with the company, their contract type, payment methods, and monthly charges. The Random Forest model had an accuracy of **79%**, showing it works well for handling complex and unbalanced data by combining several decision trees. Telecom companies like Vodafone used this model to figure out which customers were likely to leave and made improvements to their services to keep them [3].

Similarly, in the e-commerce industry, a study used the *Online Retail Customer Churn Dataset* [4], researchers predicted if online shoppers would stop buying. They used Neural Networks and Logistic Regression to look at features like how often customers purchased, how long it had been since their last purchase, and how engaged they were with the store. The Neural Network model reached **82%** accuracy, making it good at finding complex patterns in customer behavior. Logistic Regression was easier to understand but slightly less accurate at **78%**. This study helped online stores figure out which customers were losing interest and allowed them to take action to keep them shopping [4].

A broader study focusing on general churn prediction across various industries employed the *Predictive Analytics for Customer Churn Dataset* [5] to study churn in different industries. They used Decision Trees and Random Forests to analyze transaction history, customer interactions, and personal details. The Random Forest model had an accuracy of **81%**, making it a good tool for figuring out which features matter most in predicting churn. This model helped businesses like banks and insurance companies find out which customers were at risk of leaving and address their issues early [5].

3. Dataset

3.1. Dataset description

The dataset was collected from Kaggle [6], the world's largest data science and machine learning community, where visitors can learn, find data, compete, and collaborate on the cutting edge of machine learning [7]. The dataset provides bank customers churn information, containing 14 significant features

such as: customerID, surname, creditScore, geography, gender, age, tenure, balance, numOfProducts, hasCreditCard, isActiveMember, EstimatedSalary, Exited. These attributes will help us build a good observation and input to train the model.

3.2. Validating the dataset:

Relevance:

As explained in the motivation, this study aims to enhance customer retention. Where we plan to use machine learning models for banks to predict which customers are likely to churn, allowing them to act proactively. The intended outcome of this study includes lower acquisition costs, improved retention, and enhanced customer satisfaction. this dataset provides major information of customers for churn to be studied, by using the following addressed features, a prediction model will be constructed that enables banks to refine their services, address customer needs more effectively, and develop better business strategies.

accessibility:

As this dataset had been located on Kaggle [6], it was easily previewed and downloaded as a CSV file, which is the type of file input we aim to work on. In the process of previewing the dataset, skills and tools needed to properly execute the project have been discussed, these skills include what team members had previously acquired such as knowledge in Python, expertness in cleaning, EDA analysis, and understanding of classification, resulting the conclusion of effectiveness and accessibility of the dataset.

Type:

As shown in the latter (see table 1) types of data include numerical and categorical which helps in clearly stating whether if something is or isn’t in the group (for example, exited 1 or not 0) this type of data will help improve the supervised learning of churn pridiction.

Quality:

The collected dataset is well organized and labelled, it does not include any missing data. As the dataset is relatively clean, we believe the data collection process was designed well. There are only about 2 to 3 invalid inputs, which reduces the expected time and energy spent wrangling the data [8]. The previously stated declaration was observed by inspecting the dataset manually, any further input details will be handled in exploring and preprocessing phase.

By validating the **Relevance, Accessibility**, **Type,** and **Quality** [9] of the data, we were able to make choices that will support us across the entire project. As we believe choosing this dataset will empower us to go forward and prevent us from having to backtrack as much over the course of the project.

3.3. Dataset summary

The dataset provides bank customers churn information, it contains 14 features and 10000 Rows, for rows each row represents individual customer, while columns represent various customer features or characteristics as shown below:

- Customer ID: A unique identifier for each customer.
- Surname: The customer's surname or last name.
- Credit Score: A numerical value representing the customer's credit score.
- Geography: The country where the customer resides (France, Spain or Germany).
- Gender: The customer's gender (Male or Female)
- Age: The customer's age.
- Tenure: The number of years the customer has been with the bank.
- Balance: The customer's account balance.
- NumOfProducts: The number of bank products the customer uses (e.g., savings account, credit card)
- HasCrCard: Whether the customer has a credit card (1 = yes, 0 = no).
- IsActiveMember: Whether the customer is an active member (1 = yes, 0 = no).
- EstimatedSalary: The estimated salary of the customer.
- Exited: Whether the customer has churned (1 = yes, 0 = no).

3.4. short overview of the dataset

The table below shows the 13 features with their types and number of classes (if exists)

Table 1: dataset overview

Number of features: 13		Number of rows: 10000
Name of feature	**Data type**	**Number of classes (if exists)**
Customer ID	Qualitative nominal	-
Surname	Qualitative nominal	-
Credit Score	Quantitative ratio	-
Geography	Qualitative nominal	3 (Spain, Germany, France)

Gender	Qualitative nominal	2 (Male, Female)
Age	Quantitative ratio	-
Tenure	Quantitative ratio	-
Balance	Quantitative ratio	-
NumOfProducts	Quantitative ratio	-
HasCrCard	Qualitative binary	2 (0,1)
IsActiveMember	Qualitative binary	2 (0,1)
EstimatedSalary	Quantitative ratio	-
Exited	Qualitative binary	2 (0,1)

3.5. Class Distribution

For the target class 'Exited' the figure below highlights the distribution of classes 0 and 1 (not exited and exited respectively).

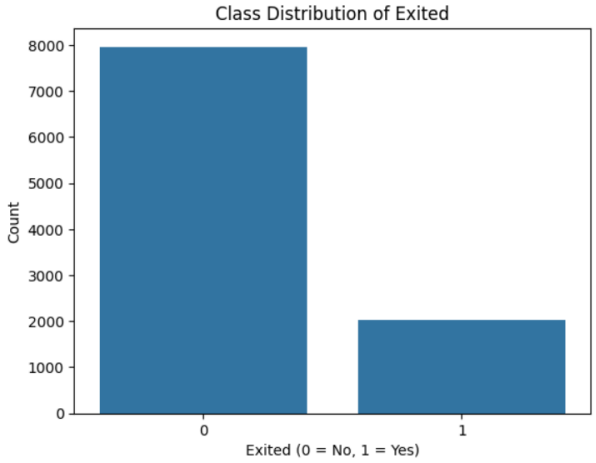


Figure 2: Target class distribution.

3.6. Descriptive statistics

The table below show Descriptive statistics of numerical features (CreditScore, Age, Tenure, Balance, and EstimatedSalary)

Table 2: Descriptive statistics

	CreditScore	**Age**	**Tenure**	**Balance**	**EstimatedSalary**
count	10002	10001	10002	10002	10002
Mean	650.56	38.92	5.01	1.53	100083.33
Std	96.66	10.49	2.89	0.58	57508.11
Min	350	18	0.00	1	11.58
25%	584	32	3	1	50983.75
50%	652	37	5	1	100185.24
75%	718	44	7	2	149383.65
Max	850	92	10	4	199992.48

3.7. Missing data

The table below shows missing data (if any)

Table 3: Missing Data

Customer ID	0
Surname	0
Credit Score	0
Geography	1
Gender	0
Age	1
Tenure	0
Balance	0
NumOfProducts	0
HasCrCard	1
IsActiveMember	1
EstimatedSalary	0
Exited	0

3.8. Representative dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	RowNumb	CustomerI	Surname	CreditScor	Geography	Gender	Age	Tenure	Balance	NumOfPro	HasCrCarc	IsActiveMe	EstimatedS	Exited
2	1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.9	1
3	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.6	0
4	3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.6	1
5	4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0
6	5	15737888	Mitchell	850	Spain	Female	43	2	125510.8	1		1	79084.1	0
7	6	15574012	Chu	645	Spain	Male	44	8	113755.8	2	1	0	149756.7	1
8	7	15592531	Bartlett	822		Male	50	7	0	2	1	1	10062.8	0
9	8	15656148	Obinna	376	Germany	Female	29	4	115046.7	4	1	0	119346.9	1
10	9	15792365	He	501	France	Male	44	4	142051.1	2	0		74940.5	0
11	10	15592389	H?	684	France	Male		2	134603.9	1	1	1	71725.73	0
12	11	15767821	Bearce	528	France	Male	31	6	102016.7	2	0	0	80181.12	0
13	12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	0
14	13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.98	0
15	14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.8	0
16	15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.65	0
17	16	15643966	Goforth	616	Germany	Male	45	3	143129.4	2	0	1	64327.26	0
18	17	15737452	Romeo	653	Germany	Male	58	1	132602.9	1	1	0	5097.67	1
19	18	15788218	Hendersor	549	Spain	Female	24	9	0	2	1	1	14406.41	0

Figure 3: representative dataset

3.9. data set preprocessing

As part of the data cleaning process, we began by addressing the null values and duplicated entries, as these were the only issues identified. Since the dataset, sourced from Kaggle and labeled as "clean," required minimal cleaning, this step aligned with both the dataset provider's guidelines and our own analysis upon reviewing the data.

For data transformation, we applied one-hot encoding to the categorical variables. Specifically, for the "Geography" feature, we encoded France as 0, Spain as 1, and Germany as 2. Similarly, the "Gender" feature was encoded with 0 for male and 1 for female

4.Methods

4.1. Feature selection

To select the most relevant features for our model, we began by removing irrelevant features, such as 'RowNumber', 'CustomerId', and 'Surname', which did not contribute to the predictive power of the model. Next, we calculated the R-squared (R^2) score for all remaining features as a baseline. We then compared this baseline score with the importance of different feature sets, derived from various perspectives on customer demographics and their relationship with the bank. The following steps outline our feature selection approach:

1. Customer and Financial Status Features: We first selected features directly related to customer demographics and their financial status, excluding any features related to the bank itself. The final set of features included:
   ['CreditScore', 'Geography', 'Gender', 'Age', 'Balance', 'NumOfProducts', 'IsActiveMember', 'EstimatedSalary'].
2. Customer-Bank Relationship Features: In this step, we focused on features that provided insights into the customer-bank relationship, removing demographic features like Age and Gender. The final feature set for this group included:
   ['CreditScore', 'Geography', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', 'EstimatedSalary'].
3. Customer and Bank-Specific Features: Finally, we selected features that were specifically related to both the customer and the bank, excluding the 'Geography' feature. The resulting feature set included:
   ['CreditScore', 'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', 'EstimatedSalary'].

After selecting these feature sets, we recalculated the R-squared value for each set and plotted the residuals of the independent variables. Features that yielded the highest R-squared scores—indicating the greatest impact on the target class, 'Exited'—were retained for the final model.

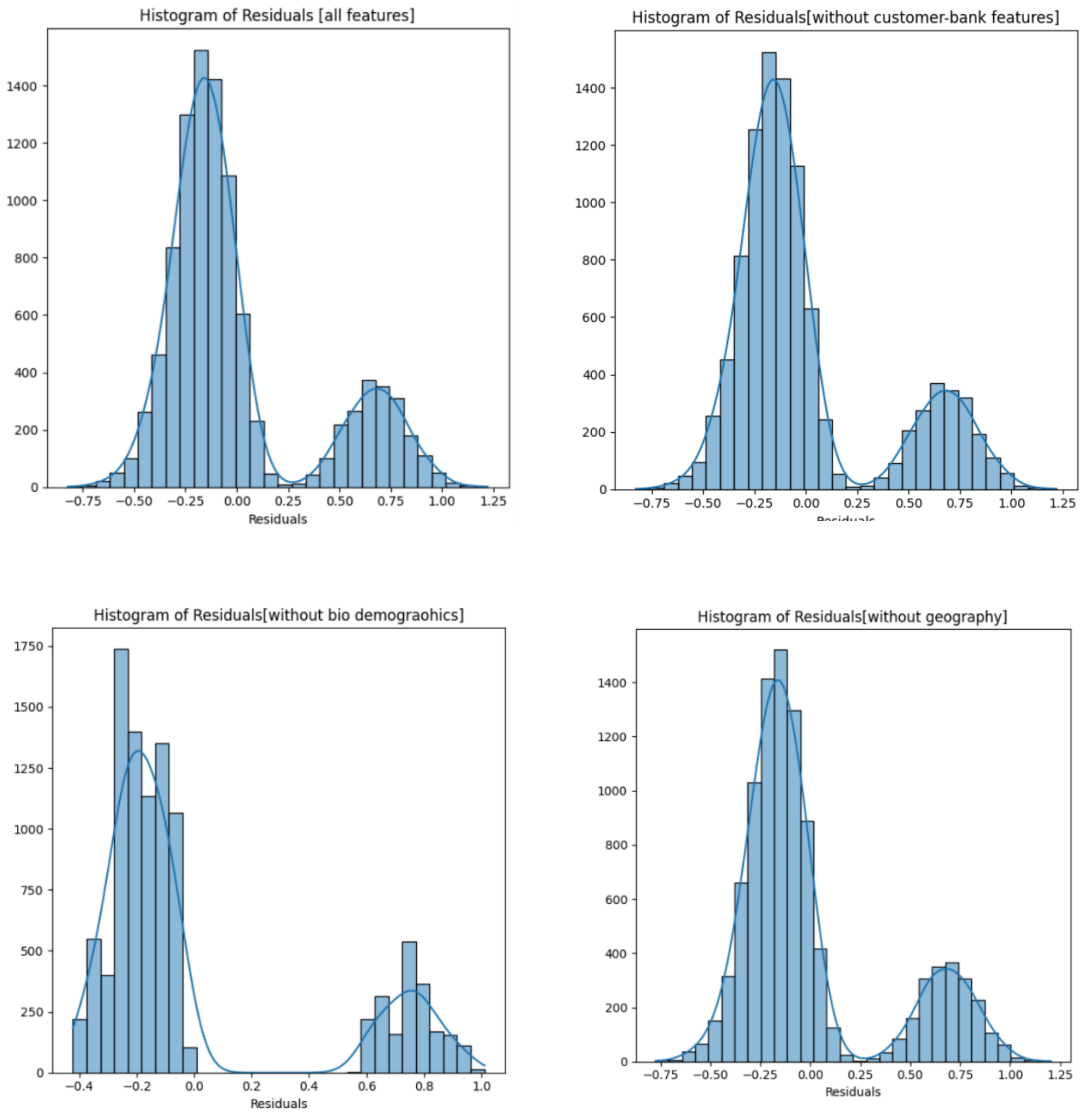


Figure 4: R-Squared figures

we know that R-squared evaluates the scatter of the data points around the fitted regression line.
It is also called the coefficient of determination, or the coefficient of multiple determination for multip le regression.
For the same data set, higher R-squared values represent smaller
differences between the observed data and the fitted values.
R-squared is the percentage of the dependent variable variation that a linear model explains[10].
And as we cannot use R-
squared to determine whether the coefficient estimates and predictions are biased,
we assessed the residual plots shown above to visually see and interpret the residuals. Where graphs are summarized the following table.

Table 4: R-Squared Results Summary

Features	R-Score	Residuals Increasment/Decreasment
allFeatures	0.14707	Standard(base)
WOTenHasfeatures	0.14685	low Increasment
WOGenGeofeatures	0.05368	High Increasment
WOGeofeature	0.13591	Significant Increasment

As shown in the results above, we observed that the R-squared value was highest at 14.71% when considering all features. Additionally, from the bimodal histogram of the residuals (for all features), we infer that the residuals had the smallest values overall, as the peaks are closer to zero compared to other graphs. In conclusion, using all features appears to be the best choice for modeling in this case. However, since the selected features still yielded a relatively low R-squared value, this raises the question:

Are Low R-squared Values Always a Problem?

According to research, regression models with low R-squared values can still be effective for several reasons. Certain fields, such as those analyzing human behaviour, inherently involve greater unpredictability, resulting in lower R-squared values. For instance, studies focused on human behaviour often produce R-squared values under 50%, as human actions are much harder to predict than physical processes[10].

Given this context, we proceeded with the prediction process using the highest R-squared value obtained (14.71%) from our feature selection, with the goal of improving the model's predictive power by exploring more advanced techniques to better capture complex patterns in the data. This approach aims to optimize model performance despite the relatively low R-squared value.

4.2.LR

Logistic Regression (LR) is a machine learning algorithm used for binary classification tasks, such as predicting whether a customer will churn. It predicts the probability of a customer belonging to one of two classes using a sigmoid function, which maps a linear combination of input features to a value between 0 and 1. LR was selected for this project because it is simple, interpretable, and effective for understanding how individual features influence the likelihood of churn. Each feature's coefficient in LR provides insight into its contribution to the prediction, making it a powerful tool for identifying key factors.

To prepare the data for LR, categorical features like Geography and Gender were converted into numerical values using one-hot encoding, and numerical features were normalized to improve model

performance and ensure faster convergence. Feature selection was based on the statistical significance of each attribute, with the final set including CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, and EstimatedSalary. Irrelevant features, such as CustomerId and Surname, were removed to simplify the model and reduce noise. Although dimensionality reduction techniques were not applied, the selected features were sufficient for Logistic Regression to perform effectively without overfitting. This approach ensured a strong foundation for predicting customer churn while maintaining interpretability and efficiency.

4.3.SVM

Support Vector Machines (SVM) are supervised learning models widely used for classification tasks. They are particularly effective in scenarios where the data is high-dimensional or where classes are not linearly separable. SVM works by finding an optimal hyperplane that separates the data into distinct classes, maximizing the margin between the support vectors—data points that are closest to the decision boundary. Two kernel functions were used in this project: the Polynomial Kernel and the Radial Basis Function (RBF) Kernel. The Polynomial Kernel is suited for capturing structured, non-linear relationships in the data. The RBF Kernel, on the other hand, provides the flexibility to model complex decision boundaries, which is essential when customer churn patterns are influenced by subtle and non-linear interactions between features.

The decision to use SVM for the Bank Churn Analysis project is rooted in its ability to handle high-dimensional data and its effectiveness in separating complex classes. Bank churn prediction involves analyzing various customer attributes, including demographic factors (e.g., age, gender), financial status (e.g., balance, estimated salary), and customer-bank relationship indicators (e.g., tenure, number of products, and active membership status). These features interact in ways that may not be linearly separable, making SVM's ability to use kernels critical.

SVM also offers robustness against overfitting.The model's regularization parameter (C) allows us to balance the trade-off between maximizing the margin and minimizing classification errors, ensuring better generalization on unseen data. Furthermore, SVM's kernel flexibility enables us to tailor the model to the data complexity, with the Polynomial Kernel capturing structured relationships and the RBF Kernel addressing intricate, non-linear patterns.

For feature engineering and dimensionality reduction, they’re essential steps to enhance performance. Preprocessing involved removing null values and duplicates, encoding categorical features like Geography (France: 0, Spain: 1, Germany: 2) and Gender (Male: 0, Female: 1), and dropping irrelevant

columns such as RowNumber, CustomerId, and Surname. Feature selection experiments focused on identifying attributes most relevant to customer churn, with the final feature set—including CreditScore, Geography, Gender, Age, Balance, NumOfProducts, IsActiveMember, and EstimatedSalary—yielding the highest R-squared value of 14.71%. To further optimize the SVM, Principal Component Analysis (PCA) was used during hyperparameter tuning to reduce dimensionality while retaining critical data variance. This combination of feature engineering and PCA allowed the SVM to handle high-dimensional data effectively and improve computational efficiency.

4.4 Neural Networks

Neural Networks (NN) are highly advanced machine learning models capable of capturing intricate patterns and relationships in data. Their ability to model complex, non-linear behaviors makes them particularly well-suited for predicting customer churn, a problem that often involves subtle and multi-dimensional feature interactions. In this project, two neural network architectures were explored and compared: the Dense Neural Network (DNN) and the 1D Convolutional Neural Network (1D-CNN). Each was chosen to leverage distinct strengths, providing complementary insights into the problem.

The DNN was selected for its capability to model complex, non-linear relationships between input features. Its architecture comprised two hidden layers with 64 and 32 neurons, each employing the ReLU activation function to introduce non-linearity and improve learning. Batch normalization was applied to stabilize and accelerate training, while dropout layers with a rate of 30% were included to prevent overfitting. The output layer utilized a sigmoid activation function, making it ideal for binary classification tasks like predicting customer churn by outputting the probability of a customer leaving.

On the other hand, the 1D-CNN was designed to detect structured interactions between features, such as localized patterns that might not be as easily captured by a DNN. This architecture included two convolutional layers, each equipped with ReLU activation to extract meaningful relationships between features. Global average pooling was used to condense the information from the convolutional layers into a single vector for classification. Similar to the DNN, batch normalization and dropout layers were applied to improve training stability and reduce overfitting. The output layer of the 1D-CNN also employed a sigmoid activation function for binary classification.

Both models were trained using the Adam optimizer and the binary cross-entropy loss function, ensuring effective learning while handling the imbalanced nature of the dataset. Input features were standardized to ensure consistent scaling, which is crucial for neural networks to converge efficiently. While the DNN excelled in capturing complex non-linear patterns, the 1D-CNN offered a unique

perspective by identifying structured relationships within the features. These complementary strengths were integral to the comparison and evaluation of the two architectures, ultimately guiding the selection of the most suitable model for customer churn prediction.

5.Experiments

5.1.LR

The training and evaluation of Logistic Regression (LR) for predicting customer churn followed a structured process. The data was first prepared by applying one-hot encoding to categorical features like Geography and Gender, while numerical features were normalized to ensure faster convergence and consistent performance. The dataset was split into 80% training and 20% testing sets, and 5-fold cross-validation was used on the training set to validate model stability. Regularization techniques (L1 and L2) were employed to prevent overfitting, with GridSearchCV used to fine-tune hyperparameters, including regularization strength (C), penalty type (l1 or l2), and the number of iterations (max_iter). The best parameters identified were {'C': 100, 'penalty': 'l1', 'max_iter': 200}. Logistic Regression's architecture involves a weighted linear combination of input features passed through a sigmoid function, outputting probabilities to classify customers as churned or not based on a decision threshold (e.g., 0.5). The model's performance was evaluated using accuracy, precision, recall, F1-score, and ROC-AUC, achieving a reasonable AUC score of ~0.76. To balance precision and recall, thresholds between 0.25 and 0.5 were analyzed, prioritizing recall to minimize false negatives. The implementation leveraged Python libraries like pandas, numpy, and scikit-learn, with all computations performed on Google Colab's CPU environment due to the dataset's manageable size.

Hyperparameter Tuning

We performed GridSearchCV to tune the hyperparameters of the Logistic Regression model.

The hyperparameters considered were:

- C: Regularization strength
- Penalty: L1 or L2 regularization
- max_iter: Maximum number of iterations for optimization

The optimal hyperparameters found through GridSearch were:

- C = 10
- Penalty = 'l1'
- max_iter = 500

The selected hyperparameters provided the best model performance during the grid search, yielding a cross-validation score of 0.7641. This means that, on average, the model correctly

predicted the target class 76.41% of the time across different data splits. This score indicates that the model generalizes well, performing consistently across subsets of the data without overfitting, and is likely to perform reliably on unseen data. In conclusion, the high cross-validation score demonstrates that the chosen hyperparameters resulted in a robust model that is well-suited for prediction.

Evaluation Metrics at Different Decision Thresholds

Once the optimal hyperparameters were determined, we evaluated the model at different decision thresholds to determine the best balance between precision and recall. Below are the results of the evaluation at decision thresholds of 0.25, 0.30, 0.35, 0.40, and 0.50.

Showing ROC curve at different thresholds

The curve in graph below shows the model's ability to distinguish between classes. The area under the curve (AUC = 0.76) indicates that the model has a good ability to differentiate between positive and negative cases. The higher the curve, the better the model is at distinguishing between the two outcomes.

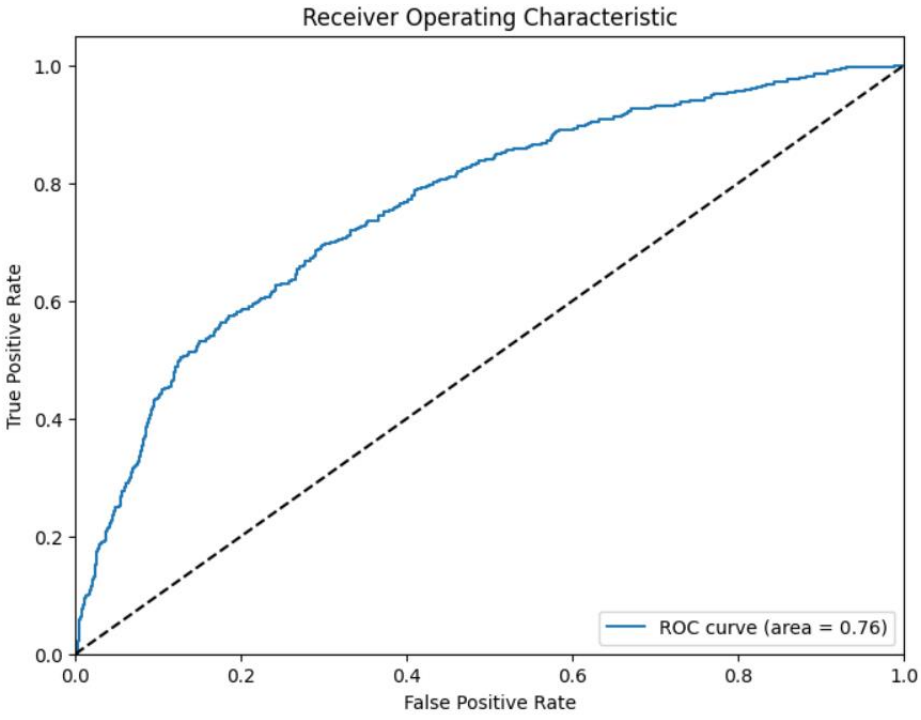


Figure 5: LR ROC

5.1.1 Threshold 0.25

Table 5: Threshold 0.25 Metrics

Metric	**Value**
Accuracy	75.05%
Precision	42.11%
Recall	58.68%
F1-Score	49.03%
ROC-AUC	0.7621

5.1.1.1. Confusion matrix

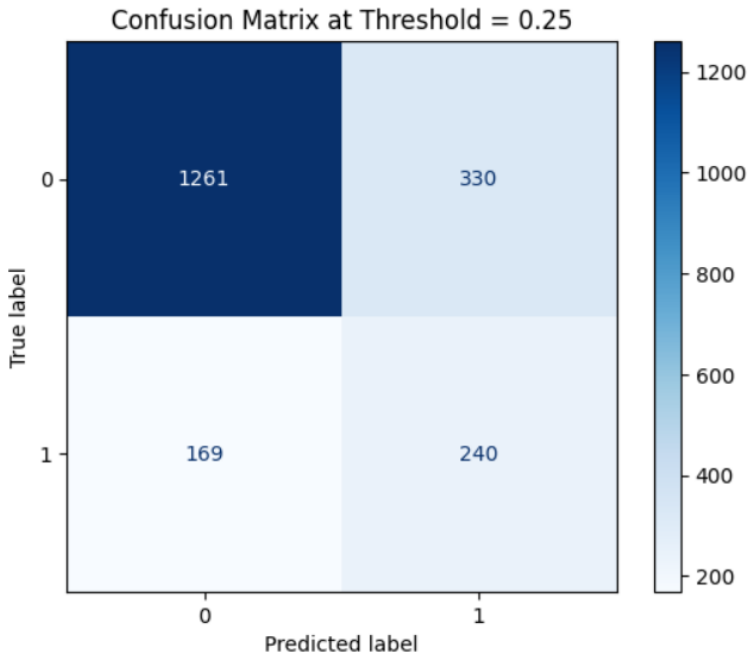


Figure 6: Confusion matrix 0.25

At this threshold, the model's accuracy was 75.05%. The precision was relatively low at 42.11%, but the recall was high at 58.68%, meaning the model was more likely to identify the positive cases but had many false positives. This threshold appears to prioritize minimizing false negatives, but at the cost of increased false positives. The F1-score of 49.03% indicates a balance between the two, though it leans slightly toward recall. The ROC-AUC score of 0.7621 shows the model's ability to distinguish between classes was reasonably strong.

5.1.2. Threshold 0.30

Table 6: Threshold 0.30 Metrics

Metric	Value
Accuracy	78.85%
Precision	48.39%
Recall	51.34%
F1-Score	49.82%
ROC-AUC	0.7621

5.1.2.1. Confusion matrix

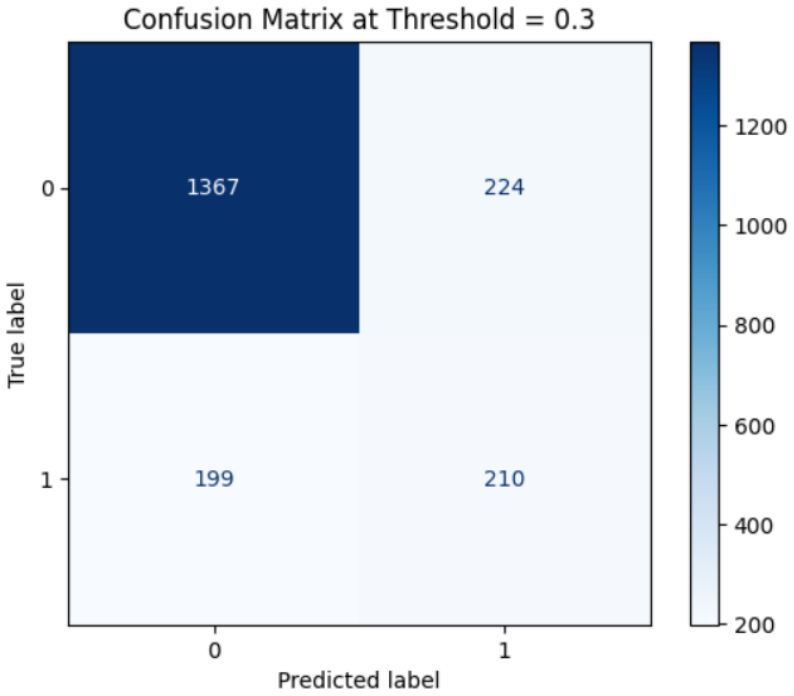


Figure 7: Confusion matrix 0.3

At this threshold, accuracy improved to 78.85%. Both precision and recall remained balanced at 48.39% and 51.34%, respectively, resulting in an F1-score of 49.82%. This threshold slightly reduces the recall compared to the 0.25 threshold, but the model becomes more precise, reducing false positives. The ROC-AUC score remains constant at 0.7621, indicating that the model's ability to distinguish between classes remained stable.

5.1.3. Threshold 0.35

Table 7: Threshold 0.35 Metrics

Metric	Value
Accuracy	80.45%
Precision	52.68%
Recall	43.28%
F1-Score	47.52%
ROC-AUC	0.7621

5.1.3.1. Confusion matrix

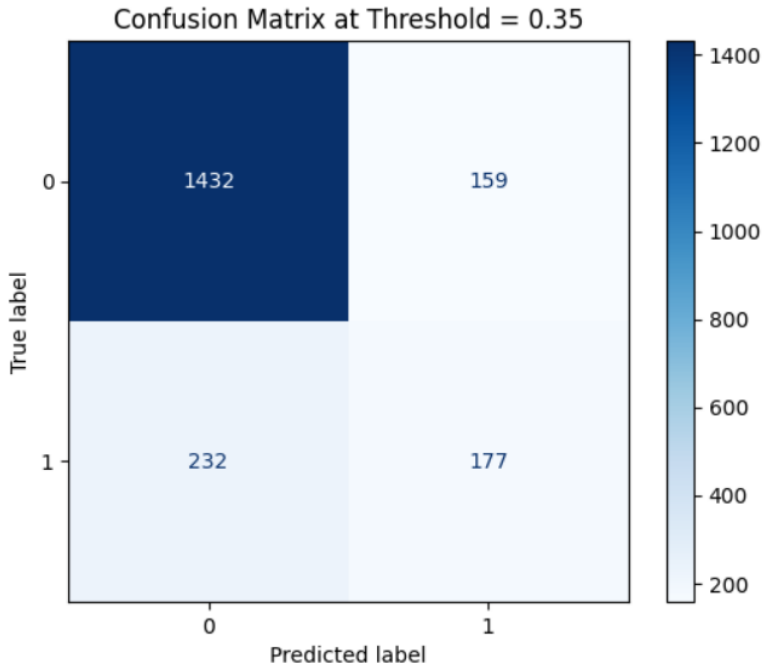


Figure 8: Confusion matrix 0.35

At a threshold of 0.35, the model's accuracy increased to 80.45%, and precision improved to 52.68%. However, recall decreased to 43.28%, reflecting the model's reduced sensitivity to identifying positive cases. The F1-score dropped to 47.52%, indicating a trade-off between precision and recall.

The ROC-AUC score remained unchanged at 0.7621.

5.1.4. Threshold 0.4

Table 8: Threshold 0.40 Metrics

Metric	**Value**
Accuracy	80.30%
Precision	53.01%
Recall	32.27%
F1-Score	40.12%
ROC-AUC	0.7621

5.1.4.1. Confusion matrix

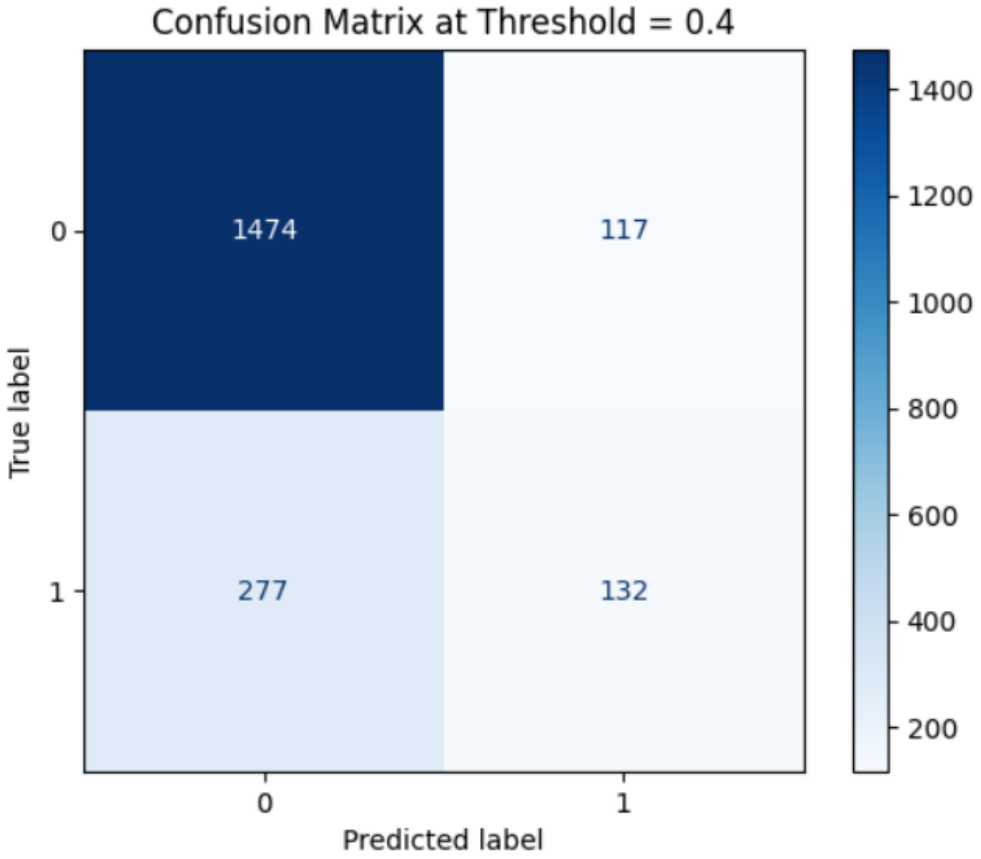


Figure 9: Confusion matrix 0.4

With the decision threshold set to 0.40, the model achieved 80.30% accuracy and 53.01% precision.

However, recall fell to 32.27%, and the F1-score dropped to 40.12%, indicating that the model is now prioritizing precision at the expense of detecting positives. The ROC-AUC score remained stable at 0.7621.

5.1.5. Threshold 0.5

Table 9: Threshold 0.50 Metrics

Metric	**Value**
Accuracy	80.85%
Precision	60.00%
Recall	19.07%
F1-Score	28.94%
ROC-AUC	0.7621

5.1.5.1 Confusion matrix

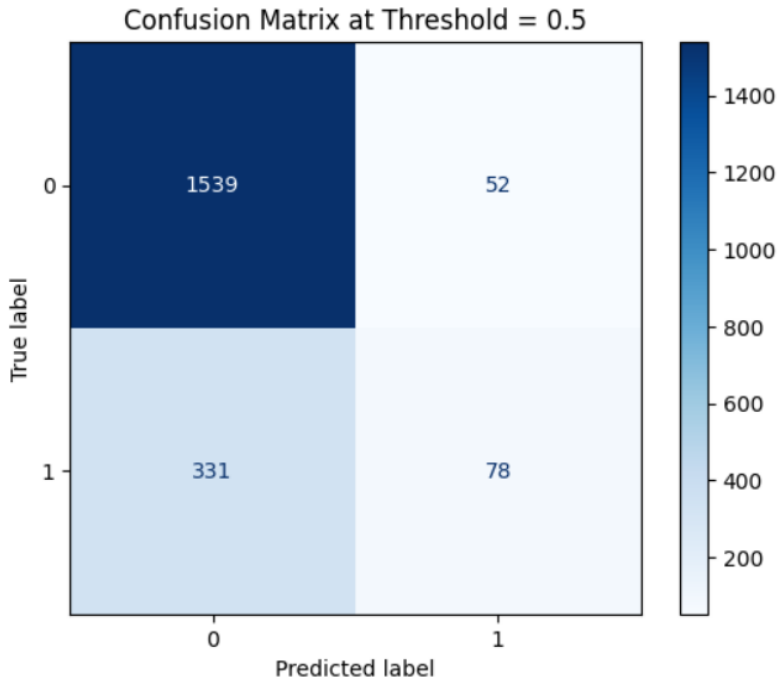


Figure 10: Confusion matrix 0.5

At a threshold of 0.50, the model achieved the highest accuracy of 80.85%, and precision improved to 60.00%. However, the recall dropped significantly to 19.07%, meaning many true positive cases were missed. The F1-score fell to 28.94%, reflecting the imbalance between precision and recall. The ROC-AUC score remained constant at 0.7621

5.2.SVM

5.2.1 Data Splitting

The dataset was divided into training and test sets using an 80-20 split. The training set was used to build the SVM models, while the test set served as unseen data to evaluate their performance. This approach ensures that the models can generalize well to new data, a critical requirement for predicting customer churn effectively.

5.2.2. Model Training

Two SVM models were trained with different kernel functions:

- **Polynomial Kernel**: Configured with degree=3 and C=1. This kernel was used to capture structured, non-linear relationships in the data.
- **RBF Kernel:** Configured with γ=0.005\gamma=0.005 and C=5. The RBF kernel creates more flexible decision boundaries, making it effective for modeling complex, non-linear relationships present in the churn data.

Both models were trained on the preprocessed features, which were normalized to ensure that all variables contributed equally to the model's performance.

5.2.3. Model Testing

The models were evaluated on the test set by generating predictions for each kernel. Both the Polynomial and RBF kernels produced similar results, with nearly identical accuracy and weighted F1 scores. However, the **RBF kernel was chosen as the preferred model** due to its flexibility and lower risk of overfitting.

While the Polynomial kernel captured structured patterns effectively, its fixed complexity could limit its adaptability, especially when applied to new datasets. In contrast, the RBF kernel's capacity to model intricate, non-linear relationships makes it better suited for generalization and robustness in predicting customer churn.

5.2.4. Hyperparameter Tuning

To optimize the SVM models, GridSearchCV was used in combination with PCA (Principal Component Analysis) for dimensionality reduction. The grid search explored the following parameter ranges:

- **C:** [1, 5, 10], which balances model complexity and classification errors.
- γ|*Gamma*: [0.0005, 0.001, 0.005], which controls the influence of individual training points.

The best parameters for the RBF kernel were identified as C=10 and γ=0.005 as they balanced accuracy and generalization. C=10 allowed the model to focus on classifying data correctly, while γ=0.005 captured complex patterns without overfitting.

5.2.5. Performance Evaluation

The models were assessed using accuracy and weighted F1 Score, both of which are crucial for evaluating classification performance.

Table 10: SVM Kernel results

Kernel	Accuracy (%)	F1 Score (%)
Polynomial kernel	79.67%	70.66%
RBF Kernel (best model-before tuning)	79.63%	70.64%
RBF Kernel (after tuning)	77.07%	78.74%[i]

The training and evaluation process was implemented using scikit-learn, with computations performed on a standard CPU. The RBF kernel demonstrated better overall performance, particularly after hyperparameter tuning.

5.3 Neural Networks

5.3.1. Dense Neural Network (DNN)

The DNN was trained with early stopping, ensuring that the model avoided overfitting while maximizing validation performance. Training and Validation Accuracy curves (figure 10) showed close alignment between training and validation accuracy by epoch 10, indicating effective generalization.

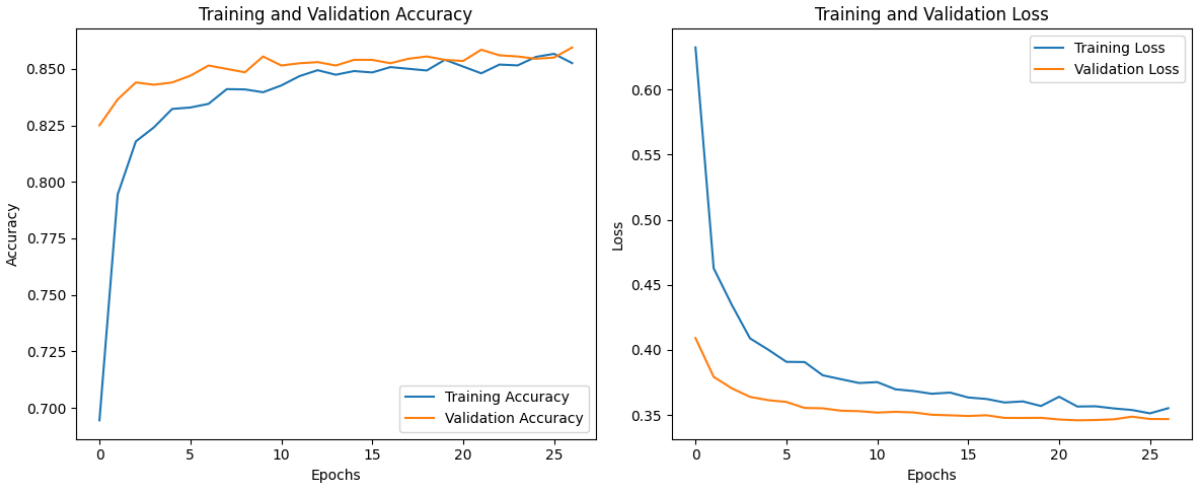


Figure 10 DNN Training and validation

Key Results:

- **Test Accuracy**: 85.35%
- **Epochs**: Training stopped after 25 epochs as validation loss stabilized.
- **Performance Metrics:**
  - **ROC Curve (Figure 11)**: Achieved an AUC of 0.86, indicating strong class separation.
  - **Precision-Recall Curve (Figure 12)**: Average precision score of 0.67, demonstrating high precision, even at lower recall levels.
- **Confusion Matrix (Figure 13)**:
  - True Negatives: 1,524
  - True Positives: 183
  - False Positives: 67
  - False Negatives: 226
- **Precision, Recall, and F1 Score over Thresholds (Figure 14)**:
  - The threshold tuning curve demonstrates the trade-off between precision and recall.
  - An optimal threshold of approximately 0.4–0.5 was identified, where the F1 score peaked, achieving the best balance between precision and recall.
  - At lower thresholds, precision is higher, while recall improves significantly at higher thresholds, showing the model's capability to adapt to different objectives.

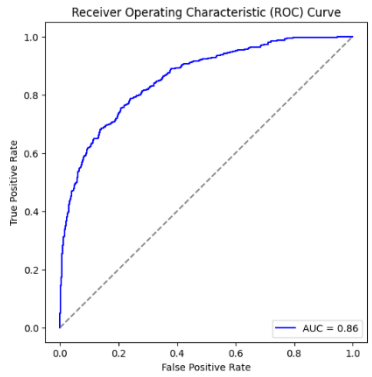


Figure 11 ROC Curve

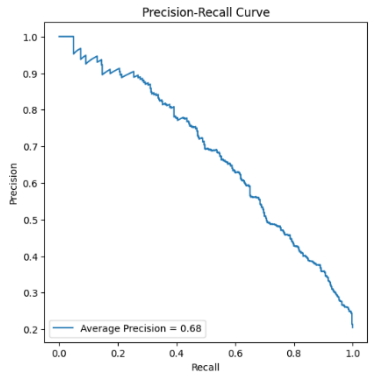


Figure 12 Precision-Recall Curve

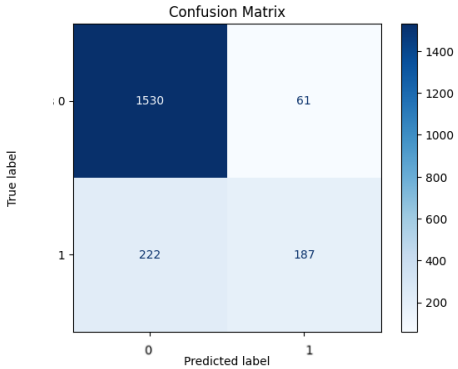


Figure 13 Confusion matrix

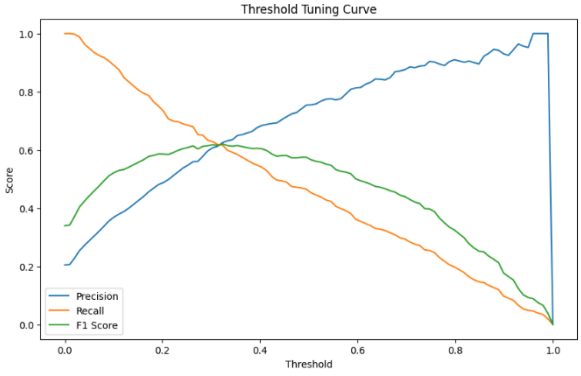


Figure 14 Precision, Recall, and F1 Score over Thresholds

5.3.2. 1D Convolutional Neural Network (1D-CNN)

The 1D-CNN was trained on the same dataset, with early stopping applied to prevent overfitting. Training and Validation Accuracy curves (Figure 9) showed good learning, with validation accuracy aligning with training accuracy by epoch 43.

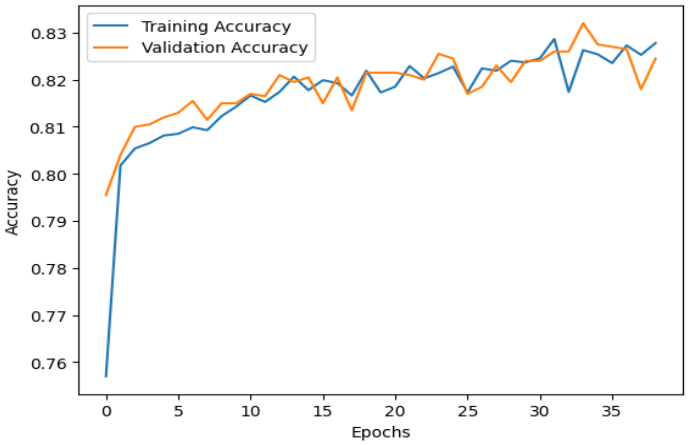


Figure 15: 1D-CNN Training vs Validation

Key Results:

- **Test Accuracy**: 82.90%
- **Epochs**: Training stopped after 43 epochs when validation loss stabilized.
- **Performance Metrics**:
  - **ROC Curve (Figure 15)**: Achieved an AUC of 0.81, demonstrating fair class differentiation.
  - **Precision-Recall Curve (Figure 16)**: Lower average precision compared to the DNN.
- **Confusion Matrix (Figure 12)**:
  - True Negatives: 1,530
  - True Positives: 187
  - False Positives: 61
  - False Negatives: 222
- **Precision, Recall, and F1 Score over Thresholds (Figure 17)**:
  - The threshold tuning curve revealed that the F1 score peaked around **0.45**, indicating the optimal trade-off between precision and recall.
  - The model demonstrated higher recall at increased thresholds, though precision dropped more rapidly compared to the DNN.
  - This highlights the model's tendency to capture more true positives (churners) but at the cost of increased false positives.

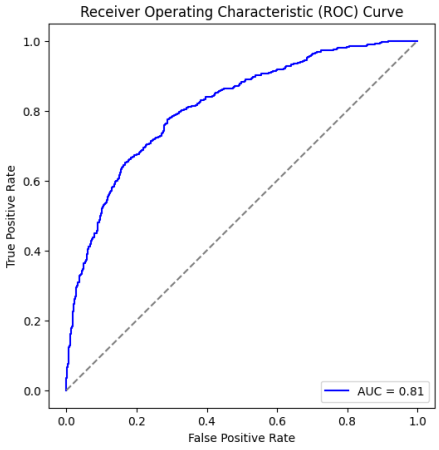


Figure 16: 1D-CNN ROC

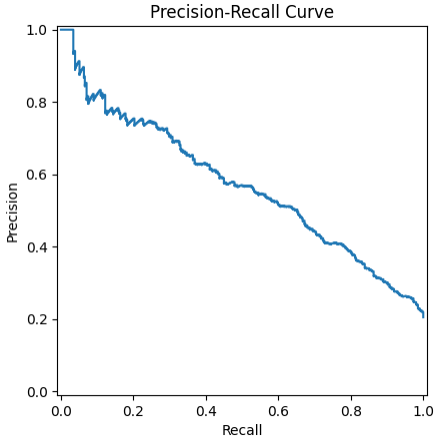


Figure 17: 1-D CNN Precision-Recall Curve

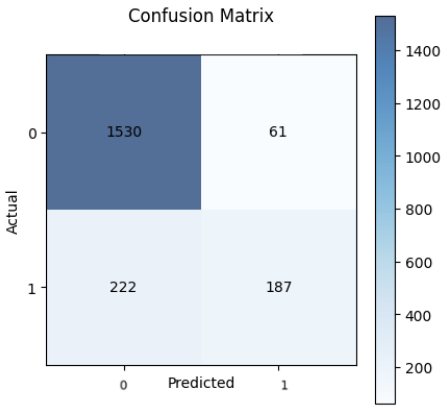


Figure 18: 1D-CNN confusion matrix

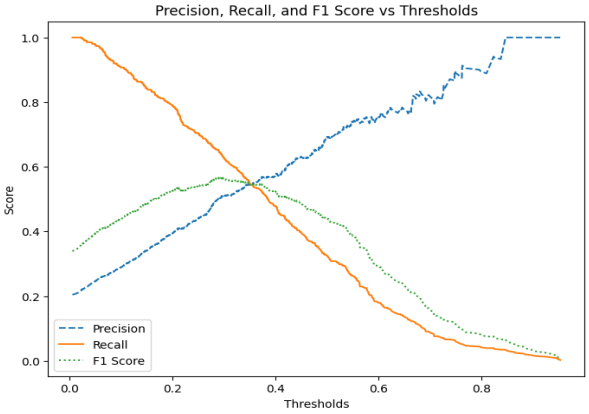


Figure 19: 1D CNN Threshold

5.3.3. Comparison and Model Selection

- **DNN Strengths**:
  - Higher accuracy (85.35%) compared to the 1D-CNN (82.90%).
  - Better balance between precision and recall, with higher AUC and precision-recall scores.
  - Demonstrated stability and generalization across training and validation sets.
- **1D-CNN Strengths**:
  - Extracted unique feature interactions, offering insights into relationships between features.
  - Complemented the DNN by focusing on structured patterns in the data.

Based on the comparison, the **Dense Neural Network (DNN)** was selected as the better-performing model for predicting customer churn. Its superior accuracy, precision-recall balance, and generalization capabilities make it more suitable for this task. However, the insights from the 1D-CNN provide valuable complementary perspectives on the data, suggesting potential avenues for future improvements or hybrid models.

6.Results

In this section, we present the detailed results of the model evaluation across three different models: Linear Regression, Support Vector Machine (SVM), and Neural Networks. We performed a thorough evaluation of each model's performance using various classification metrics, including accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC score. Additionally, we performed hyperparameter tuning where necessary, assessed the model's performance across different decision thresholds, and analyzed how these thresholds impacted the precision-recall trade-offs.

6.1. Linear regression

The obtained results are presented in the following figures. The plot below illustrates the trade-off between precision and recall as the decision threshold is adjusted next to it is plot visualizer of the relationship between the F1-Score, precision, and recall at various decision thresholds.

.

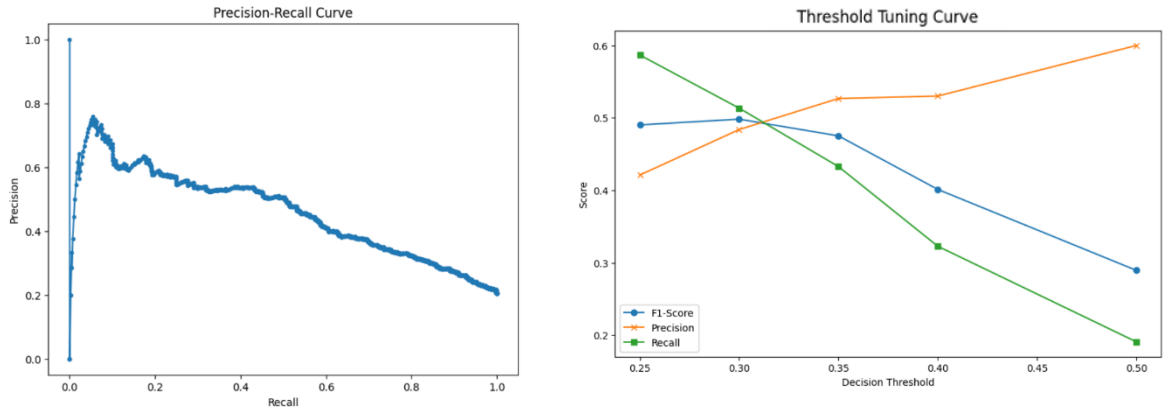


Figure20: LR Precision-Recall and Threshold

Looking at the previously shown confusion matrix (Figure 6), at a threshold of 0.25, we see the model's classification results: 240 true positives, 1261 true negatives, 330 false positives, and 169 false negatives. At this threshold, the model prioritizes recall, which is crucial for our application, as we aim to minimize false negatives (failing to predict exits). Since leaving unseen customers is critically dangerous, minimizing false negatives is key. By selecting a threshold of 0.25, we enhance the model's ability to detect potential exits, even if it slightly reduces precision due to the increase in false positives. This approach is further supported by the precision-recall curve, where we observe that the model starts with high precision, but as recall increases, precision decreases. This indicates that at higher thresholds, the model sacrifices precision to capture more positive cases. Since the goal of this research is to predict customers who are likely to exit before they actually leave, this trade-off is essential. While a good balance between precision and recall is desirable, our primary focus is on maximizing recall to capture as many potential exits (true positives) as possible.

Additionally, the second plot (Figure 20), visualizes the relationship between the F1-Score, precision, and recall across different decision thresholds. At lower thresholds (e.g., 0.25, 0.3, 0.35), recall is

higher, which aligns with our priority of identifying as many positive cases as possible. While the F1-Score peaks around 0.3, suggesting an optimal balance between precision and recall, we yet choose the threshold of 0.25 because it maximizes recall, with its importance explained previously of its significance for our application.

6.2. SVM

The obtained results are presented in tables and visualizations to provide a clear and concise comparison of model performance across different metrics. This includes accuracy, precision, recall, F1-score, and AUC-ROC, as well as confusion matrices and ROC for deeper insights into model behavior.

The table below provides a detailed classification report for the best-performing model, which used the RBF kernel with support vector machines (SVM). The performance of the model on both classes (0 and 1) is shown, as well as the overall accuracy, F1-score, and weighted averages.

Table 11: SVM results

Metric	**Class 0**	**Class 1**	**Macro Avg**	**Weighted Avg**
Precision	0.92	0.46	0.69	0.83
Recall	0.78	0.73	0.75	0.77
F1-Score	0.84	0.56	0.70	0.79
Support	1991	508	2499	2499

Overall Performance:

- **Accuracy**: 77.07%
- **F1 Score**: 78.74%

The confusion matrix below shows the breakdown of correct and incorrect classifications

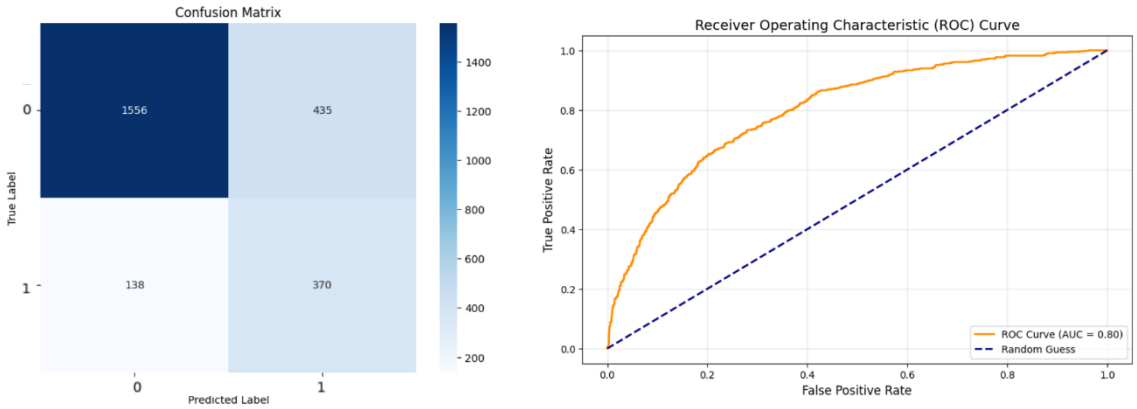


Figure 21: SVM Confusion matrix and ROC

The confusion matrix shown in (Figure 21) reveals the following classification results: 370 true positives, 1556 true negatives, 435 false positives, and 138 false negatives.

In analyzing the overall performance, we observe that the model performs better in terms of precision for Class 0 (non-exiting customers), indicating a high accuracy in predicting customers who are not likely to leave. However, recall for Class 1 (exiting customers) is more critical in this case, as our primary goal is to predict customers who are likely to exit. While the model's precision for non-exiting customers is strong, it sacrifices some precision to capture more true positive cases (exiting customers). This trade-off results in an increase in false positives, but it is a necessary adjustment for minimizing false negatives, which could otherwise lead to missed predictions of exiting customers.

The model's performance demonstrates a reasonable balance between identifying exiting customers (true positives) and avoiding false negatives, which is key in customer retention strategies. Thus, the current model's performance aligns with the objective of detecting potential exits, even if it leads to slightly lower precision in predicting non-exiting customers.

6.3.DNN

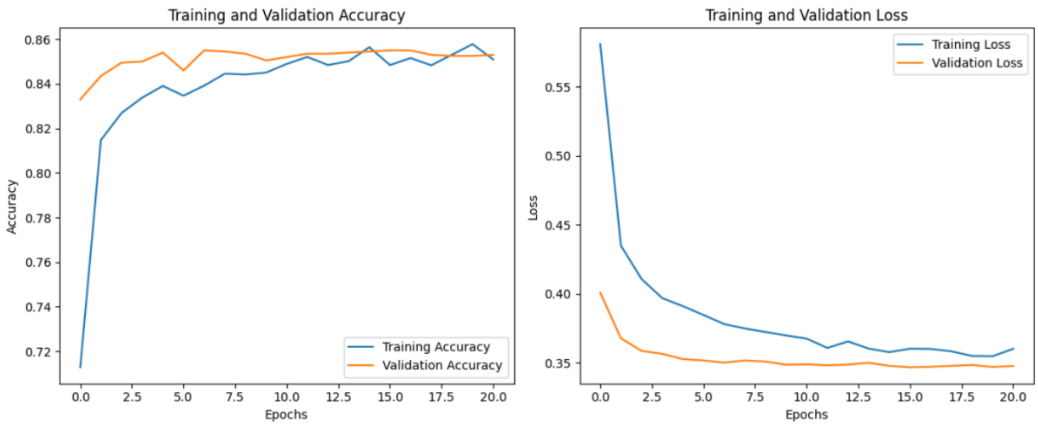


Figure 22: DNN Accuracy, Loss plots

In the above plots (Figure 22), both high training and validation accuracy suggest that the model is learning well and generalizing well to unseen data. Getting the two lines closer to each other by epoch 10, this indicates that the model has minimal overfitting and underfitting, as it performs well on both the training set and validation set. Which leads to the conclusion of

well-tuning as the algorithm has successfully learned the underlying patterns in the data without overfitting to noise or irrelevant features.

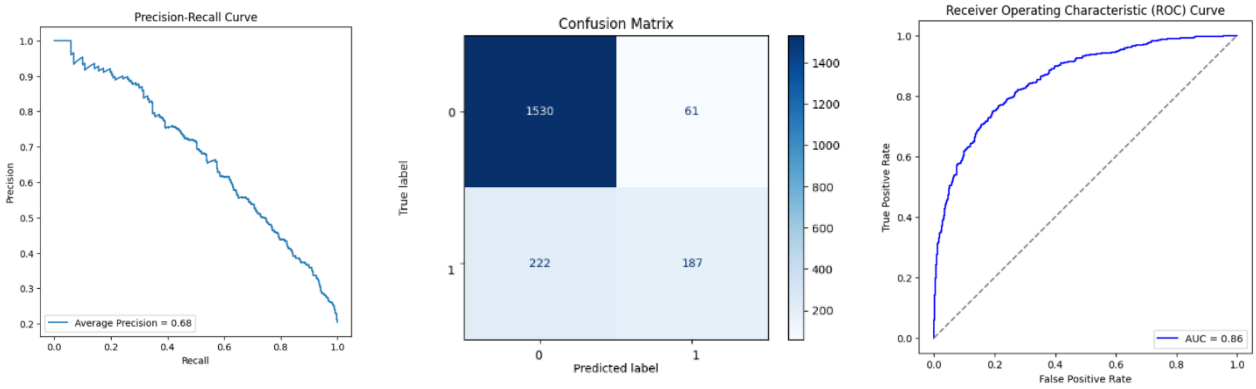


Figure 23: DNN Precision-Recall, confusion matrix, and ROC plots

As shown (Figure 23), in the first graph the curve highlights the trade-off between precision and recall across different decision thresholds. The gradual decline in precision as recall increases indicates the model's ability to balance retrieving relevant positive samples while avoiding false positives.

While the ROC (Receiver Operating Characteristic) curve (last graph in Figure 23) provides a visual representation of the model's ability to differentiate between the classes, with the Area Under the Curve (AUC) value being a key metric for assessing classifier performance. The AUC of 0.86 indicates that the model performs well in distinguishing between the two classes (churners vs. non-churners). The closer the curve is to the top-left corner, the better the model's sensitivity (true positive rate) and specificity (true negative rate), reflecting a strong ability to correctly classify both churners and non-churners.

Looking at the confusion matrix, the model effectively identifies non-churners with 1530 true negatives, but it also makes 222 false negatives—missing some actual churners. This suggests that while the model excels at recognizing customers who are unlikely to churn, it has room to improve in identifying those who are at risk of leaving.

Additionally, the model correctly predicts churn for 187 true positives but also generates 61 false positives, where non-churners are wrongly identified as churners. While false positives are generally less concerning than false negatives in a churn prediction model, reducing them is still important to avoid unnecessary retention efforts for customers who are unlikely to leave.

In summary, the model performs well in identifying non-churners but could benefit from an increased recall to capture more at-risk churners, improving retention efforts. Striking a balance between reducing false positives (minimizing unnecessary interventions) and false negatives (ensuring fewer churners are missed) would enhance the model's overall effectiveness in driving timely and targeted customer retention strategies.

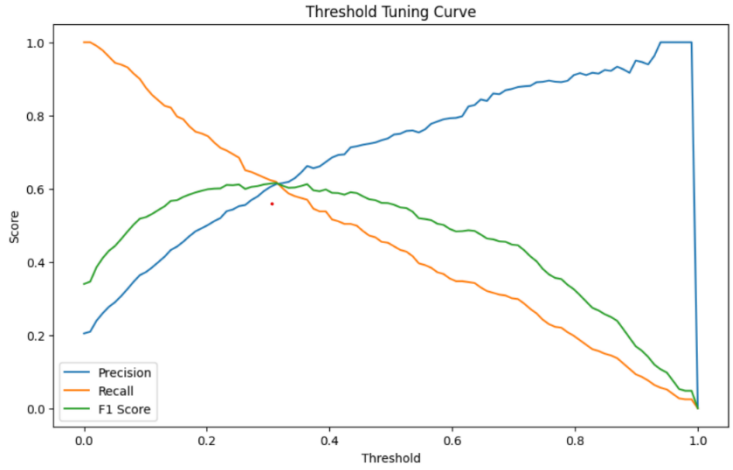


Figure 24: DNN Threshold

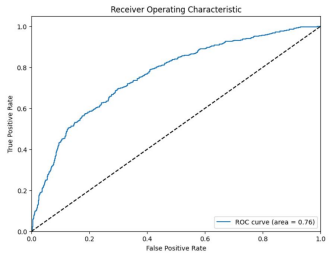
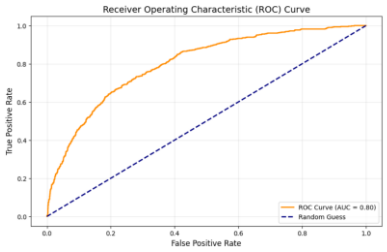
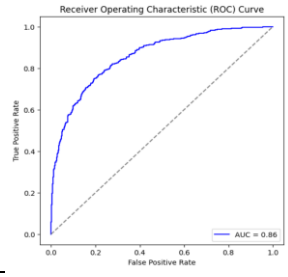
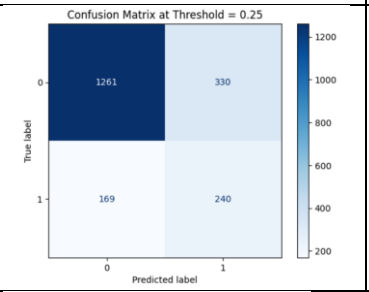
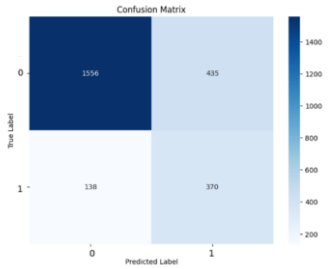
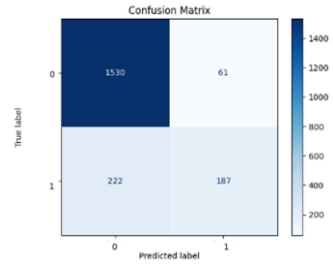
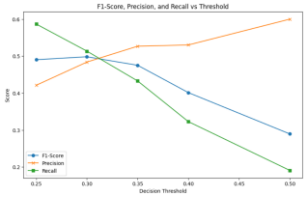
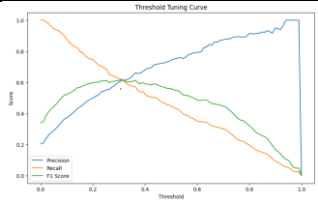
In the last graph in this section, (Figure 24) we observe the trade-off between recall, precision, and F1-score. What is interesting to notice is in the first third of the plot, F1-score rises, indicating an improvement in the balance between recall and precision as the threshold increases. This suggests that the model is better able to correctly classify both positive and negative instances. However, in the second third of the plot, we observe a decline in recall, which negatively impacts the F1-score. Despite recall and precision exhibiting opposite slopes, their combined effect on the F1-score is noticeable. In the last third of the plot, the decline becomes more pronounced, with all three metrics—precision, recall, and F1-score—gradually approaching zero as the threshold reaches 1. This significant drop highlights how the model's performance deteriorates at higher thresholds, with fewer correct classifications overall.

6.4. Highlighting Results

The results of the three machine learning models (Linear Regression, Support Vector Machine (SVM), and Deep Neural Network (DNN)) are presented below. These models were evaluated based on key performance metrics such as **Precision**, **Recall**, **F1-Score**, and **Accuracy**. We also utilized visualizations including **confusion matrices**, **ROC curves**, to better understand how well the models generalize to unseen data.

Below is a table summarizing the performance metrics of the models for both Class 0 (non-exiting) and Class 1 (exiting), as well as the Macro Average and Weighted Average:

Table 12: Models vs Metrics table

	Linear regression (threshold =0.25)	SVM				DNN
		Class 0	Class 1	Macro Avg	Weighted Avg	
F1-Score	49.03%	0.84	0.56	0.70	0.79	55.9%
		Overall: 78.74%				
Precision	42.11%	0.92	0.46	0.69	0.83	73.2%
Recall	58.68%	0.78	0.73	0.75	0.77	44.7%
Accuracy	75.05%	77.07%				85.35%
ROC-AUC	0.7621 	.08 				0.86 
Confusion matrix						
Threshold curve		–				

6.3. Model Generalization to Unseen Data and Interpretation

Before drawing any conclusions, it is important to acknowledge the issue of class imbalance in the dataset. The target classes—**non-exited** (non-churner) and **exited** (customers likely to churn)—are highly imbalanced, with approximately 8,000 instances in the non-exited class and around 2,000 instances in the exited class. Given this imbalance, it is expected that the model will perform better on the majority class (non-exited), as there is simply more data available for training. With this in mind, the following results shall be interpreted with the class imbalance in consideration, while also evaluating the model's performance in identifying the minority class (exited customers) as it weighs more in value. The insights presented here take the imbalance into account, and the choice of the best model is made by putting imbalance and research end **goal** in mind.

LR

Linear Regression (Threshold = 0.25): The Linear Regression model achieved an accuracy of 75.05%, which indicates a relatively good level of performance. However, the model shows a low precision of 42.11% and a moderate recall of 58.68%, which suggests it struggles to correctly identify positive cases (customers likely to exit). Although this model scores better in recall -which serves our goal- its performance indicates that it does not generalize well to unseen data, as its recall and precision are not optimal, leading to a higher rate of false positives and false negatives.

This model is generally less suitable for applications where accurate predictions of customer exits are essential.

SVM

SVM: The SVM model outperforms Linear Regression, achieving an accuracy of 77.07% and an F1-Score of 56.9%. This indicates a better balance between precision and recall. With a precision of 46% and recall of 73%, the SVM model effectively identifies exiting customers while minimizing false positives. The model demonstrates strong generalization to unseen data, as reflected in the high accuracy and balanced precision-recall metrics.

Given that the positive class is underrepresented but still achieves a high recall (73%), we conclude that the SVM's ability to draw a clear decision boundary between the two classes is effective. This suggests that the positive class entities exhibit low intra-class variance and a high inter-class margin, making the separation of classes efficient.

SVM generalizes better out of the three models to unseen data, making it a reliable model for predicting customer exits.

DNN

DNN: The **DNN** model achieved the highest **accuracy** at **85.35%** and a **high precision** of **73.2%**. However, the **recall** is **44.7%**, which is relatively low, indicating that the model misses a significant number of exiting customers. Despite this, at first one might think **DNN** generalizes well to unseen data, by its high **accuracy** and **precision**, yest this conclusion is misleading knowing that in the case of imbalanced datasets, accuracy can be deceptive, as it does not account for the poor recall of the minority class (exiting customers in this case). The low recall suggests that the model may be underperforming in detecting the critical instances, despite its seemingly good overall performance.

Thus, while the DNN model appears strong, its ability to generalize, especially for the minority class, requires further improvement.

7. conclusion

Key Insights

1. **Precision vs. Recall Trade-off**: A key insight from the comparison of these models is the precision-recall trade-off. While the DNN model achieves the highest precision, it sacrifices recall, resulting in a substantial number of missed exits (false negatives). Similarly, Linear Regression also struggles with precision and recall, indicating a tendency to overlook potential exits. On the other hand, SVM strikes a better balance between precision and recall, as reflected by its high F1-score. This makes SVM a more suitable choice for applications where both false positives and false negatives must be

carefully managed, especially when the positive class (exiting customers) holds higher importance and needs to be prioritized.

2. **The No Free Lunch (NFL) Theorem:** suggests that no single machine learning algorithm is universally superior for all types of problems. In other words, if we average the performance of an algorithm over all possible problems, it will perform no better than a random algorithm. This means that the effectiveness of any given model is problem-dependent, and there is no "one-size-fits-all" solution. The theorem highlights that for every learning algorithm, there exists a problem for which it is particularly well-suited, as well as problems where it performs poorly. There is no algorithm that will always perform the best across every domain, and the key to successful model selection is understanding the characteristics of the problem you're working on. In practice, this implies that the model selection process should take into account the specific nature of the data and the task. For example, Neural Networks (NNs) are well-suited for high-dimensional data like images, where feature extraction is complex, while Support Vector Machines (SVMs) are more effective for smaller datasets with well-defined boundaries. SVMs are convex, making them reliable and easier to solve deterministically, but they require knowledge of the appropriate kernel. In contrast, NNs automatically learn useful features through backpropagation, making them flexible and effective for complex data. However, NNs are computationally intensive and require careful tuning of hyperparameters, such as architecture and learning rate, to avoid convergence issues. While SVMs excel in deterministic and mathematically rigorous problem settings, NNs scale well with large datasets and are more robust when trained on GPUs. Ultimately, the choice between SVM and NN depends on the dataset size, complexity, and task requirements, as there is no "one-size-fits-all" solution.[13] With discussion above, we answer the question that might have arisen: ***why did DNN score worse than a baseline model SVM?***

In conclusion this study investigates the performance of three machine learning models—**Linear Regression**, **Support Vector Machine (SVM)**, and **Deep Neural Networks (DNN)**—in predicting customer churn (exit likelihood) for a banking application. The models were evaluated based on their ability to predict whether a customer will exit, with a particular emphasis on balancing precision and recall due to the imbalanced nature of the dataset.

Among the models tested, the **SVM** exhibited the highest overall accuracy of 77.07%, with an F1-Score of 78.74%, making it the most reliable model for predicting customer exits. However, it is important to acknowledge the challenges we faced during the modelling process. One such challenge was the **low R-squared values** observed during feature selection. Despite attempting to refine the model by carefully selecting relevant features, the R-squared values remained low, indicating that the models had difficulty capturing the full complexity of customer behaviour with the available features. This limitation suggests that while the selected features contributed to the model, additional or more refined features might be required to improve prediction accuracy.

Another challenge was encountered with the **DNN model**, which, although promising in its accuracy performance, it yielded relatively low F1-scire and recall results. This highlights the sensitivity of neural networks to hyperparameters and training configurations, and how careful tuning and cross-validation are essential in achieving optimal results. Additionally, the **SVM** model also performed well with an accuracy of 77.07%, demonstrating a good balance between precision and recall, though it still faced difficulties in predicting the minority class (exited customers) as effectively as what we aimed our chosen model to be.

The **Linear Regression** model, while showing an accuracy of 75.05%, struggled significantly with predicting the exited customers, reflecting the limitations of using a linear model for a binary classification task, particularly in imbalanced datasets where nonlinear relationships might exist. The **class imbalance** issue, where the non-exited class was much larger than the exited class, exacerbated the challenges faced by all models. This imbalance skewed the models towards predicting the majority class more accurately, making it harder to accurately identify the exited customers.

Despite these challenges, it is important to note that while **accuracy** is a standard metric, **recall** for the exited class was prioritized in this research, as accurately identifying customers likely to leave is the primary goal. The results clearly highlight the trade-off between precision and recall, and emphasize the need to adjust decision thresholds in order to achieve the best balance for the business context.

For future improvement, further research into hyperparameter tuning, feature engineering, and techniques to address class imbalance (e.g., resampling or class weighting) could help improve

model performance. Additionally, exploring more sophisticated models and incorporating external factors such as customer sentiment could further enhance predictive accuracy. This study demonstrates the potential of machine learning in predicting customer exits, but addressing the highlighted limitations will be crucial to improving the robustness and reliability of these models.

8. Appendix:

Contributions

Table 13: Contribution

Tasks done by student	**Name of student**
Revision Motivation: NN Revision Experiment: NN	Aisha Alsaggaf
Motivation Motivation: LR Experiment: LR	Dimah Alharbi
Background Motivation: NN Experiment: NN	Shaden Alturki
Introduction Motivation: SVM Experiment: SVM	Athbah Alaliwei
Abstract Dataset Experiment: LR Results Conclusion Report Format	Ghaina Alhassnan

9. References

References:

- [1] IBM, "What is Customer Churn?" Available: https://www.ibm.com/think/topics/customer-churn.
- [2] McKinsey & Company, "Experience-Led Growth: A New Way to Create Value," March 2023. Available: https://www.mckinsey.com/capabilities/growth-marketing-and-sales/our-insights/experience-led-growth-a-new-way-to-create-value.
- [3] Pecan AI, "Predictive Modeling," Available: https://www.pecan.ai/blog/predictive-modeling/
- [4] Invesp, "Customer Acquisition Vs. Retention Costs – Statistics And Trends," 2020. Available: https://www.invespcro.com/blog/customer-acquisition-retention-costs/.
- [5] Customer Success Collective, "Customer retention vs. acquisition: What's the best method?," 2021. Available: https://www.customersuccesscollective.com/.
- [6] Kaggle (2021). Telco Customer Churn Dataset. Available at: https://www.kaggle.com/datasets/blastchar/telco-customer-churn
- [7] Kaggle (2023). Online Retail Customer Churn Dataset. Available at: https://www.kaggle.com/datasets/hassaneskikri/online-retail-customer-churn-dataset
- [8] Kaggle (2023). Predictive Analytics for Customer Churn Dataset. Available at: https://www.kaggle.com/datasets/safrin03/predictive-analytics-for-customer-churn-dataset
- [9] Meshram, S. (2023) Bank customer churn prediction, Kaggle. Available at: https://www.kaggle.com/datasets/shubhammeshram579/bank-customer-churn-prediction (Accessed: 04 September 2024).
- [10] “How To Interpret R-squared in Regression Analysis,” Statisticsbyjim. https://statisticsbyjim.com/regression/interpret-r-squared-regression/
- [11] GeeksforGeeks (2024) What is data munging?, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/what-is-data-munging/#:~:text=Data%20munging%2C%20sometimes%20called%20data%20wrangling%20or%20data,and%20value%20for%20various%20downstream%20uses%2C%20including%20analytics. (Accessed: 15 September 2024).
- [12] Goble, S. (2022) Data selection for nascent data scientists, Medium. Available at: https://towardsdatascience.com/data-selection-for-nascent-data-scientists-2fb27f534723 (Accessed: 15 September 2024).

- [13]“Are neural networks better than SVMs?,” Cross Validated. https://stats.stackexchange.com/questions/510052/are-neural-networks-better-than-svms

- [14] Kagglex Fellowship Program Kaggle. Available at: https://www.kaggle.com/kagglex (Accessed: 10 September 2024).

i