Yuleica Ledezma
UCB Data Analytics Bootcamp April-Oct 2024

Project Title:

Optimizing Lead Conversion with Machine Learning Models

Executive Summary:

This project's goal is to develop a predictive model to identify high-conversion leads, improving the current lead conversion rate from 38%. Currently, the sales team relies on manual lead scoring based on limited data and employee intuition, which leads to inefficiencies and missed opportunities. By building a data-driven model that assigns a score to each lead, the team can prioritize high-potential prospects and allocate their time more effectively, leading to increased efficiency and conversion rates.

The success of this model will be evaluated not only by its **accuracy**, but by its **precision** and **recall**:

- **Precision** ensures the model correctly identifies leads most likely to convert, reducing time spent on false positives. This allows the sales team to focus on high-value prospects, maximizing efficiency.
- **Recall** ensures that the model captures a wide range of potential leads, ensuring that valuable opportunities are not missed, optimizing the overall conversion funnel.

By leveraging predictive modeling, this project aims to improve both the accuracy and efficiency of lead prioritization, ultimately enhancing the sales funnel and boosting conversion rates. This will significantly improve over the current manual scoring process, which is based on subjective judgment rather than data-driven insights.

Problem Statement:

X Education markets online courses to potential customers through various digital channels. However, only **38%** of captured leads convert into paying customers, largely due to the current approach of manually scoring leads based on incomplete data and employee hunches. This process is both time-consuming and prone to error, as it overlooks key patterns and relationships hidden within the data.

The aim of this project is to develop a predictive model that assigns a conversion score to each lead, based on data-driven insights. By doing so, the sales team will be able to focus on leads with the highest likelihood of converting, optimizing their efforts and reducing time spent on low-potential prospects. Additionally, the model will help improve the **lead conversion funnel**, ensuring that leads are more efficiently filtered and nurtured throughout the sales process.

Yuleica Ledezma
UCB Data Analytics Bootcamp April-Oct 2024

Objectives:

- Develop a predictive lead scoring model to improve lead prioritization.

Data Overview:

The dataset used includes lead data sourced from CSV file (9,241 rows x 37 columns) from Kaggel, capturing features like lead origin, engagement metrics, and historical conversion outcomes. Data types are mostly categorical and numerical, including variables such as time spent on the website and response to marketing campaigns. The dataset will be cleaned and preprocessed to ensure accuracy before modeling.

Methodology:

The project involves:

- **Data Cleaning and Preprocessing**: Handling missing values, outliers, and categorical data encoding.
- **Exploratory Data Analysis (EDA)**: distributions, and correlations between features.
- **Feature Engineering**: Creating new variables based on user engagement levels.
- **Model Building**: will test multiple models, including Logistic Regression (as a baseline), Random Forest, and Neural Networks. Hyperparameter tuning will be used to optimize each model.
- **Model Evaluation**: Precision, recall, and F1 score will be key metrics to evaluate the model's effectiveness in identifying high-conversion leads.

Tools and Technologies:

- **Languages**: Python
- **Libraries**: Pandas, Scikit-learn, Matplotlib, TensorFlow/Keras,
- **Tools**: Jupyter Notebook, GitHub, Kaggle (dataset), Keras Tuner (for hyperparameter optimization)
- **Machine Learning**: Logistic Regression,Logistic Regression PCA, Random Forest, RF Hyperparameter tuning, Neural Networks (with Keras Tuner)

Project Timeline:

- **09/22 - 09/28 Week 1**: Data collection, cleaning, and EDA
- **09/29 - 10/05 Week 2**: Model building and testing
- **10/06 - 10/12 Week 3**: Model evaluation and optimization
- **10/13 - 10/17 Week 4**: Final report and presentation

Key Deliverables:

- A cleaned, preprocessed dataset ready for modeling.

- A trained and evaluated model with performance metrics like precision, recall, F1 score, and accuracy.
- A report summarizing findings, insights, and recommendations (within the Jupyter notebook)
- Visualizations of the model's performance and key insights.
- A presentation for stakeholders to showcase the model's impact.

Success Criteria:

- Achieving model accuracy above 85% (targeting 94% or higher, based on preliminary tests with various models).
- High precision and recall scores, ensuring the model effectively prioritizes high-conversion leads.
- Reduction in time spent on low-potential leads, enabling the sales team to focus on quality prospects.

Risk Management:

- **Data Quality Issues**: Ensuring the dataset is thoroughly cleaned and validated.
- **Overfitting**: hyperparameter tuning to prevent overfitting and ensure generalization.

Budget and Resources:

- **Software**: Free Python libraries and Kaggle.
- **Human Resources**: Data analyst and collaboration with the sales team for feedback and integration (in theory)

Stakeholders:

- **Data Analyst**: Responsible for model development and testing
- **Sales and Marketing Team**: End users of the lead scoring system
- **Upper Management**: Overseeing the project's alignment with business goals

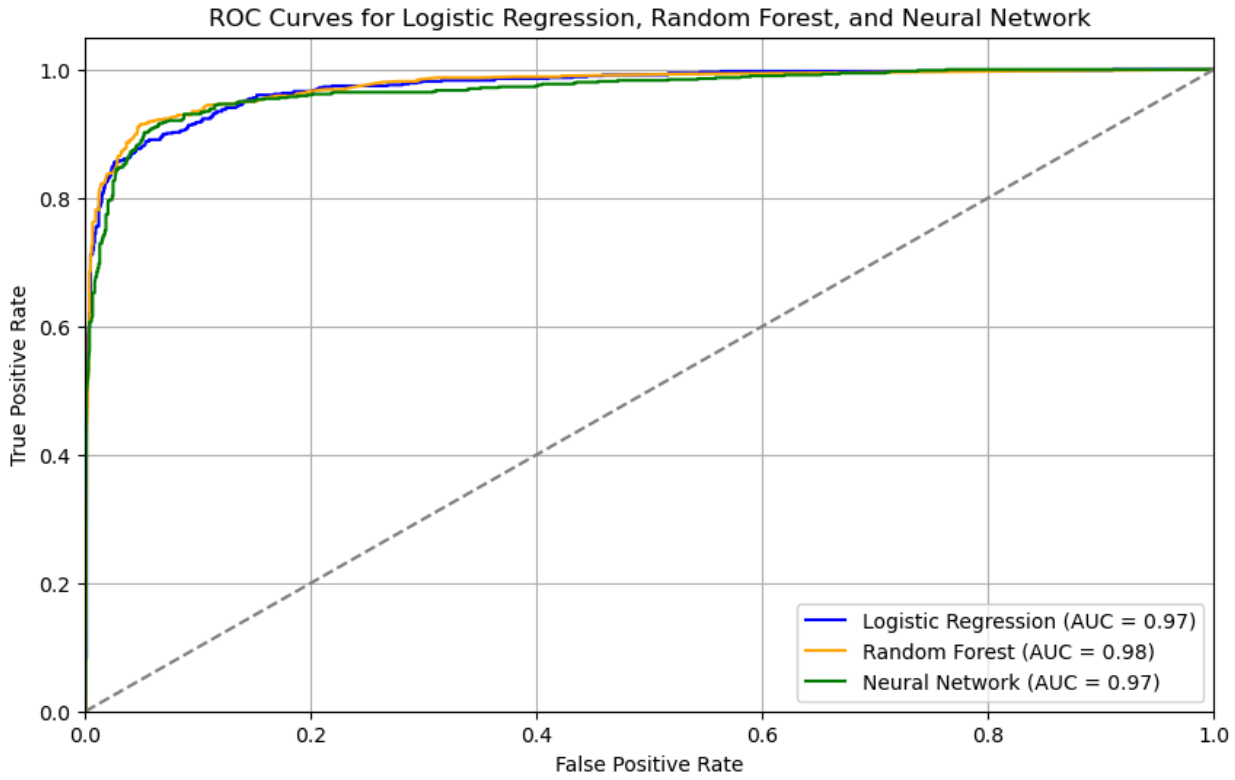
Ethics and Compliance:

- Ensure the data is handled according to relevant privacy laws (e.g., GDPR), and all predictions are made with ethical considerations, particularly in terms of data privacy and bias.

Conclusion:

This proposal outlines the development of a predictive lead scoring model to optimize the sales process at X Education. By leveraging machine learning, the model will help prioritize high-quality leads, reduce wasted effort, and increase overall conversion rates. The outcome is expected to significantly enhance sales efficiency and drive revenue growth through targeted lead engagement.

Yuleica Ledezma
UCB Data Analytics Bootcamp April-Oct 2024



	Log Regression(baseline)	Log Reg PCA	Random Forest	RF Hyperparameter Tuning	Deep Learning	Keras Tuner
Accuracy	.92	.92	.93	.94	.92	.94
Precision	.92	.93	.92	.94	.91	.95
Recall	.87	.87	.90	.90	.90	.89
F1 Score	.89	.90	.91	.92	.91	.91

Conclusion:

The goal of this project was to boost lead conversion rates (which started at 38%) by building a model that could better predict which leads would convert. Multiple models were tested, including Logistic Regression (as a baseline), PCA optimization for the Logistic Regression model, Random Forest, Hyperparameter Tuning with GridSearchCV, Deep Learning, and Keras Tuner. Given the ROC curves and results for Accuracy, Precision, Recall, and F1, here are some key observations:

Process Overview:

The dataset used for this project was large, with 9,241 rows and 37 columns, containing both categorical and numerical data. Each column was carefully analyzed during **Exploratory Data Analysis (EDA)** to understand distributions, correlations, and data patterns. **Feature Engineering** was performed to create new variables based on user engagement levels, improving the predictive power of the models. Data preprocessing included:

- **Numerical scaling**: Ensuring that numeric features were standardized using scaling techniques.
- **Categorical encoding**: Applying one-hot encoding to handle categorical features, making them usable in machine learning models.

Three primary models were implemented: Logistic Regression, Random Forest, and Deep Learning, with hyperparameter tuning applied to optimize their performance.

Class Imbalance:

A key challenge in the dataset was a significant class imbalance, with **5,594 "not converted" leads** compared to **3,425 "converted" leads**. This imbalance could have impacted the model's performance, particularly for Logistic Regression, as it tends to be biased toward the majority class. Therefore, we prioritized evaluation metrics like **Precision, Recall**, and **F1 Score** to account for the imbalance and measure the model's effectiveness in identifying leads that would convert.

Despite the class imbalance, all models performed well due to the careful **feature engineering** and **data preprocessing** steps taken. The models were further optimized with hyperparameter tuning, leading to even better results. The success criteria set at the beginning—achieving over **85% accuracy**—was comfortably met by all models.

Accuracy:

Most models achieved high accuracy, ranging from **92% to 94%**. While accuracy is an important indicator, focusing on it alone doesn't provide a full understanding of the model's business impact, especially in the context of lead conversion.

Precision and Recall:

- **Precision**: Particularly crucial for sales lead generation, precision was as high as **95%** in the Keras Tuner model. This means that fewer false positives (leads unlikely to convert) were predicted, streamlining the sales process and reducing unnecessary effort. The team can focus more on high-quality leads, saving both time and resources.
- **Recall**: Models like Random Forest, Hyperparameter Tuning, and Deep Learning had recall scores as high as **90%**, meaning they successfully captured a large portion of leads that could convert. This ensures that valuable opportunities were not missed.

F1 Score:

The **F1 score**, which balances precision and recall, was solid across all models (up to **92%**), indicating a good overall balance between correctly identifying converting leads and avoiding false positives.

Business Impact:

This model helps the sales team focus on the best leads, leading to more conversions with less effort. Without a model, the sales team might treat all leads equally, wasting time on less likely prospects. By using the model's predictions, the team can prioritize high-probability leads, working more efficiently and increasing conversions.

- **Fewer False Positives**: High precision means sales reps spend less time chasing dead-end leads, allowing them to zero in on the ones that are truly likely to convert.
- **Higher Coverage of Potential Leads**: High recall ensures the team isn't missing out on any important opportunities.

This leads to **cost savings**, as the sales team can work more efficiently, focusing their efforts where they matter most. Automating the lead scoring process also eliminates the need for manual lead qualification, lowering operational costs.

Recommendation:

Given the results, the **Random Forest model with Hyperparameter Tuning** is recommended. It strikes the best balance between precision, recall, and accuracy, making it the optimal solution to help the sales team prioritize high-quality leads and boost conversions while saving resources. Additionally, despite the initial class imbalance in the dataset, the models performed exceptionally well due to thorough analysis, feature engineering, and tuning efforts.

This success meets the original project criteria, with all models achieving the target of **over 85% accuracy**.

Code Schema/Table of Contents

- Import Dependencies
- Data Preprocessing
    - Data Loading
    - Data Inspection
    - Exploratory Data Analysis (EDA)
        - Handling "Select" Values
        - Standardizing Column Names
        - Calculating Missing Values
        - Dropping Columns with Excessive Missing Data
        - Individual Feature Analysis
        - Correlation Heatmap of Numerical Features
    - Clean Data Frame
    - Feature Engineering and Transformation
        - Numerical Scaling
        - Categorical Encoding
- Model Implementation
    - Define Target Variable
    - Calculate Baseline
        - Conversion Ratio
        - Naive Prediction
    - Supervised Learning
        - Logistic Regression
            - Baseline Logistic Regression
            - Logistic Regression Optimization
            - PCA
        - Random Forest
            - Random Forests Improvement from Baseline Logistic Regression
            - Random Forest Visualizations
                - Feature Importance Visualization
                - ROC Curves Comparison (Logistic Regression and Random Forest)
            - Random Forest Optimization
                - Overfitting Random Forest
                - Hyperparameter Tuning GridSearchCV
                - Feature Selection
        - Deep Learning (Neural Networks)
            - Deep Learning Optimization (Keras Tuner)
            - ROC Curves Logistic Regression, Random Forest and Neural Network
- Conclusions
    - Model Performance