**ANL588 FINAL REPORT**

**Propensity to Buy Prediction: Lead Conversion in EdTech**

**A logo with red and blue letters

Description automatically generated**

**Submitted by**

**Hou Heng Sheng**

**SCHOOL OF BUSINESS**

**Singapore University of**

**Social Sciences**

**Presented to Singapore University of Social Sciences in partial fulfillment of the requirements for the**

**Degree of [Graduate Diploma in/Master of] Analytics and Visualisation**

**2023**

**ABSTRACT**

Amid the surging demand for digital skills, the EdTech sector has become pivotal in bridging the educational gaps. This project aimed to optimize lead conversion for an assumed EdTech company, "EdTech Inc.", by using data analytics and machine learning. By employing a dataset containing 37 columns and 9,240 rows, sourced from Kaggle, challenges such as missing values, categorical attributes, and outliers were meticulously addressed. An initial exploratory data analysis revealed essential trends, including the high effectiveness of SMS communication and the high conversion rates among employed professionals. A Random Forest model was then deployed, achieving an precision score of 90.3% on the test set. The most influential predictor for lead conversion was found to be the 'Total Time Spent on Website'. Finally, to facilitate practical applications, an interactive Tableau dashboard was designed, empowering sales teams to prioritize leads and merge their intuitive experience with data-driven insights. The study underscores the importance of tailored communication strategies, targeted marketing, and an efficient online presence in enhancing lead conversion in the EdTech sector.

**TABLE OF CONTENTS**

1. Introduction 4

2. Background 5

3. Methodology 7

4. Data 9

5. Data Wrangling 14

6. Exploratory Data Analysis 18

7. Random Forest 22

8. Tableau Dashboard 25

9. Findings and Recommendations 30

10. Conclusion 36

References 37

Appendix 39

1. **Introduction**

External global forces such as COVID-19 and advancements in artificial intelligence (AI) have created an increased emphasis on digitalization. The result has been a reconfiguration of traditional job roles and the emergence of a pressing demand for digital skills, altering the landscape of employability. This phenomenon has resulted in an unprecedented demand for digital skills. Which has led to the rise of Education Technology (EdTech) industry as more people race to such companies to pick up new skills to stay relevant and increase their employability. The EdTech sector has effectively leveraged digital platforms to dissolve the geographical and financial barriers that previously hindered learning opportunities, making education more accessible and cost-effective (Srikrishna, 2022).

In this context, the concept of inbound marketing leads assumes a pivotal role. Inbound marketing leads refer to potential customers who have demonstrated their interest in a company's products or services, typically by taking actions like filling up webforms on websites, subscribing to news letters or attending a webinar. Identifying and nurturing these leads is of paramount importance for EdTech companies to successfully convert them into paying customers. The growing interest in EdTech and its offerings has introduced a new challenge: predicting which potential leads are most likely to transform into paying customers. This challenge underscores the need for predictive tools and mechanisms that can strategically target promising leads. To address this challenge, our project will leverage advanced data analytics and machine learning to build a predictive model capable of effectively estimating the probability of lead conversion. This project seeks to develop an advanced data analytics solution with the goal of predicting the likelihood of lead conversion within an EdTech company.

**2. Background**

**Company Overview**

In this project, we will take on the role of an EdTech company known as EdTech Inc. EdTech Inc. As previously mentioned, due to the rapidly expanding customer base there is an urgent need to optimize the productivity of our sales team. The end goal is to quickly convert leads into paying customers in order to sustain our growth trajectory and maintain a competitive edge.

**Business Problem**

The primary challenge confronting our business operations revolves around the accurate identification of leads with the highest potential to convert into paying customers. Currently, our sales teams face difficulties in prioritizing leads efficiently and accurately targeting them, resulting in suboptimal resource allocation. This challenge stems from the absence of a structured predictive tool or mechanism that facilitates the prioritization of sales efforts directed towards leads. This issue has the potential to adversely affect our conversion rates and significantly impact marketing initiatives and resource allocation decisions. Consequently, to address this challenge effectively, it is imperative that we adopt a data-driven and analytical approach.

Research conducted by Wu et al. (2023) underscores the significance of implementing a lead scoring model supported by predictive machine learning algorithms. However, they also noted that conventional lead scoring models tend to be overly simplistic, as many companies rely solely on basic metrics like demographics and firmographics to rank leads. Therefore, in this project, our intent is to expand the scope of our data collection to capture the unique nuances and preferences of each individual lead.

**Motivation**

Previous attempts to predict lead conversions relied on logistic regression models, assigning scores within a 0-100 range (Ashish, 2019). While this approach can provide valuable insights to our sales personnel, enabling them to prioritize their efforts and boost productivity and efficiency, it has its limitations. Specifically, it may not have been able to pinpoint which specific groups of prospects exhibited a higher likelihood of making purchases. This information is crucial for optimizing the allocation of marketing resources and minimizing waste. With these considerations in mind, we aim to build upon prior efforts by creating a model that can (1) empower EdTech Inc. to significantly enhance its marketing strategies by identifying key trends and patterns and (2) provide actionable insights that can enhance the efficiency and productivity of our sales team. Ultimately, this endeavor should ideally result in increased revenue generation and a higher return on investment for our marketing expenditures.

1. **Methodology**

For our methods, we have devised three distinct approaches each serving a specific purpose and contributing to our overall goal of enhancing lead conversion rates.

**Exploratory Data Analysis (EDA)**

Our first approach revolves around the utilization of Exploratory Data Analysis (EDA). EDA involves employing data visualization techniques to uncover underlying patterns, trends, and potential relationships within the dataset. This initial step equips our sales and marketing team with invaluable insights, allowing them to formulate actionable strategies aimed at improving lead conversion rates. By uncovering hidden data patterns, we can empower our teams to make informed decisions and optimize their efforts effectively.

**Random Forest**

Our second method involves the development of a Random Forest model. The Random Forest algorithm is an ensemble learning technique which combines various decision trees. We have chosen Random Forest over standalone decision trees for its robustness against overfitting issues and its ability to harness the collective predictions of individual trees, thereby enhancing predictive accuracy. In addition, in contrast to previous attempts using logistic regression for conversion rate prediction, our selection of the Random Forest model aims to identify features with the highest relative importance. This knowledge hopefully helps to uncover the factors influencing conversion rates, and thus providing our sales and marketing teams with actionable insights to improve their conversion strategies.

**Tableau Dashboard**

In the final phase, we plan to transform the outputs of our models into an interactive Tableau dashboard. This user-friendly interface will serve as an important tool for our sales team, offering them swift and actionable insights. Through the Tableau dashboard, our sales team can efficiently identify leads with a high potential for conversion. This functionality enhances the allocation of time and resources, making our sales and marketing efforts more effective and efficient.

1. **Data**

**Data Source and Acquisition**

In this section, we outline our data acquisition strategy for the project. Our data source is Kaggle. To obtain the dataset, we will utilize Kaggle's API and Python programming language. First, we'll install the Kaggle Python package by running the following command: pip install kaggle. Next, we will use the code block presented below to download the dataset into our local directory.

!kaggle datasets download -d ashydv/leads-dataset

This approach provides several advantages. It allows us to systematically organize our acquired data in a structured manner, ensuring that we can easily access the dataset for future updates. This flexibility also enables us to always keep our data current and relevant. The results of these steps are visualized in Figure 1. This step will set the stage for our comprehensive data exploration, preprocessing, and modeling.

****

Figure 1

**Data Description**

The dataset comprises a total of 37 columns and 9240 rows. It contains lead information sourced from various marketing channels, including digital advertising and web search, among others. Additionally, it includes a wide range of variables, such as engagement history, reasons for inquiry, employment status, and more. Within this dataset, the primary target variable is "convert," where "1" indicates a converted lead, and "0" represents a non-converted lead. Figure 2 provides a comprehensive list of all variables in the dataset. It is also important to note that this lead dataset aligns with other lead datasets from the author's professional experience. However, it lacks certain personal information, such as names, email addresses, job titles, and company details. We presume this omission is deliberate to safeguard personal identifiable information, considering the dataset's availability on a public platform. Consequently, we conclude that while it may not encompass all the details found in a typical lead dataset, it is sufficiently rich for the objectives of this project.

A screenshot of a computer

Description automatically generated

Figure 2

**Data Issues**

**Categorical Values**

As highlighted in Figure 2, we observe a high number of columns containing text and categorical values. It is worth noting that machine learning algorithms perform optimally with numeric values rather than text or categorical ones (Géron, 2019). As a result, it is imperative that transformation of these categorical values into numerical representations is needed to prepare the dataset for machine learning models effectively.

**Incomplete Data**

Again, referring to Figure 2, it is also observed that many attributes contain missing values. Machine learning models with missing data often exhibit reduced statistical strength, increased bias, and may lead to suboptimal business decision-making (Kang, 2013). Notably, "Lead quality" stands out as the attribute with the lowest completeness rate, at only 48%. While one might expect an attribute measuring lead quality to be meticulously recorded, our investigation revealed that this variable actually relies on subjective input from employees based on their intuition and available lead data. Given its low completeness rate, we may consider dropping this particular column altogether. However, a more nuanced approach will be necessary for addressing other columns with missing data. Excessive column removal may result in significant data loss and introduce potential bias (Roth, 1994). As such, we will adopt a targeted strategy that involves dropping entire columns when appropriate and imputing values for others.

**Presences of Outliers**

The other issue we identified is the presence of outliers. Referring to Figure 3 the columns “TotalVists” and “Page Views Per Visit” displays outlier values. The maximum values for “TotalVisits and “Page Views Per Visit” are 251 and 55 while their median is 3 and 2 respectively. The wide disparity between their median and maximum values indicates the presence of outliers in these two columns.

A screenshot of a computer

Description automatically generated

Figure 3

A boxplot is constructed to visualize and confirm the outliers. The boxplot in Figure 4 clearly illustrates the existence of outliers. These outliers will need to be addressed through data wrangling. Outliers may skew machine learning models and produce wrong correlation results (Gudivada et al., 2017). This can lead to incorrect business strategies and lead to poor business performance, which is something undesirable.

A graph of a graph of a number of numbers

Description automatically generated with medium confidence

Figure 4

1. **Data Wrangling**

In the prior section we identified three key data issues that would need to be addressed. They are 1) missing values, 2) presences of outliers and 3) categorical attributes. In this section, we will describe the steps taken to remedy these issues and provide a thorough rationale for our chosen approaches.

**Missing Values**

To address the concern of missing values, we employed a systematic approach. First and foremost, we made the decision to remove the "Lead Quality" column from our dataset. There are two reason for the decision. Firstly, it contains a substantial amount of missing data, and secondly due to its subjective nature, it makes accurate imputation challenging. Also, with more than 50% of the data missing in this column, we deemed its removal as a reasonable and practical choice.

For the remaining columns with missing data, which are mostly comprised of categorical variables, we adopted the SimpleImputer class from the scikit-learn library. The SimpleImputer class allows us to replace missing data with a specified constant, such as the median, mean, or the most frequent values. Our strategy for handling these categorical columns involved replacing missing values with the most frequent category. This approach is chosen so as to preserve the integrity of the data, ensuring that we do not introduce any bias into the dataset while maintaining completeness for the categorical attributes.

However, our imputation strategy for numerical values is more nuanced, as we take into consideration the presence of outliers in specific columns. In our prior exploratory analysis, we identified outliers in the "TotalVisits" and "Page Views Per Visit" columns. Given that outliers can exert a significant influence on the mean value and potentially skew our results, we decided to use the median imputation strategy for these particular columns. The median, being a robust measure of central tendency, is less affected by extreme data points and, therefore, provides a more reliable representation of the typical value in these cases. For the remaining numerical columns, where significant outliers were not detected, we chose to use the mean value for imputing missing data. The data wrangling process for handling missing values is carried out in Figure 5

A screenshot of a computer

Description automatically generated

Figure 5

**Outliers**

In part 3 we recognize the significance of addressing outliers in our dataset. Our primary approach involves the removal of rows containing outliers. Our primary approach involves the identification and removal of outlier data points. We have considered alternative methods such as replacing outliers with median or mean values, however, according to Oluleye (2023) such approaches can introduce unwanted bias into the dataset, potentially leading to incorrect conclusions and outputs.

To systematically identify and remove outliers from the "TotalVisits" and "Page Views Per Visit" variables, we employ the Interquartile Range (IQR) method. This approach entails calculating the first and third quartiles and using these values to compute the IQR. Any data points that fall beyond the upper or lower bounds of the IQR are considered outliers and are subsequently removed from the dataset. This process results in the elimination of 360 rows, representing approximately 4% of the entire dataset. Figure 6 illustrates the above mention process.

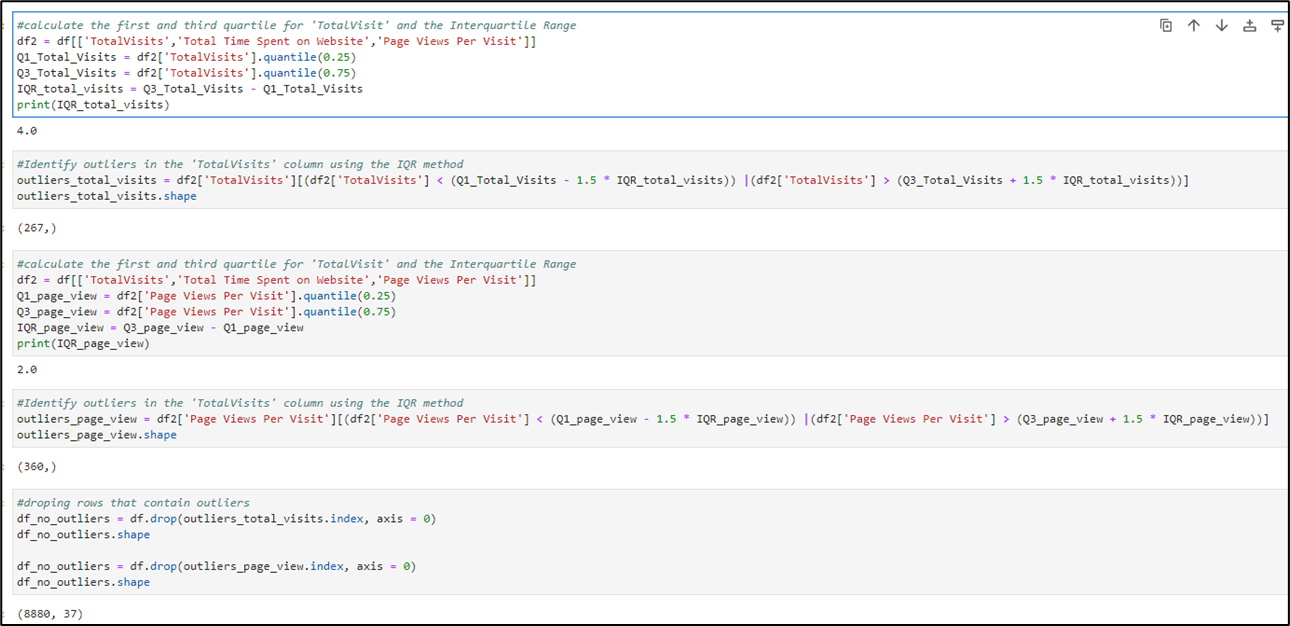


Figure 6

**Categorical Attributes**

The final step in our data wrangling process focuses on converting categorical data into numeric values. This transformation is essential to prepare our data for machine learning, as many machine learning algorithms require numeric inputs. To accomplish this conversion, we leverage OneHotEncoder and ColumnTransformer from scikit-learn library.

For the numerical columns, we specify 'passthrough' in the ColumnTransformer. This ensures that no alterations are made to these columns, preserving their original numeric values. For the categorical data, we employ the OneHotEncoder. This technique converts categorical values into a binary matrix, where each category becomes a binary column, and each data point is represented by a 1 in the corresponding category column and 0s elsewhere. This process effectively transforms text-based categorical data into a numeric format suitable for machine learning algorithms. Figure 7 illustrates block for the above-mentioned process.

**A screen shot of a computer

Description automatically generated**

Figure 7

1. **Exploratory Data Analysis**

In this section we will embark on the initial phases of our proposed methodology which involves exploring the data to uncover any trends, patterns or anomalies that can provide insight into how to improve lead conversion in EdTech Inc. For our exploration we employed data visualization, data visualization makes it easier and quicker to notice patterns and trends and they do a far better job at attracting attention to key issues compared to a table of numbers (Wexler, 2021).

**Understanding Class Imbalance**

In the initial phase of our exploratory data analysis, we focus on assessing the dataset for any potential imbalance. Our findings reveal a distribution of approximately 60% to 40%, which, while not a perfect 50:50 equilibrium, is remarkably close. Hence it is determined that no specialized data handling procedures will be necessary.

**A screenshot of a graph

Description automatically generated**

Figure 8

**Insights**

In the following section through data visualization, we will delve into some pivotal data insights. We start with the "Last Activity" variable, our bar chart in Figure 9 highlights some interesting patterns. Notably, "Email Opened" emerges as the biggest activity, displaying a substantial number of leads, with conversion rates remaining relatively decent when benchmarked against other activities. "SMS Sent," on the other hand, ranks as the second most prevalent prospect-engagement activity but exhibits the highest conversion rate, implying its potential as a more effective approach for lead outreach. Given the ubiquity of mobile phones, SMS communication provides a convenient means for people to access information on the go, potentially enhancing its efficacy. Conversely, "Olark Chat" registers a notably low conversion rate, raising questions about its suitability for engaging with prospects.

A graph of a chart

Description automatically generated with medium confidence

Figure 9

Next, we move on to the next variable "Lead Source". Our findings from Figure 10 reveal that **"**Direct Traffic**"**, representing prospects who directly visit our webpage, exhibits the highest lead count. However, it is noteworthy that this source registers a relatively modest conversion rate of 37%. This observation prompts a need for a deeper analysis of our website's performance, as it raises questions about why conversion rates aren't higher, considering that prospects arriving directly should possess preexisting interest. Following direct traffic, the second-highest lead source is **"**Google**"**, demonstrating a commendable conversion rate of nearly 50%. Additionally, we observe that referrals exhibit exceptionally high conversion rates, but this outcome is unsurprising, as referrals usually stem from individuals or entities who have expressed prior interest in our services. Another noteworthy high-conversion source is traffic from the **"**WelingKar Website**"**, an educational institution's platform. The strong conversion rate from this source aligns with the assumption that individuals inclined towards higher education are more likely to explore additional courses that complement their ongoing studies at such institutions. These findings underscore the importance of fostering closer collaboration between our marketing and sales teams and similar educational institutions to leverage this high-conversion potential.

A screen shot of a graph

Description automatically generated

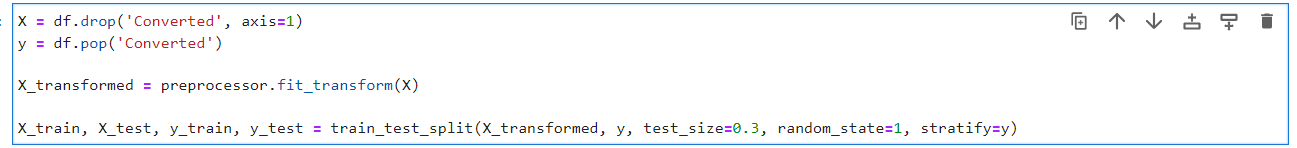
Figure 10

Lastly in our exploration of the "Current Occupation" variable we observed a notable pattern in Figure 10. The dataset primarily comprises unemployed individuals, which is unsurprising, given the motivation of unemployed individuals to enhance their skill sets in pursuit of better employment prospects. And the conversion rate is relatively decent, at approximately 50:50. However, an interesting finding emerges when looking at the data for employed professionals. Their conversion rate stands impressively at around 91%. This aligns with our initial contextualization in the introduction, where we highlighted external market forces driving the demand for new skills to maintain relevance. This finding holds strategic significance, suggesting the marketing team should consider allocating additional resources to attract and engage working professionals as a high-potential customer segment.

1. **Random Forest**

As mentioned in section 3 our second method is the creation of a Random Forest Model. By creating a Random Forest model, we aim to address the most critical challenges for EdTech Inc which is to effectively predict and optimize lead conversions. And to accomplish this we will be leverage Python programming language and the scikit-learn library.

We initiate the model building process by splitting the data into feature and target sets, further dividing them into training and testing subsets. For this project, 70% of the data will be used for training, with the remaining 30% earmarked for testing. Additionally, we employ the OneHotEncoder to convert categorical attributes into numerical values for the Random Forest model.

Figure 11

Having completed the data split, we proceed to train the model using the RandomForestClassifier function from scikit-learn and generate predictions using the test dataset. Figure 12 provides a code block illustrating this process.

A white background with black text

Description automatically generatedFigure 12

Once the model is trained and predictions are made on the testing data, we evaluate its performance using key indicators: accuracy, recall, and precision. Figure 13 displays the results.

A screenshot of a computer code

Description automatically generated

Figure 13

For the Initial results we observe an accuracy score of 91.7%, a recall score of 85.5%, and a precision score of 92.5%. These results appear promising when compared to previous attempts utilizing logistic regression by Ashish (2019), which yielded an accuracy score of 90.7%. However, it's important to note that all three metrics are scored at 100% against training data, indicating potential issues with overfitting. This suggests that the model might not perform well with unseen or new data. Thus, these initial results are suboptimal, primarily due to the presence of overfitting, and thus require hyperparameter tuning.

To address overfitting and identify the best estimator, we employ RandomizedSearchCV. This approach efficiently explores various hyperparameter combinations using cross-validation and selects the best estimator. RandomizedSearchCV is used instead of an exhaustive grid search over the exhaustive GridSearchCV due to its computational efficiency and faster results.

Once the optimal hyperparameters are determined, the model is retrained using these settings, and predictions are made on the test data. The precision score is used as the primary metric for selecting the best estimator. Precision score focuses on the correctness of positive predictions and is particularly relevant when false positives are costly. This is inline with our business needs where we want to prioritize finite efforts and resources on leads that will convert into paying customers and subsequently reduce efforts and resources on false positive leads, which is a costly situation.

Figure 14 shows the code block used for the hyperparameter tunning and corresponding results. The results of the tuned Random Forest Classifier indicate that it is performing well on both the training and test sets, with high accuracy, recall, and precision. The precision on the training set (93%) and test set (90%) is also quite high. As mentioned, having a high precision is desirable because it indicates that when the model predicts a lead will convert, it is likely to be correct. Ultimately, this precision-oriented model development supports the strategic allocation of resources towards the most promising leads, thereby optimizing the conversion process.

A screenshot of a computer program

Description automatically generated

Figure 14

1. **Tableau Dashboard**

After developing our predictive model, the next step involves the creation of an interactive and user-friendly Tableau dashboard. This dashboard is designed to empower sales professionals to efficiently identify leads that our model predicts as likely to convert – we refer to these leads as "Hot" leads. We achieve this by utilizing our model's predictions on the entire dataset and subsequently utilizing this output as the foundational data for constructing the dashboard. It's important to note that due to a lack of new data from the Edtech company, we are reusing the original dataset to simulate the process of integrating model outputs into a dashboard. Additionally, we employ the "predict\_proba" function, which provides probability estimates for each class. Specifically, it calculates the fraction of decision trees within the random forest that predicts each class.

A screen shot of a computer code

Description automatically generated

Figure 15

Figure 15 illustrates this process. To begin, we utilize our tuned model, "rf\_estimator\_tuned," to predict the entire dataset, excluding the "converted" column. After making these predictions, we append the predicted values and probability estimates back into the original data frame, ultimately exporting it in CSV format. This CSV file becomes the foundation for constructing the dashboard.

A screenshot of a dashboard

Description automatically generated

Figure 16 ([Link](https://public.tableau.com/app/profile/james.hou5128/viz/LeadPrioritization/LeadPrioritizationDashboard))

Figure 16 shows the final version of the Tableau dashboard, which is organized into three distinct sections. The first section (Lead Overview) offers an overview of our leads using a donut chart. This chart provides a visual representation of the proportion of "Hot" leads relative to the total number of leads. It is a straightforward and intuitive way for salespeople to quickly identify the quantity of "Hot" leads that require their immediate attention. This section presents both the total number of leads and the number predicted to be converted into paying customers. This provides a quick snapshot of the potential conversion rate.

The second section (Lead Status By XXX) of the dashboard is dedicated to ranking leads based on various attributes that can be filtered. Users can access a selector, which permits them to choose which variable to use for slicing and dicing the view. Figure 17 demonstrates how this selector can be used to view leads by their source. This feature empowers salespeople to dissect and explore the data, facilitating data-driven decision-making. For example, they can explore which leads are most promising based on specific criteria from the drop down.

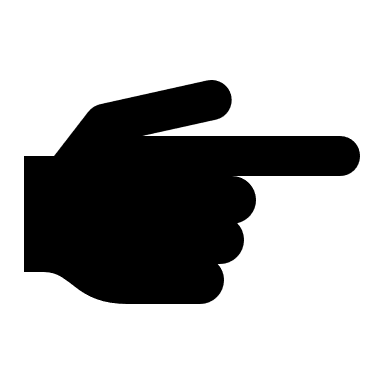
A graph of numbers and a bar

Description automatically generated with medium confidence

Figure 17

The third and final section (Lead Detail) is designed to offer detailed lead information. It includes probability estimates, presented as percentages, indicating the confidence level of lead conversion. This additional data point enables sellers to prioritize their outreach efforts more effectively, relying on data-driven insights. It's worth mentioning that, in a real-world scenario, this section would include personally identifiable information, such as contact details and lead names.

The entire dashboard is interactive, allowing users to select and filter data points. Figure 18 provides an example of selecting "Hot" leads from a specific location, such as Mumbai.

A screenshot of a computer

Description automatically generated

Figure 18

In summary, this dashboard is intended as a valuable tool for sales professionals, helping them visualize and prioritize leads. It enables them to focus their efforts and resources on "Hot" leads and further refine their selections by manipulating various attributes. This approach allows them to combine their experience with data-driven insights, enhancing the probability of conversion and increasing their overall productivity and efficiency.

As an example, Figure 19 illustrates how an experienced salesperson with expertise in selling finance management courses can leverage this dashboard. Among the pool of "Hot" leads, they can select the specialization criteria, instantly obtaining a list of relevant leads in the "Lead Details" section. This blending of experience and data-driven insight serves to boost the likelihood of successful conversions while enhancing productivity and efficiency.

A screenshot of a computer

Description automatically generated

Figure 19

1. **Findings and Recommendations**

After completing our data analytics focused on using machine learning to predict lead conversion in the EdTech industry, this section presents the key findings and recommendations based on insights derived from exploratory data analysis, the Random Forest model, and the Tableau dashboard.

**Lead Activity**

Upon examining the impact of lead activity on conversion rates through EDA, a significant observation emerged. Olark Chat, despite being the third most frequently used outreach option, displayed a notably low conversion rate. In contrast, SMS Sent boasted a high conversion rate of 63% while accounting for 30% of the entire lead population. These findings suggest that prioritizing SMS as the primary means of reaching out to prospective leads may enhance their engagement and conversion rates. This recommendation stems from the ubiquity and accessibility of mobile phones, making SMS a more effective communication method.

**Current Occupation**

Analysis of the dataset revealed that the majority of leads, approximately 85%, were unemployed. Despite this high total lead volume, their conversion rate was only 44%. Several factors might contribute to this, including financial constraints, competing priorities, and immediate needs. To gain deeper insights, it is recommended to conduct surveys targeting unemployed leads who declined signing up. Such surveys can pinpoint the primary obstacles, enabling the design of tailored strategies, such as subsidized fees, free trials, or internship opportunities, to entice this group.

In contrast, employed professionals exhibited an impressive conversion rate of approximately 91%. This is in line with the earlier context on the boom in digitalization and the redefinition of traditional job roles and scope. This underscores the importance of targeting working professionals as a high-potential customer segment. As such, it is recommended to shift or increase marketing campaigns towards this demographic.

**Random Forest Model**

The initial model displayed strong accuracy, recall, and precision scores but exhibited signs of overfitting with a 100% score on the training data. This indicates potential shortcomings in real-world applications. Hyperparameter tuning resulted in a final model with strong performance, signifying its effectiveness in predicting lead conversion. Precision score, which gauges the correctness of positive predictions, was prioritized to address our business problem.

With this model we also look at the top 20 feature importance in Figure 20. The top 20 feature importance, as shown in Figure 19, confirmed the relevance of features like Last Activity (SMS Sent) and Current Occupation (working professional) in influencing the model's predictions. Notably, Total Time Spent on Website emerged as the most crucial feature. This was an unsurprising feature since it is logical that when an individual has any interest in anything they would likely spend more time on it. Thus, resulting in a positive correlation between the amount of time spent on the website to the probability of conversion. Conversely, other similar features such as TotalVisits and Page Views Per Visit are ranked at the bottom as less influential factors. This suggests that the number of visits count has little correlation on the conversion rate. This is an interesting finding as an intuitive assumption would be that increased returns visits and number of visits should equate to total time spent on the website as well. This could suggest poor website interface or user experience where the website layout might be confusing or difficult to navigate. This also corroborates the findings in Figure 9 while direct traffic yields low conversion rates.

A graph with blue and white bars

Description automatically generated

Figure 20

While the above is the final model we produced, we also duplicate the flow of our analytic process with different tweaks and adjustments to seek out additional insights.

**Retention of Outliers**

Random Forest is a machine learning model that is particularly robust to outliers (Breiman, 2001). One of the adjustments we made is to retain all outliers and to observe any changes to the model and feature importance. Figure 21 shows the result after tunning the model with RandomizedSearchCV and Figure 22 shows the feature importance. In terms of precision score, the model with outliers retained shows a slightly better precision score on testing data of 90.8% compared to 90.3%. The difference appears to be immaterial. In Figure 20, we see that the outliers have affected Page Views Per Visit and TotalVisits importance as they moved up in terms of importance however, they are still ranked in the bottom 3. In summary, this new model with outliers retained did not provide much additional insight as the result did not swing too much. However, as Random Forest is proven to be robust against outliers, for future iterations, outliers will be retained.

**A number of numbers on a white background

Description automatically generated**

Figure 21

**A graph of a bar graph

Description automatically generated with medium confidence**

Figure 22

**Retention of Lead Quality**

As mentioned in chapter 4 and 5, Lead Quality was dropped entirely due to over 50% of missing values due to it being a subjective input. In this investigation we will retain Lead quality column duplicate the same process as above. The results are in Figure 23 and Figure 24. Similar to the findings above the final results do not differ much from the original tuned model with precision score of 89.9% vs the original 90.3% with a marginal decline of 0.4%. Furthermore, the feature importance rankings did not place Lead Quality among the top 20 influential factors. Hence, the initial decision to remove the Lead Quality column appears justified.

A number and numbers on a white background

Description automatically generated

Figure 23

A graph with blue and white bars

Description automatically generated

Figure 24

In conclusion, the findings and recommendations highlight the significance of prioritizing SMS as a communication method, targeting employed professionals, and the importance of Total Time Spent on Website in predicting lead conversion. The exploration of outliers and the retention of the Lead Quality column did not yield substantial benefits, affirming the initial analytical approach. However, it is also important to note some drawbacks regarding this project. The data source itself poses a challenge, due to anonymity of the origins of the dataset, several factors of the data lack context on what they could possibly mean or how they may be derived. Examples include “Lead Profile” and “Asymmetrique Scores”. How and where these data points are derived are not explained properly. Hence this creates a barrier to how some of the final findings may be interpreted. While this drawback is recognized as a limitation to a more holistic project, we find that the insights and recommendation presented as well as the development of the Random Forest model can still sufficiently guide strategic decisions and improve the efficiency of lead conversion efforts in the EdTech industry.

1. **Conclusion**

In the age of digitalization, intensified by external drivers like COVID-19 and rapid technological advancements, the EdTech industry has seen unprecedented growth. This surge has resulted in the pressing need for companies like EdTech Inc. to enhance their lead conversion strategies. With the objective of improving lead conversion prediction, this project took a deep dive into data analytics and machine learning.

Our Exploratory Data Analysis (EDA) highlighted significant insights, such as the effectiveness of SMS as a communication tool and the higher conversion potential of employed professionals.

The Random Forest model employed was adept at identifying the nuances of leads that could be potential conversions. The model, after fine-tuning, showed a decent precision score of 90.3% that can prove highly beneficial for business operations. Particularly, the model highlighted the importance of metrics like "Total Time Spent on Website" and thus suggesting the importance of a robust user experience for lead conversion.

However, while the machine learning model offered profound insights, the real power lies in its practical application. The Tableau dashboard developed for this purpose is the bridge between insights and actionable strategies. It empowers sales professionals to effectively prioritize high-conversion potential leads, ensuring optimal resource allocation.

In summary, this project has charted a clear path forward. As the EdTech landscape continues to evolve, such data-driven insights and strategies will be paramount in ensuring companies remain competitive, efficient, and responsive to their potential customers' needs

**References:**

1. Ashish. (2019). *Leads dataset.* Kaggle. <https://www.kaggle.com/datasets/ashydv/leads-dataset/code>
2. Breiman L. *Random forests*. Machine learning. 2001 Oct 1; 45(1):5–32
3. Géron, A. (2019). *Hands-on machine learning with scikit-learn, Keras, and tensorflow concepts, tools, and techniques to build Intelligent Systems*. O’Reilly.
4. Gudivada, V., Apon, A., and Ding, J. (2017). *Data quality considerations for big data and machine learning: going beyond data cleaning and transformations*. Int. J. Adv. Softw. 10, 1–20
5. Kang, H. (2013). The prevention and handling of the missing data. *Korean Journal of Anesthesiology*, *64*(5), 402. https://doi.org/10.4097/kjae.2013.64.5.402
6. Oluleye, A. (2023). *Exploratory data analysis with Python cookbook : over 50 recipes to analyze, visualize, and extract insights from structured and unstructured data (1st ed.).* Packt Publishing.
7. Roth, P. (1994). *Missing data: A conceptual review for applied psychologists*. Personnel Psy-chology,47, 537-560
8. Srikrishna, A. (2022). *Higher Edtech: The catalyst to upskilling in India.* Entrepreneur. https://www.entrepreneur.com/en-in/growth-strategies/higher-edtech-the-catalyst-to-upskilling-in-india/440106
9. Wexler, S. (2021). *The big picture : how to use data visualization to make better decisions-faster*. McGraw-Hill.
10. Wu, M., Andreev, P., &amp; Benyoucef, M. (2023). *The state of lead scoring models and their impact on sales performance*. Information Technology and Management*.* <https://doi.org/10.1007/s10799-023-00388-w>

**Appendix**

Python Code:

#import libraries

import pandas as pd

import os

import zipfile

import matplotlib.pyplot as plt

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.impute import SimpleImputer

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

from sklearn import metrics

from sklearn.metrics import (

    accuracy\_score,

    recall\_score,

    precision\_score,

    make\_scorer,

    confusion\_matrix,

    ConfusionMatrixDisplay,

    confusion\_matrix,

    r2\_score)

from sklearn.model\_selection import GridSearchCV

from sklearn.model\_selection import RandomizedSearchCV

from sklearn.calibration import CalibratedClassifierCV

from sklearn.preprocessing import StandardScaler

import numpy as np

#Downloading dataset from Kaggle

!kaggle datasets download -d ashydv/leads-dataset

#Unzipping the file

zipfile\_path = os.getcwd()+'\leads-dataset.zip'

with zipfile.ZipFile(zipfile\_path, 'r') as zip\_ref:

    zip\_ref.extractall(os.getcwd())

#reading the dataset into pandas

df = pd.read\_csv(os.getcwd()+'\Leads.csv')

#df = df.dropna(subset=['Lead Quality'])

df.info()

df.describe()

#ploting out boxplot

df2 = df[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']]

fig, axes = plt.subplots(nrows=1, ncols=len(df2.columns), figsize=(12, 6))

for idx, column in enumerate(df2.columns):

    df2.boxplot(column=column, ax=axes[idx])

    axes[idx].set\_title(f'Box Plot of {column}')

    axes[idx].set\_ylabel('Values')

    axes[idx].set\_xlabel(column)

plt.tight\_layout()

plt.show()

Impute

# drop lead quality

df = df.drop('Lead Quality', axis=1)

#assign different imputation strategy

numerical\_imputer = SimpleImputer(strategy='median')

categorical\_imputer = SimpleImputer(strategy='most\_frequent')

numerical\_imputer2 = SimpleImputer(strategy='mean')

#assign variables into 3 different buckets for imputation

numerical\_cols = ['TotalVisits','Page Views Per Visit']

numerical\_col2 = ['Lead Number','Total Time Spent on Website','Asymmetrique Activity Score','Asymmetrique Profile Score']

categorical\_cols = ['Prospect ID', 'Lead Origin', 'Lead Source', 'Do Not Email', 'Do Not Call','Last Activity','Country','Specialization','How did you hear about X Education','What is your current occupation','What matters most to you in choosing a course','Search','Magazine','Newspaper Article','X Education Forums','Newspaper','Digital Advertisement','Through Recommendations','Receive More Updates About Our Courses','Receive More Updates About Our Courses','Tags','Update me on Supply Chain Content','Get updates on DM Content','Lead Profile','City','Asymmetrique Activity Index','Asymmetrique Profile Index','I agree to pay the amount through cheque','A free copy of Mastering The Interview','Last Notable Activity']

df[numerical\_cols] = numerical\_imputer.fit\_transform(df[numerical\_cols])

df[numerical\_col2] = numerical\_imputer2.fit\_transform(df[numerical\_col2])

df[categorical\_cols] = categorical\_imputer.fit\_transform(df[categorical\_cols])

Removing Outliers

#calculate the first and third quartile for 'TotalVisit' and the Interquartile Range

df2 = df[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']]

Q1\_Total\_Visits = df2['TotalVisits'].quantile(0.25)

Q3\_Total\_Visits = df2['TotalVisits'].quantile(0.75)

IQR\_total\_visits = Q3\_Total\_Visits - Q1\_Total\_Visits

print(IQR\_total\_visits)

#Identify outliers in the 'TotalVisits' column using the IQR method

outliers\_total\_visits = df2['TotalVisits'][(df2['TotalVisits'] < (Q1\_Total\_Visits - 1.5 \* IQR\_total\_visits)) |(df2['TotalVisits'] > (Q3\_Total\_Visits + 1.5 \* IQR\_total\_visits))]

outliers\_total\_visits.shape

#calculate the first and third quartile for 'TotalVisit' and the Interquartile Range

df2 = df[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']]

Q1\_page\_view = df2['Page Views Per Visit'].quantile(0.25)

Q3\_page\_view = df2['Page Views Per Visit'].quantile(0.75)

IQR\_page\_view = Q3\_page\_view - Q1\_page\_view

print(IQR\_page\_view)

#Identify outliers in the 'TotalVisits' column using the IQR method

outliers\_page\_view = df2['Page Views Per Visit'][(df2['Page Views Per Visit'] < (Q1\_page\_view - 1.5 \* IQR\_page\_view)) |(df2['Page Views Per Visit'] > (Q3\_page\_view + 1.5 \* IQR\_page\_view))]

outliers\_page\_view.shape

#droping rows that contain outliers

df\_no\_outliers = df.drop(outliers\_total\_visits.index, axis = 0)

df\_no\_outliers.shape

df\_no\_outliers = df.drop(outliers\_page\_view.index, axis = 0)

df = df\_no\_outliers

df.shape

Convert Categorical to Numbers

# Use ColumnTransformer with OneHotEncoder

categorical\_cols = ['Prospect ID', 'Lead Origin', 'Lead Source', 'Do Not Email', 'Do Not Call','Last Activity','Country','Specialization','How did you hear about X Education','What is your current occupation','What matters most to you in choosing a course','Search','Magazine','Newspaper Article','X Education Forums','Newspaper','Digital Advertisement','Through Recommendations','Receive More Updates About Our Courses','Receive More Updates About Our Courses','Tags','Update me on Supply Chain Content','Get updates on DM Content','Lead Profile','City','Asymmetrique Activity Index','Asymmetrique Profile Index','I agree to pay the amount through cheque','A free copy of Mastering The Interview','Last Notable Activity']

numerical\_cols = ['TotalVisits','Total Time Spent on Website','Lead Number','Page Views Per Visit','Asymmetrique Activity Score','Asymmetrique Profile Score']

preprocessor = ColumnTransformer(

    transformers=[

        ('num', 'passthrough', numerical\_cols),

        ('cat', OneHotEncoder(handle\_unknown='ignore'), categorical\_cols)

    ])

Building the model

def metrics\_output(model,flag=True):

    '''

    model : classifier to predict values of X

    '''

    # Generate predicted values

    y\_pred = model.predict(X\_test)

    # defining an empty list to store train and test results

    score\_list=[]

    #Predicting on train and tests

    pred\_train = model.predict(X\_train)

    pred\_test = model.predict(X\_test)

    #Accuracy of the model

    train\_acc = model.score(X\_train,y\_train)

    test\_acc = model.score(X\_test,y\_test)

    #Recall of the model

    train\_recall = metrics.recall\_score(y\_train,pred\_train)

    test\_recall = metrics.recall\_score(y\_test,pred\_test)

    #Precision of the model

    train\_precision = metrics.precision\_score(y\_train,pred\_train)

    test\_precision = metrics.precision\_score(y\_test,pred\_test)

    score\_list.extend((train\_acc,test\_acc,train\_recall,test\_recall,train\_precision,test\_precision))

    # If the flag is set to True then only the following print statements will be dispayed. The default value is set to True.

    if flag == True:

        print("Accuracy on training set : ",model.score(X\_train,y\_train))

        print("Accuracy on test set : ",model.score(X\_test,y\_test))

        print("Recall on training set : ",metrics.recall\_score(y\_train,pred\_train))

        print("Recall on test set : ",metrics.recall\_score(y\_test,pred\_test))

        print("Precision on training set : ",metrics.precision\_score(y\_train,pred\_train))

        print("Precision on test set : ",metrics.precision\_score(y\_test,pred\_test))

    return score\_list # returning the list with train and test scores

X = df.drop('Converted', axis=1)

y = df.pop('Converted')

X\_transformed = preprocessor.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_transformed, y, test\_size=0.3, random\_state=1, stratify=y)

# Build and train a Random Forest model

rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

rf\_model.fit(X\_train, y\_train)

y\_pred\_rf = rf\_model.predict(X\_test)

rf\_estimator\_score=metrics\_output(rf\_model)

Hyperparameter Tunning

# Choose the type of classifier.

rf\_estimator\_tuned = RandomForestClassifier(random\_state=1)

# Grid of parameters to choose from

parameters = {"n\_estimators": [150,200, 250],

    "min\_samples\_leaf": np.arange(5, 10),

    "max\_features": np.arange(0.2, 0.7, 0.1),

    "max\_samples": np.arange(0.3, 0.7, 0.1),

    "max\_depth" : [None, 10, 20, 30, 40, 50]

             }

# Choosing precision score as the metric for comparison

scorer = make\_scorer(precision\_score)

# Run the randomizedSearchCV

rand\_obj = RandomizedSearchCV(rf\_estimator\_tuned, parameters, scoring=scorer, cv=5)

rand\_obj = rand\_obj.fit(X\_train, y\_train)

# Choose the best classifier

rf\_estimator\_tuned = rand\_obj.best\_estimator\_

# Train the best classifier

rf\_estimator\_tuned.fit(X\_train, y\_train)

# Obtain predicted values from the test set

y\_pred\_rf\_t = rf\_estimator\_tuned.predict(X\_test)

rf\_estimator\_score=metrics\_output(rf\_estimator\_tuned)

Feature Importance

# Calculate feature importances

feature\_importances = rf\_estimator\_tuned.feature\_importances\_

# Get the names of the categorical columns after one-hot encoding

categorical\_encoder = preprocessor.named\_transformers\_['cat']

categorical\_feature\_names = categorical\_encoder.get\_feature\_names\_out(input\_features=categorical\_cols)

# Combine numerical and one-hot encoded categorical feature names

all\_feature\_names = numerical\_cols + list(categorical\_feature\_names)

# Create a DataFrame to store feature names and their importances

feature\_importance\_df = pd.DataFrame({'Feature': all\_feature\_names, 'Importance': feature\_importances})

# Sort the DataFrame by importance in descending order

feature\_importance\_df = feature\_importance\_df.sort\_values(by='Importance', ascending=False)

# Select the top 20 features by importance

top\_20\_features = feature\_importance\_df.head(20)

# Print the top 20 feature importances

print("Top 20 Feature Importances:")

# Plot the top 20 feature importances

plt.figure(figsize=(12, 8))

plt.barh(range(len(top\_20\_features)), top\_20\_features['Importance'], align='center')

plt.yticks(range(len(top\_20\_features)), top\_20\_features['Feature'])

plt.xlabel('Feature Importance')

plt.title('Top 20 Random Forest Feature Importance')

plt.gca().invert\_yaxis()

plt.show()

Output into CSV

predictions = rf\_estimator\_tuned.predict(X\_transformed)

probability = rf\_estimator\_tuned.predict\_proba(X\_transformed)

df['Predictions'] = predictions

df['Probability'] = probability[:,1]

df.to\_csv('random\_forest\_predictions.csv', index=False)