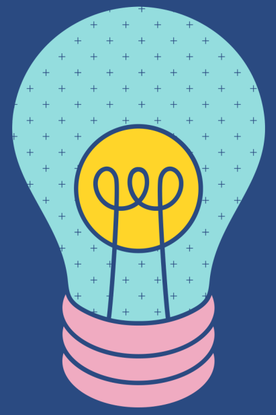
CLASSIFICATION & HYPOTHESIS TESTING

ExtraaLearn and The Education Technology Revolution

Presented by Jeffandy St.Hubert

August 5, 2024



Agenda

KEY TOPICS DISCUSSED IN THIS PRESENTATION

Business Problem Overview

Solution Approach

Data Overview

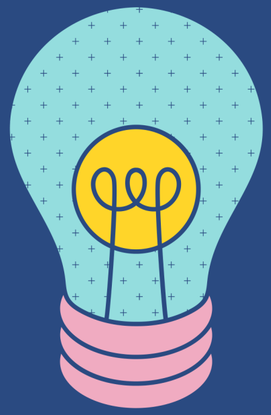
EDA Results - Univariate and Multivariate

Data Preprocessing

Model Performance Summary

Conclusion and Recommendations

Thank You

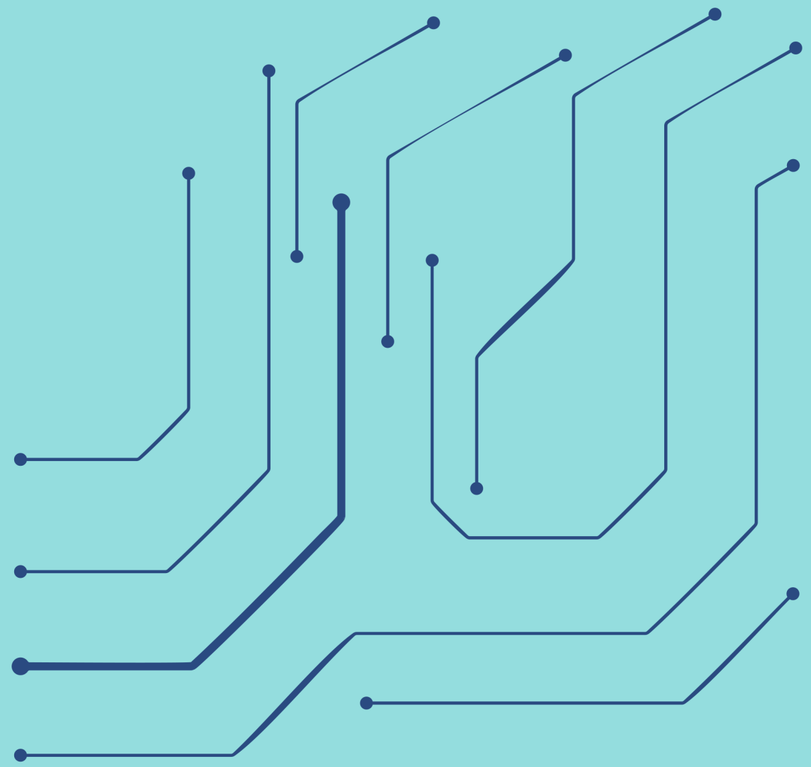


The Education landscape is changing and ExtraaLearn can lead.

Business Problem Overview

Technology is a powerful tool that has altered the teaching and learning experience worldwide. According to a forecast, the Online Education market will be worth $286.62 billion very soon and see an annual growth rate (CAGR) of 10.26% for the next few years.

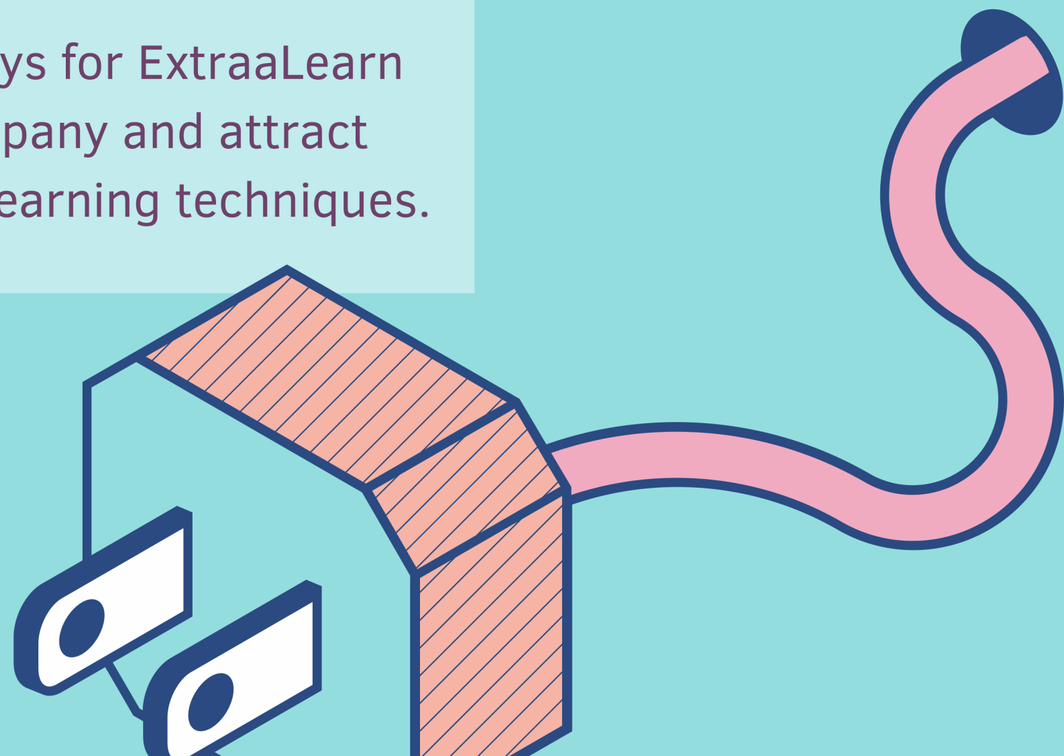
ExtraaLearn has the opportunity to lead this movement by converting the thousands of interested individuals (leads) into accomplished learners who are upskilling and reskilling.

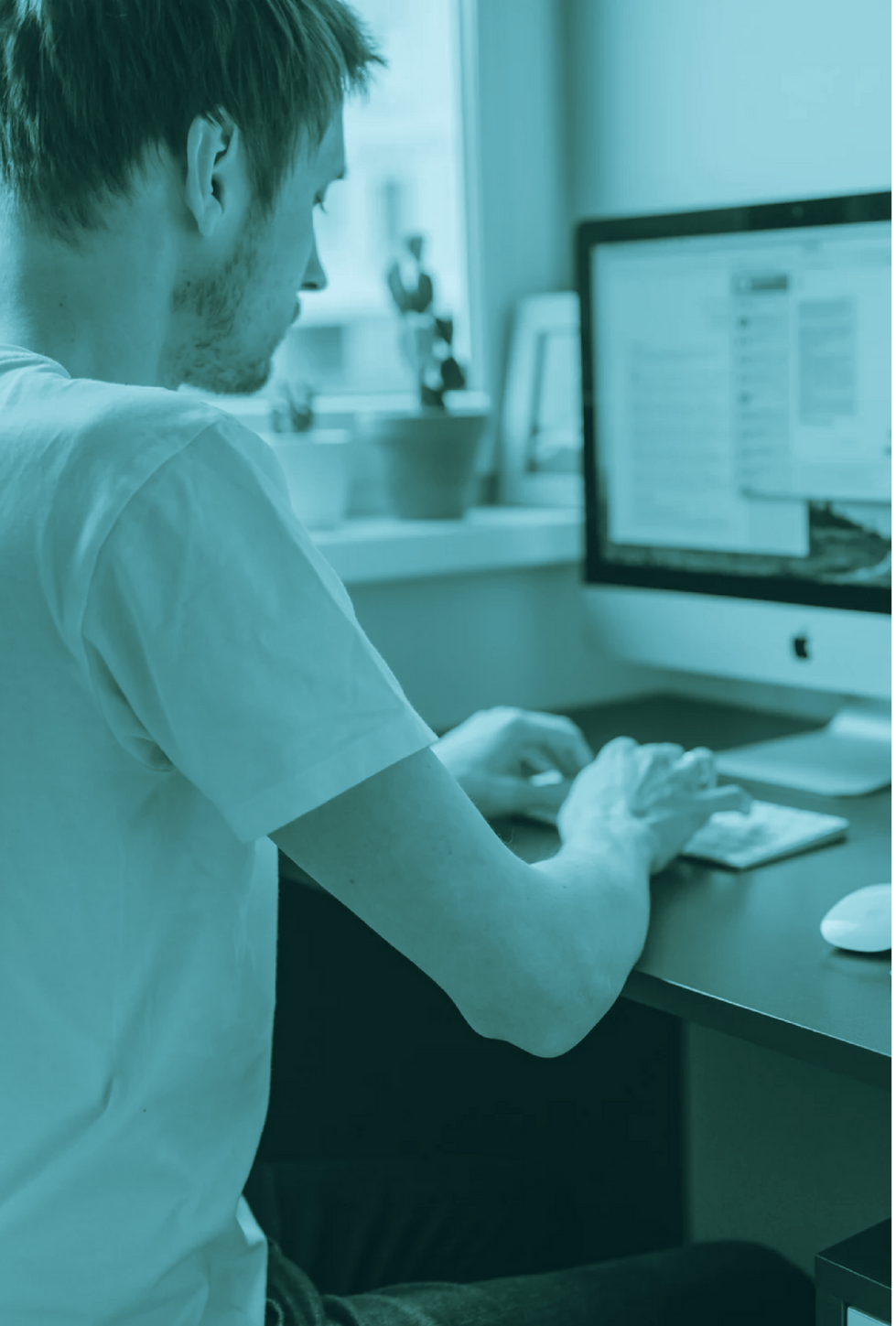


Solution Approach

The Education landscape is changing and ExtraaLearn can lead.

In this presentation, we will explore ways for ExtraaLearn to position itself as a cutting-edge company and attract new prospects using various machine learning techniques.





Data Overview

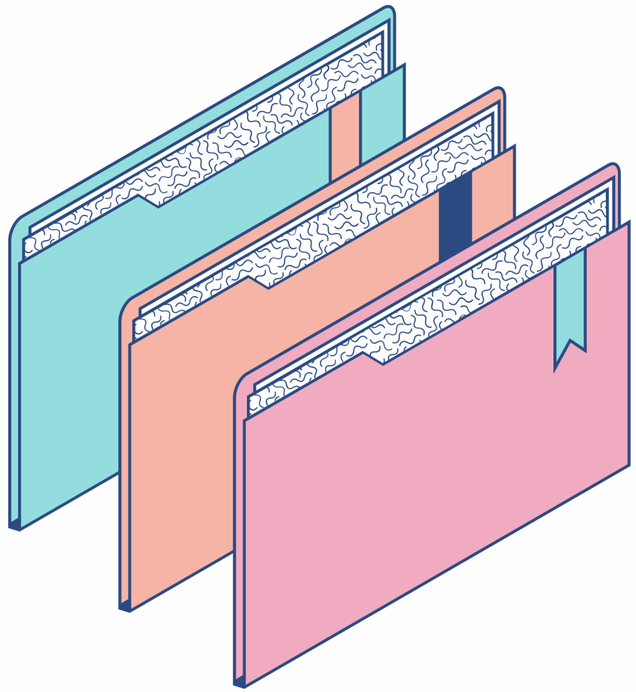
THE DATA CONTAINS DIFFERENT ATTRIBUTES OF PROSPECTIVE LEARNERS (LEADS) AND THEIR INTERACTION WITH EXTRAALEARN, INCLUDING:

WEBSITE_VISITS
TIME_SPENT_ON_WEBSITE
PAGE_VIEWS_PER_VISIT
CURRENT_OCCUPATION
FIRST_INTERACTION
PROFILE_COMPLETED
LAST_ACTIVITY
PRINT_MEDIA_TYPE1
PRINT_MEDIA_TYPE2
DIGITAL_MEDIA
EDUCATIONAL_CHANNELS
REFERRAL
STATUS

Data Overview Continued

THE SHAPE OF THE DATA SHOWS THAT WE HAVE 15 FEATURES AND 4,612 ENTRIES

```
✓ 0s   data.shape
       (4612, 14)
```





EDA Univariate Results

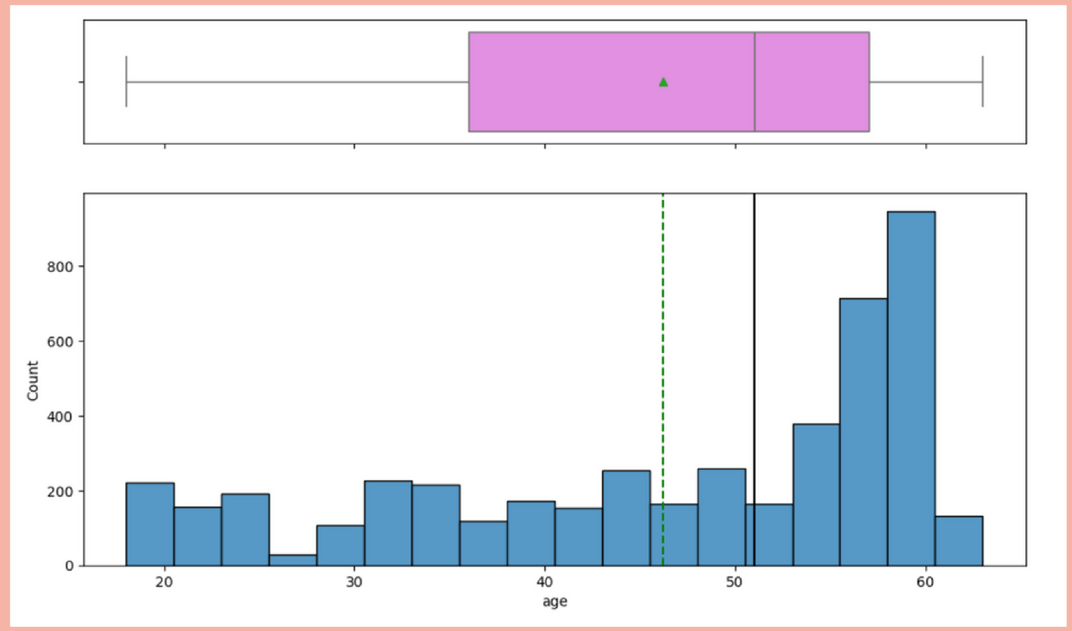
EXPLORATORY DATA ANALYSIS SHOWS THAT THE MEAN AGE IS 46, AND THE TIME SPENT HAS OUTLIERS, WHICH IS TO BE EXPECTED SINCE IT IS CALCULATED IN SECONDS.

```
[13] data.describe().T #here is the statistical summary of the data
```

	count	mean	std	min	25%	50%	75%	max
age	4612.00000	46.20121	13.16145	18.00000	36.00000	51.00000	57.00000	63.00000
website_visits	4612.00000	3.56678	2.82913	0.00000	2.00000	3.00000	5.00000	30.00000
time_spent_on_website	4612.00000	724.01127	743.82868	0.00000	148.75000	376.00000	1336.75000	2537.00000
page_views_per_visit	4612.00000	3.02613	1.96812	0.00000	2.07775	2.79200	3.75625	18.43400
status	4612.00000	0.29857	0.45768	0.00000	0.00000	0.00000	1.00000	1.00000

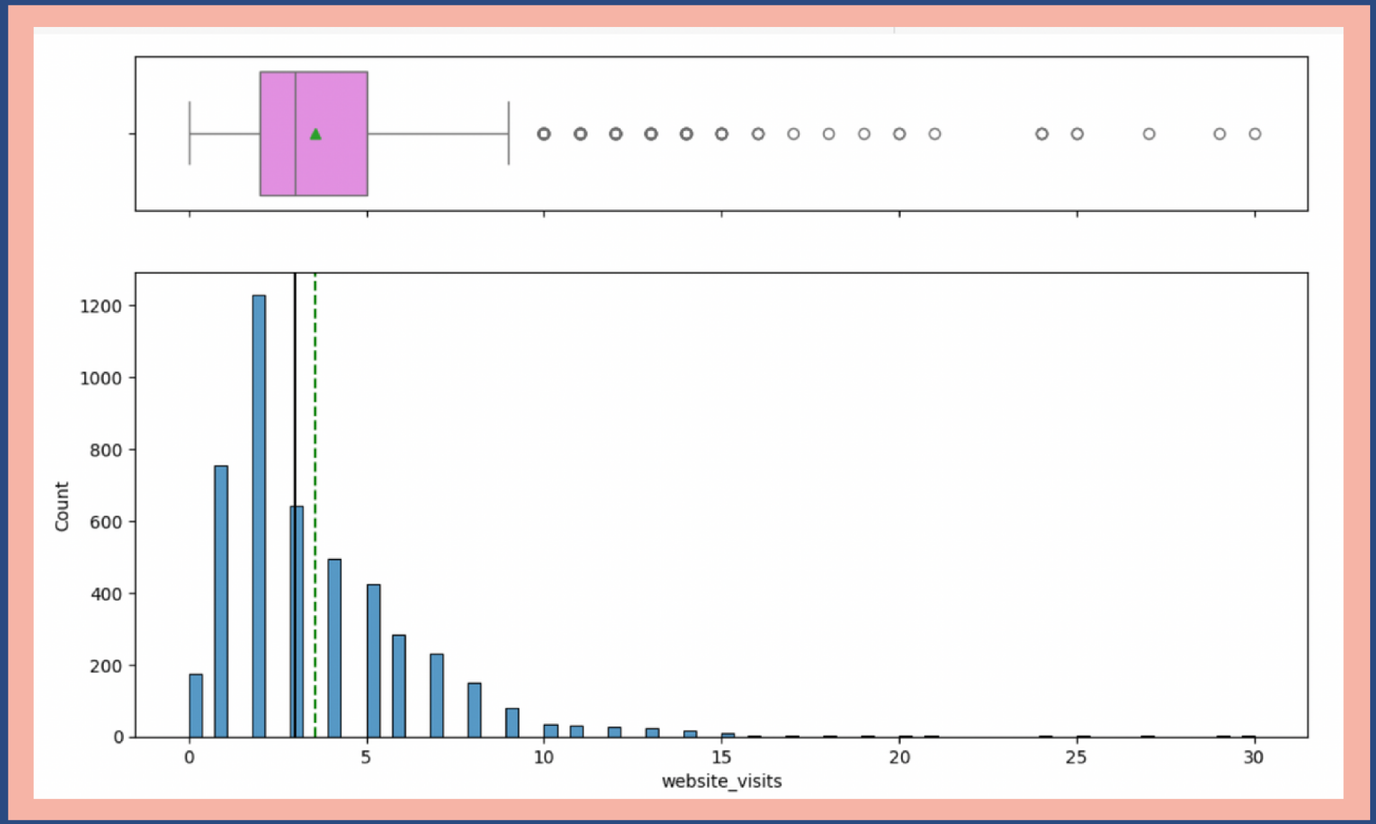
EDA Univariate Results

OBSERVATIONS ON AGE: WE SEE THAT THE MEDIAN AGE IS APPROXIMATELY 51 YEARS OLD. IT IS ALSO NOTABLE THAT THE MEDIAN AGE IS GREATER THAN THE MEAN. THE LARGEST AGE GROUPS ARE IN THEIR LATE FIFTIES AND EARLY SIXTIES.



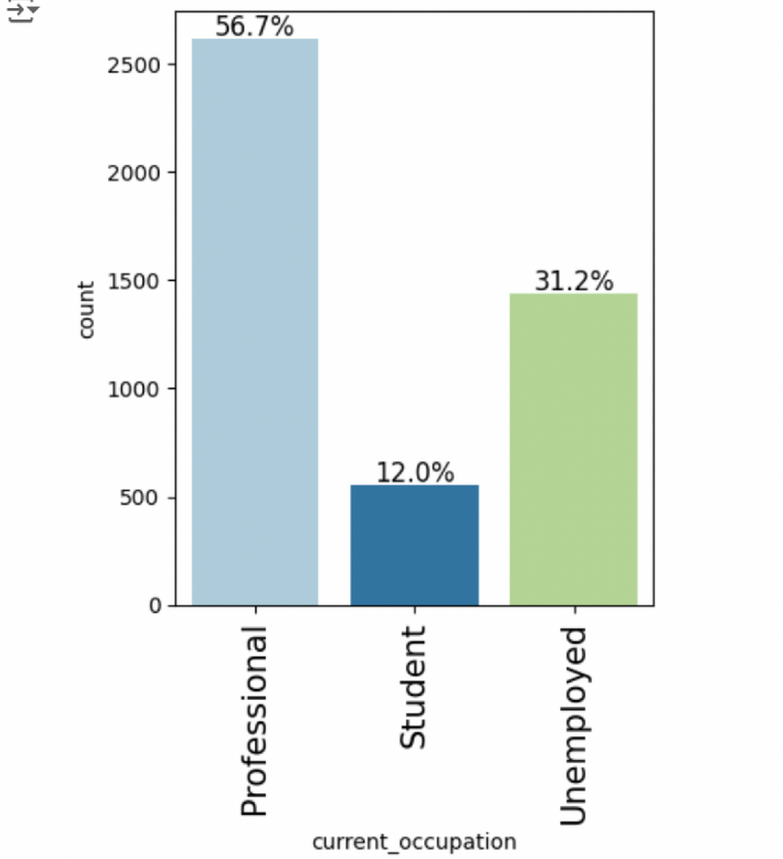
EDA Univariate Results

OBSERVATIONS ON WEBSITE VISITS: WEBSITE VISITS ARE POSITIVELY SKEWED, SHOWING A MEAN AND A MEDIAN OF APPROXIMATELY 3 VISITS. PERHAPS THIS IS THE AMOUNT OF TIMES YOU HAVE TO MAKE AN IMPRESSION ON PROSPECTIVE CLIENTS. THE MEDIAN TIME USERS HAVE SPENT ON THE WEBSITE IS APPROXIMATELY 4 MINUTES. THE AVERAGE TIME USERS HAVE SPENT ON THE WEBSITE IS APPROXIMATELY 12 MINUTES.



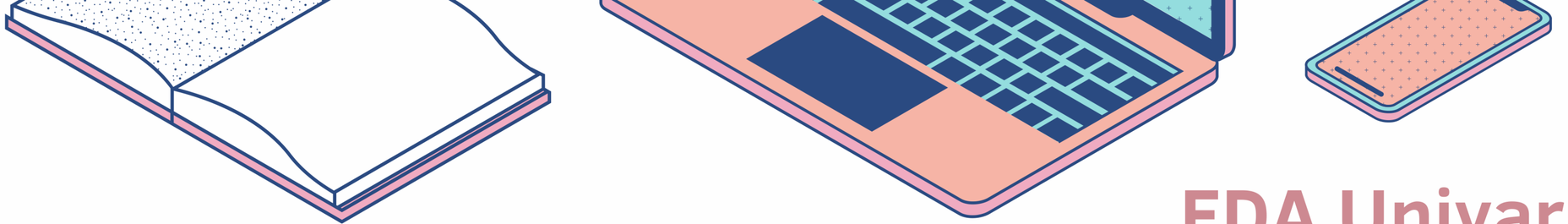
EDA Univariate Results

```
labeled_barplot(data, "current_occupation", perc=True)
```



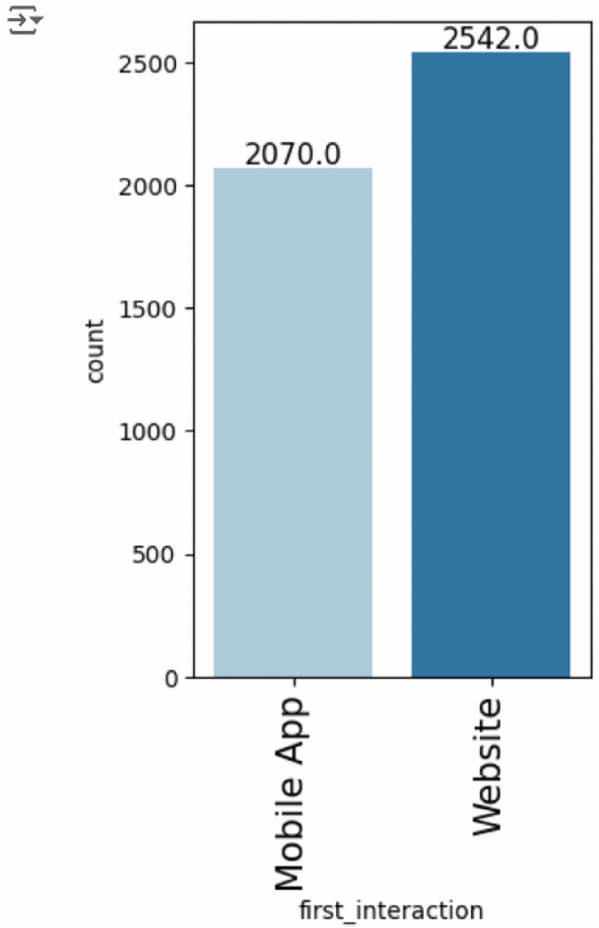
OBSERVATIONS ON OCCUPATION:

56.7% OF THE VISITS TO THE WEBSITE WERE FROM PROFESSIONALS, 31.2% OF THE VISITORS WERE UNEMPLOYED, AND THE SMALLEST GROUP WAS 12%.



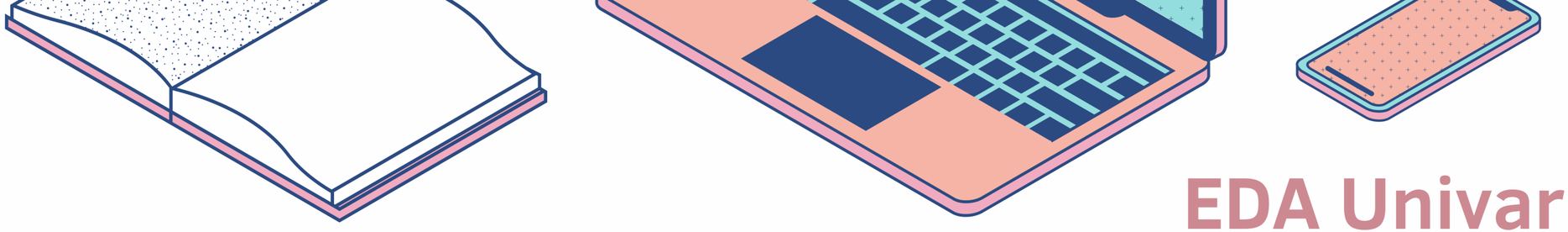
EDA Univariate Results Continued

```
[26] labeled_barplot(data, "first_interaction")
```



OBSERVATIONS ON FIRST INTERACTION:

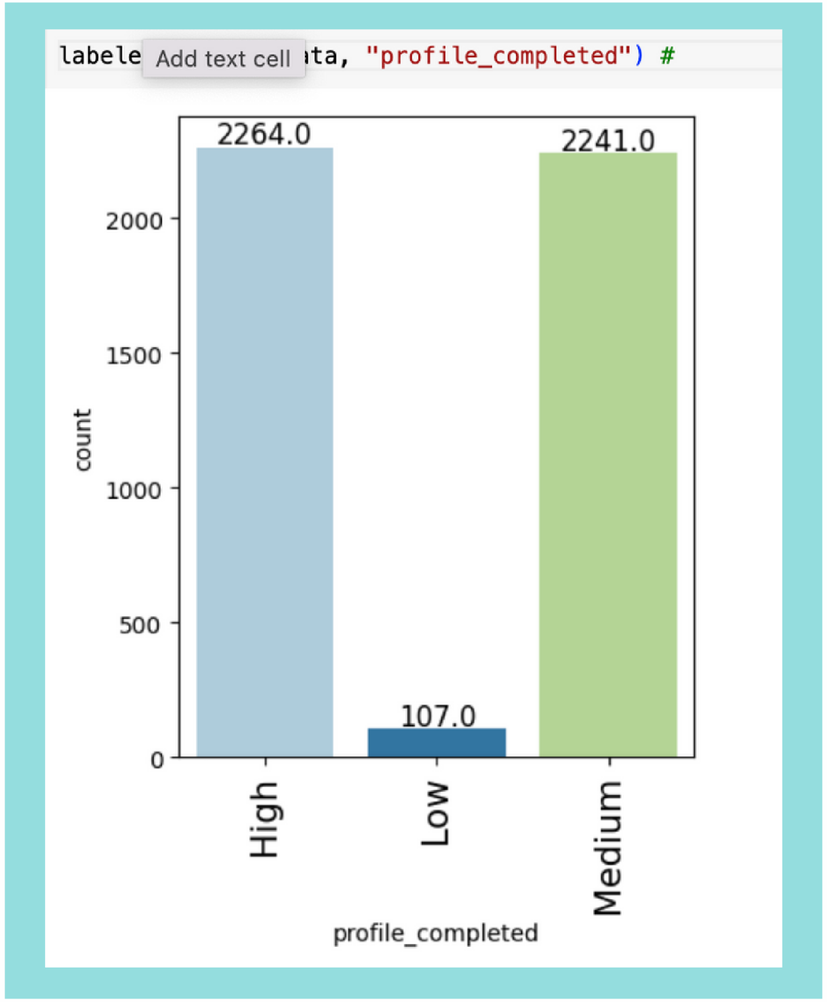
FIRST-TIME VISITORS PREDOMINANTLY CAME FROM WEBSITE VISITORS (55%), AND THE OTHER PORTION (45%) CAME FROM THE MOBILE APP.

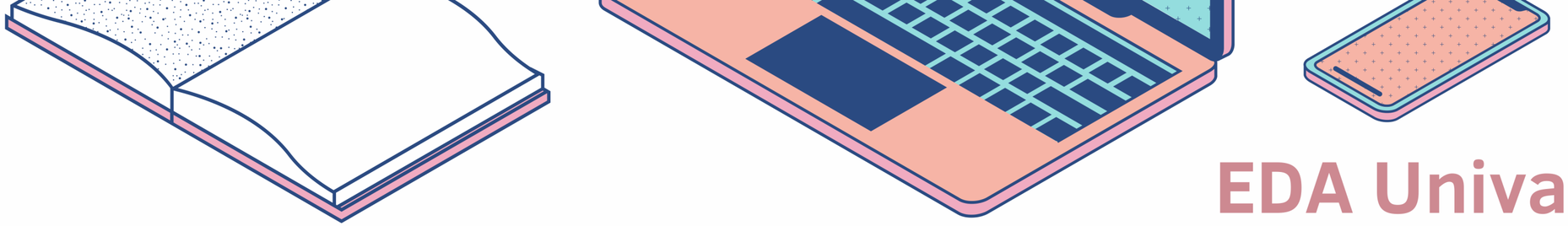


EDA Univariate Results Continued

OBSERVATIONS ON % OF PROFILE COMPLETED:

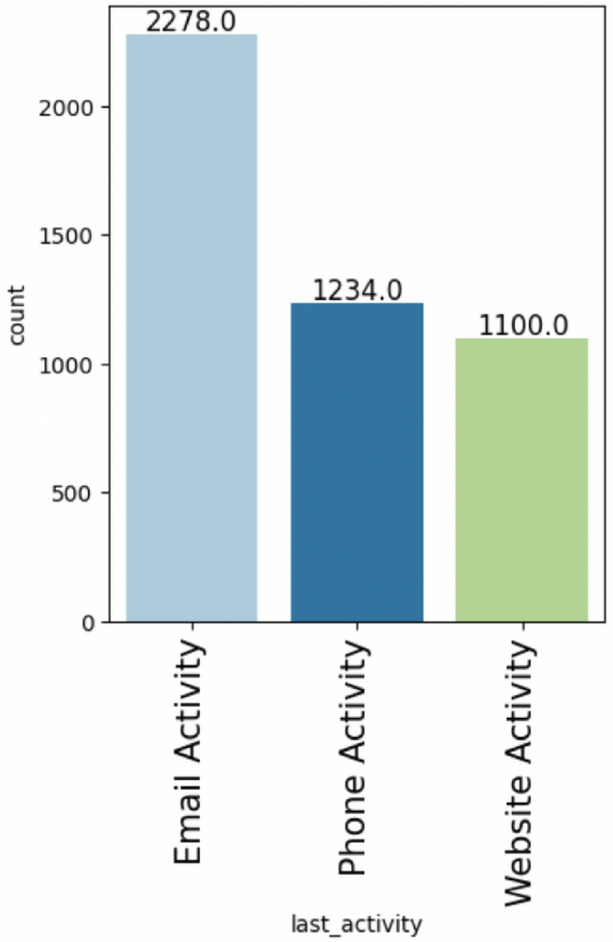
MOST VISITORS EITHER HAVE A HIGH OR MEDIUM COMPLETION RATE FOR THEIR PROFILES. ONLY 2% OF THE USERS IN THE DATA SET HAD A LOW PROFILE COMPLETION.





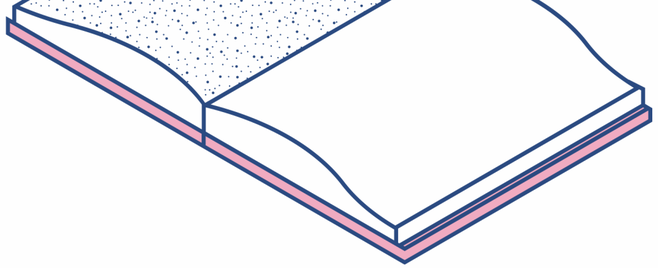
EDA Univariate Results Continued

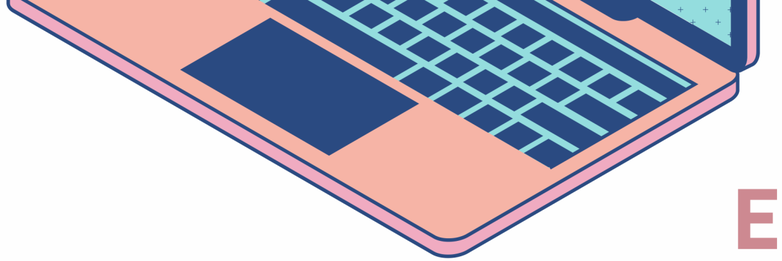
```
labeled_barplot(data, "last_activity") #
```

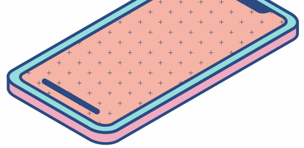


THE LAST ACTIVITY FOR MOST LEADS WAS AN EMAIL, FOLLOWED BY PHONE INTERACTION, AND LASTLY, WEBSITE ACTIVITY.

THIS SHOWS THAT HIGH PRIORITY SHOULD BE PLACED ON EMAIL RESPONSES TO PROSPECTS.

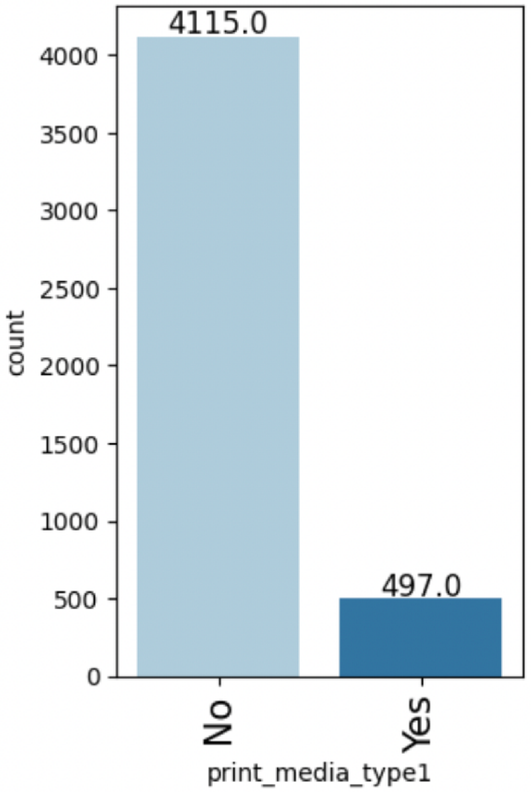






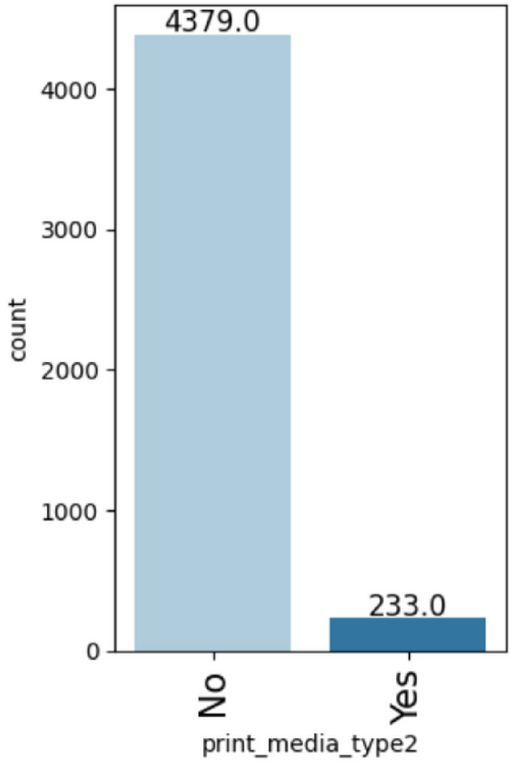
EDA Univariate Results Continued

```
labeled_barplot(data, "print_media_type1")
```



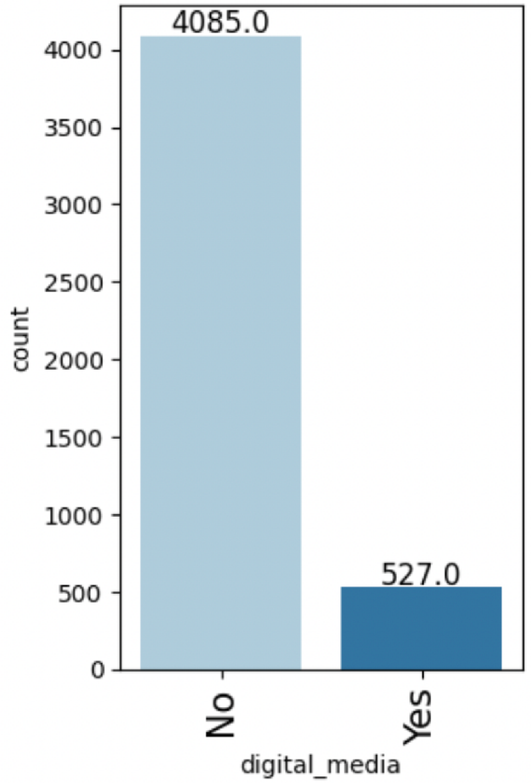
newspaper

```
labeled_barplot(data, "print_media_type2")
```



magazine

```
labeled_barplot(data, "digital_media") #
```

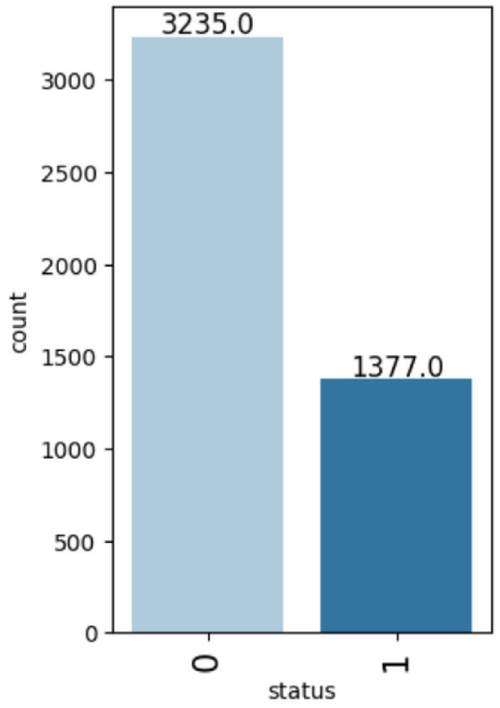


digital media

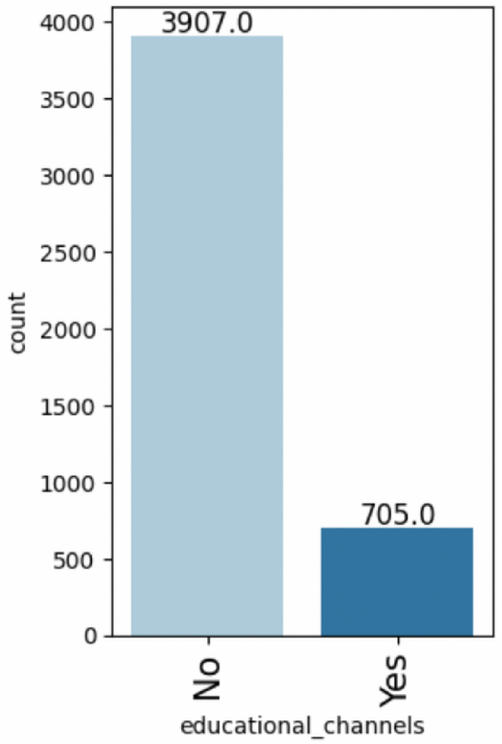
EDA Univariate Results Continued

APPROXIMATELY 30% OF LEADS CURRENTLY ARE CONVERTING TO PAID CUSTOMERS

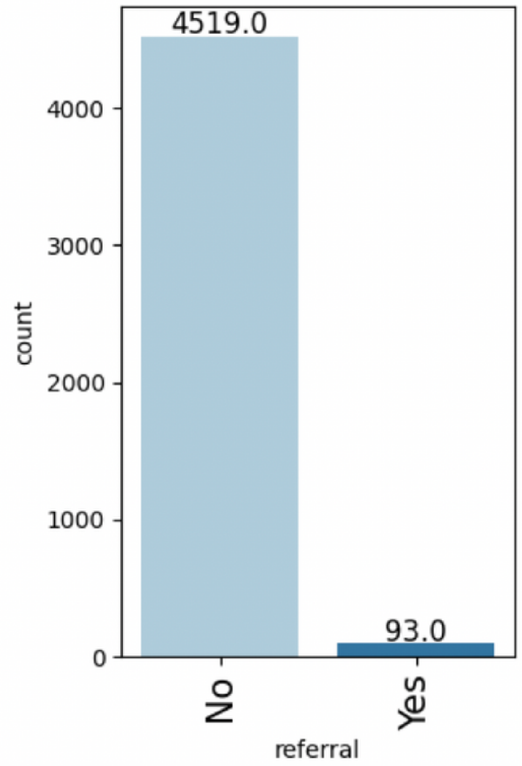
```
labeled_barplot(data, "status")
# 0 = not converted
# 1 = yes converted
```




```
labeled_barplot(data, "educational_channe
```




```
labeled_barplot(data, "referral") #
```



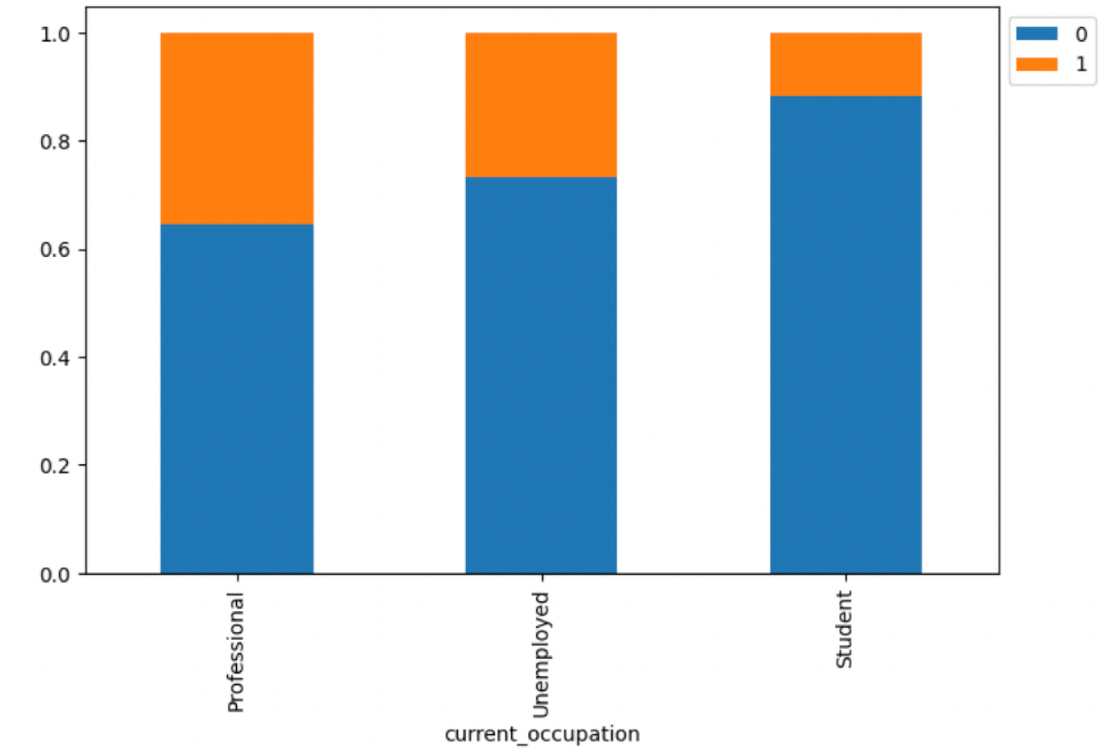
EDA Multivariate Results

OBSERVATIONS ON BIVARIATE ANALYSIS: THE GREATER THE TIME ON THE WEBSITE, THE GREATER THE CHANCE OF CONVERTING TO PAID CUSTOMER.

EDA Multivariate Results

```
stacked_barplot(data, "current_occupation", "status")
```

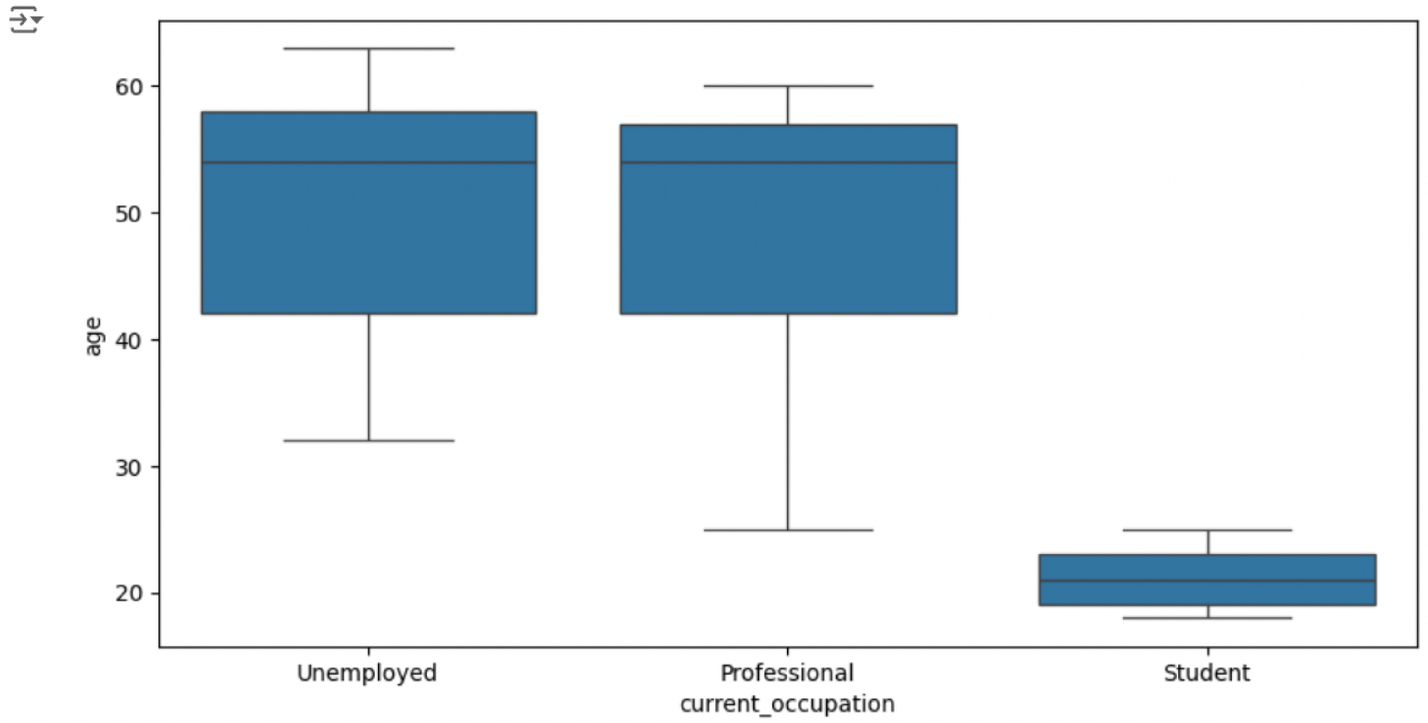
```
status                0     1   All
current_occupation
All                3235  1377  4612
Professional       1687   929  2616
Unemployed         1058   383  1441
Student             490    65   555
```



EDA Multivariate Results

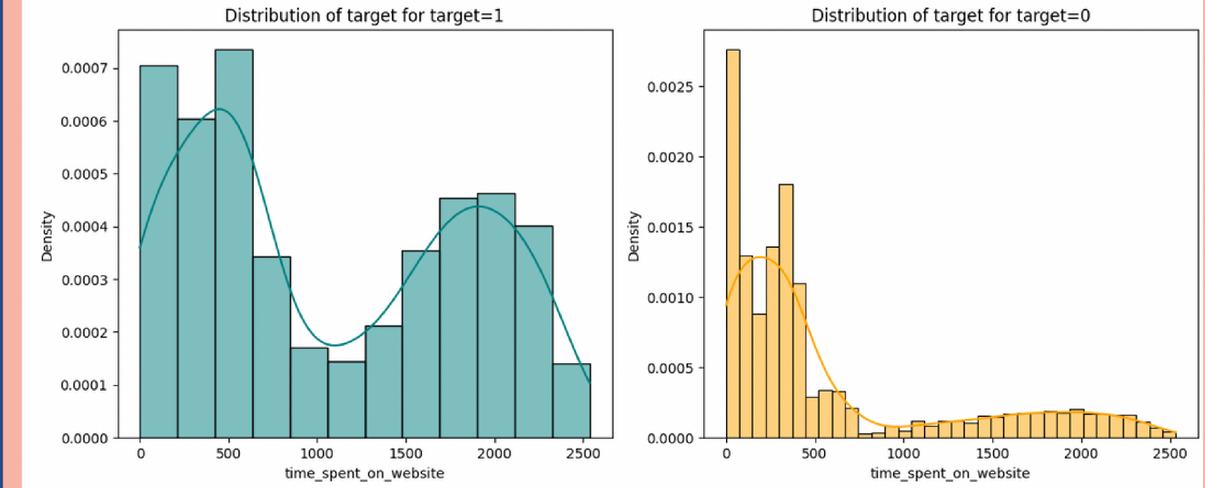
Age can be a good factor to differentiate between such leads

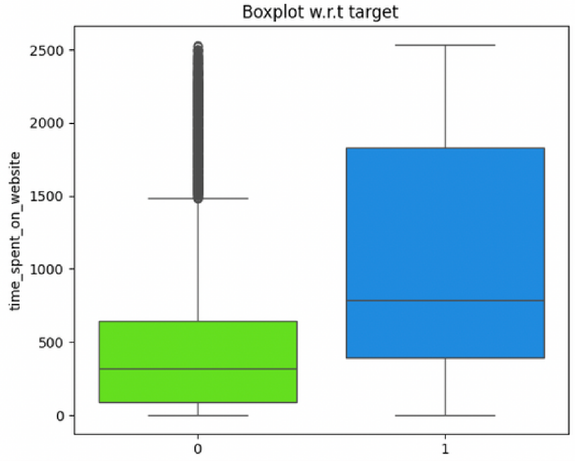
```
plt.figure(figsize=(10, 5))
sns.boxplot(data = data, x = data["current_occupation"], y = data["age"])
plt.show()
```

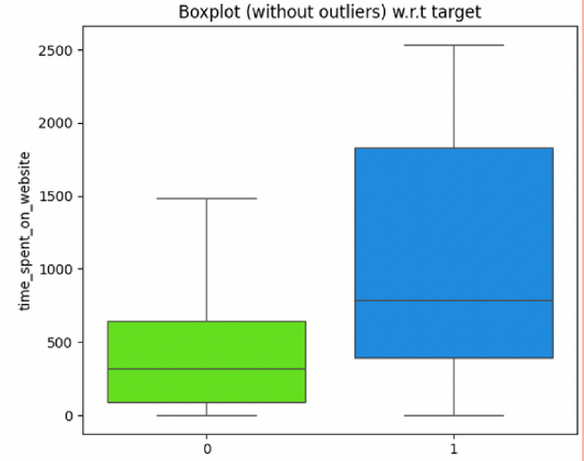


EDA Multivariate Results

```
distribution_plot_wrt_target(data, "time_spent_on_website", "status")
```

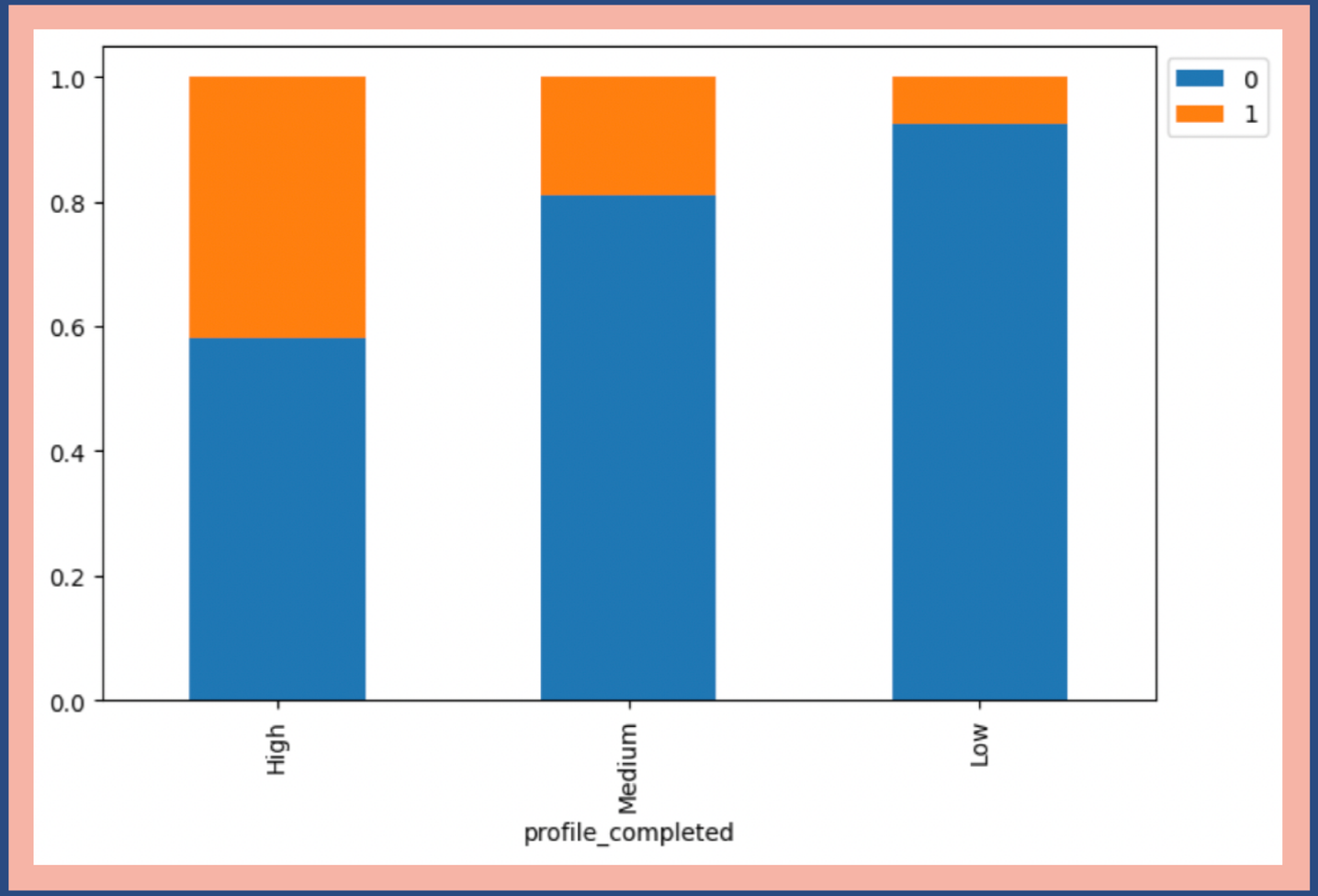






EDA Multivariate Results

OBSERVATIONS ON BIVARIATE ANALYSIS: PROFILE COMPLETION & CONVERSION

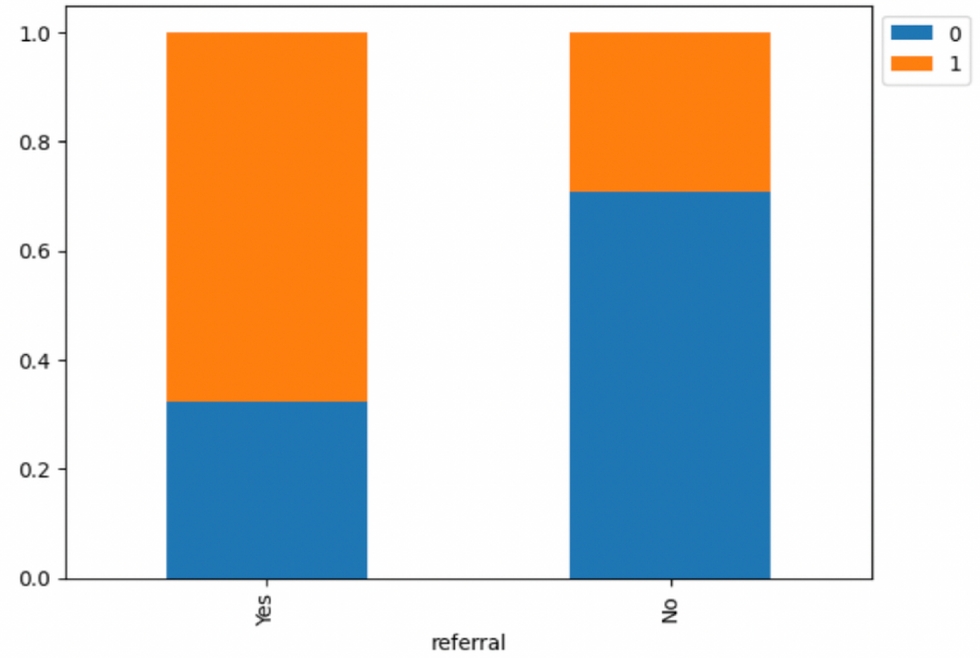


EDA Multivariate Results

OBSERVATIONS ON BIVARIATE ANALYSIS: REFERRAL & CONVERSION

```
stacked_barplot(data, "referral", "status")
```

status	0	1	All
referral			
All	3235	1377	4612
No	3205	1314	4519
Yes	30	63	93



DATA PREPROCESSING

- ENCODING CATEGORICAL FEATURES
- SPLITTING THE DATA INTO TRAIN AND TEST
- EVALUATING THE MODEL THAT WE BUILD ON THE TRAINING DATA

```
X = data.drop(["status"], axis=1)
Y = data["status"]

X = pd.get_dummies(X, drop_first=True)

X_train, X_test, y_train, y_test = train_test_split(
    X, Y, test_size=0.30, random_state=1
)
```

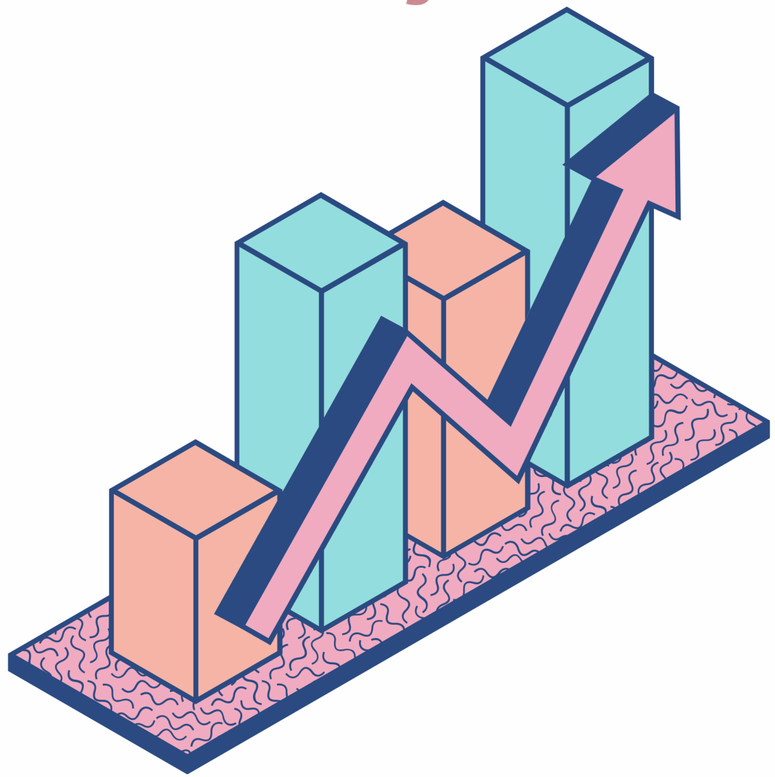
```
[65] print("Shape of Training set : ", X_train.shape)
     print("Shape of test set : ", X_test.shape)
     print("Percentage of classes in training set:")
     print(y_train.value_counts(normalize=True))
     print("Percentage of classes in test set:")
     print(y_test.value_counts(normalize=True))
```

```
Shape of Training set :  (3228, 16)
Shape of test set :  (1384, 16)
Percentage of classes in training set:
status
0   0.70415
1   0.29585
Name: proportion, dtype: float64
Percentage of classes in test set:
status
0   0.69509
1   0.30491
Name: proportion, dtype: float64
```

Model Performance Summary

TWO WRONG PREDICTIONS:

1. **PREDICTING A LEAD WILL NOT BE CONVERTED TO A PAID CUSTOMER IN REALITY, THE LEAD WOULD HAVE CONVERTED TO A PAID CUSTOMER.**
2. **PREDICTING A LEAD WILL BE CONVERTED TO A PAID CUSTOMER IN REALITY, THE LEAD WOULD NOT HAVE CONVERTED TO A PAID CUSTOMER.**



IF WE PREDICT THAT A LEAD WILL NOT GET CONVERTED AND THE LEAD WOULD HAVE CONVERTED THEN THE COMPANY SUFFER A GREATER LOSS BY LOSING A CUSTOMER.

BUILDING MODEL TREE

Model Performance Summary Continued

Building Decision Tree Model

```
# Fitting the decision tree classifier on the training data
d_tree = DecisionTreeClassifier(random_state=1)

# Fit the classifier
d_tree.fit(X_train, y_train)
```

```
DecisionTreeClassifier(random_state=1)
```

Checking model performance on training set

```
# Make predictions on the training data
y_pred_train = d_tree.predict(X_train)

# Evaluate the model using the training data
metrics_score(y_train, y_pred_train)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2273
           1       1.00      1.00      1.00       955

    accuracy                           1.00      3228
   macro avg       1.00      1.00      1.00      3228
weighted avg       1.00      1.00      1.00      3228
```

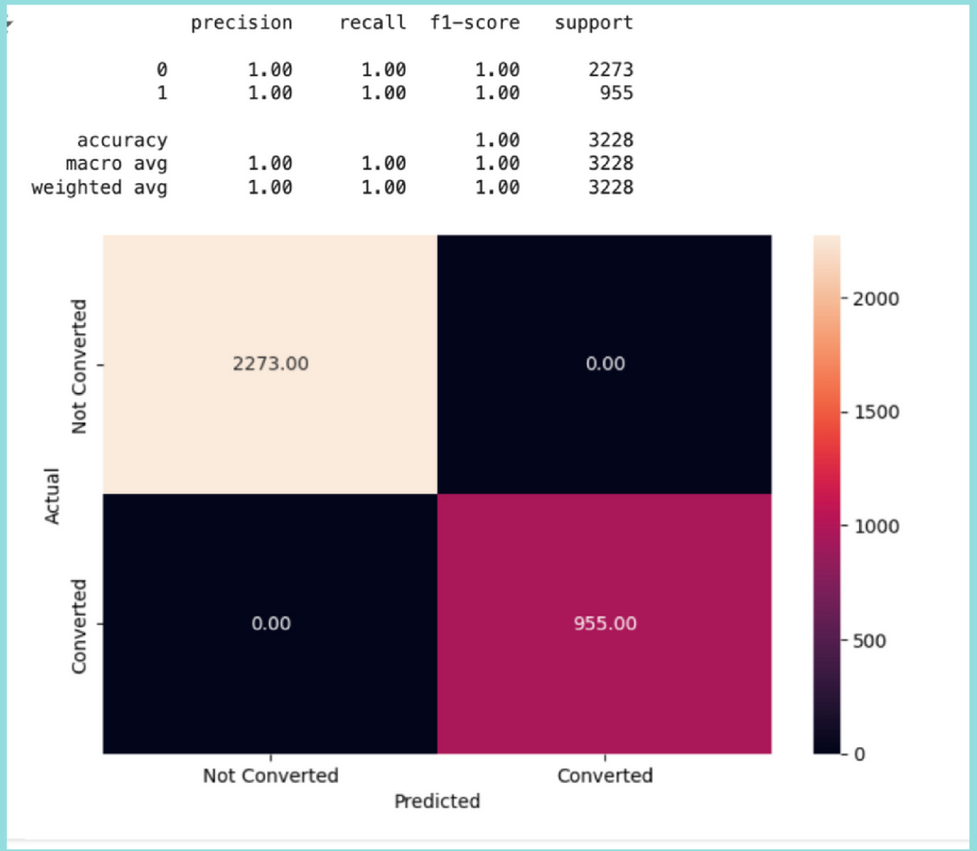
BUILDING MODEL TREE

Model Performance Summary Continued

Checking model performance on training set

```
# Make predictions on the training data
y_pred_train = d_tree.predict(X_train)

# Evaluate the model using the training data
metrics_score(y_train, y_pred_train)
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2273
1	1.00	1.00	1.00	955
accuracy			1.00	3228
macro avg	1.00	1.00	1.00	3228
weighted avg	1.00	1.00	1.00	3228

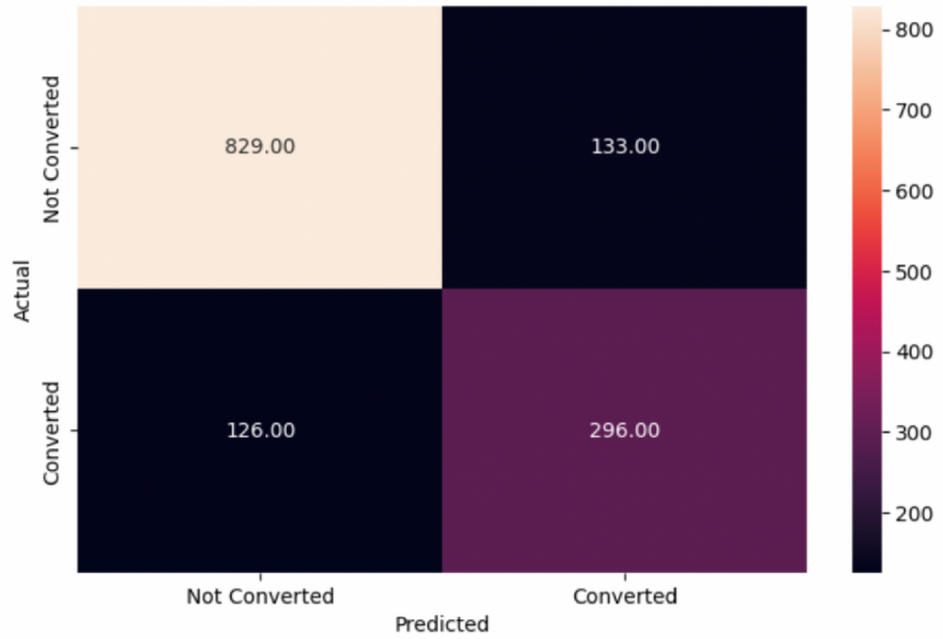


MODEL IS OVERFITTING!

Model Performance Summary Continued

```
# Checking performance on the testing data

# Make predictions on the test data
y_pred_test1 = d_tree.predict(X_test)

# Evaluate the model using the test data
metrics_score(y_test, y_pred_test)
```

```
              precision    recall  f1-score   support

           0       0.87      0.86      0.86       962
           1       0.69      0.70      0.70       422

    accuracy                           0.81      1384
   macro avg       0.78      0.78      0.78      1384
weighted avg       0.81      0.81      0.81      1384
```

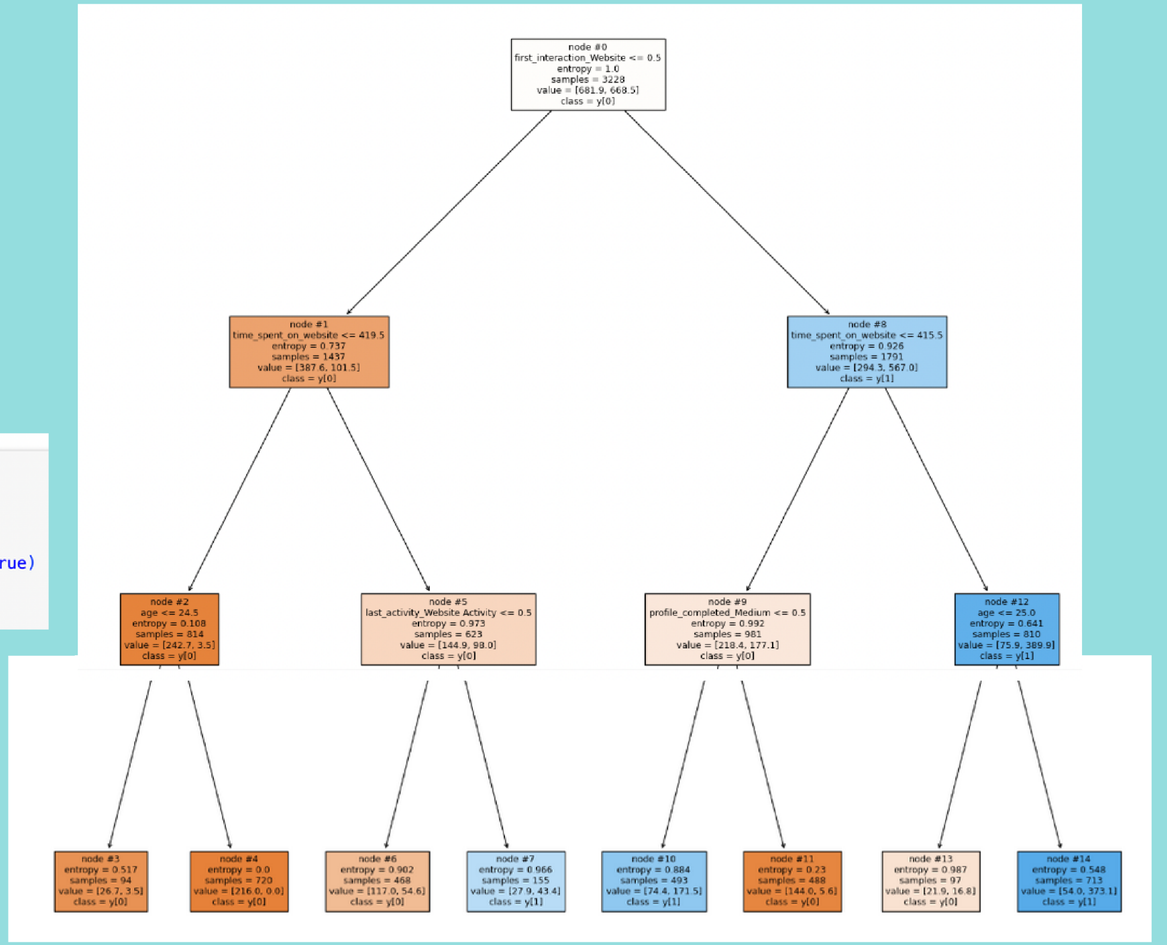


OVERFITTING CONFIRMED

VISUALIZING DECISION TREE

```
features = list(X.columns)

plt.figure(figsize = (20, 20))

tree.plot_tree(d_tree_tuned, feature_names = features, filled = True, fontsize = 9, node_ids = True, class_names = True)

plt.show()
```

Model Performance Summary Continued



FINAL MODEL

Model Performance Summary Continued

```
RandomForestClassifier(class_weight='balanced', max_depth=6, max_features=0.9,
                       max_samples=0.9, min_samples_leaf=20, n_estimators=110,
                       random_state=1)
```

```
[86] # Checking performance on the training data
     y_pred_train4 = rf_estimator_tuned.predict(X_train)

     # Evaluate the model using the training data
     metrics_score(y_train, y_pred_train4)
```

	precision	recall	f1-score	support
0	0.94	0.83	0.88	2273
1	0.69	0.87	0.77	955
accuracy			0.84	3228
macro avg	0.81	0.85	0.83	3228
weighted avg	0.86	0.84	0.85	3228

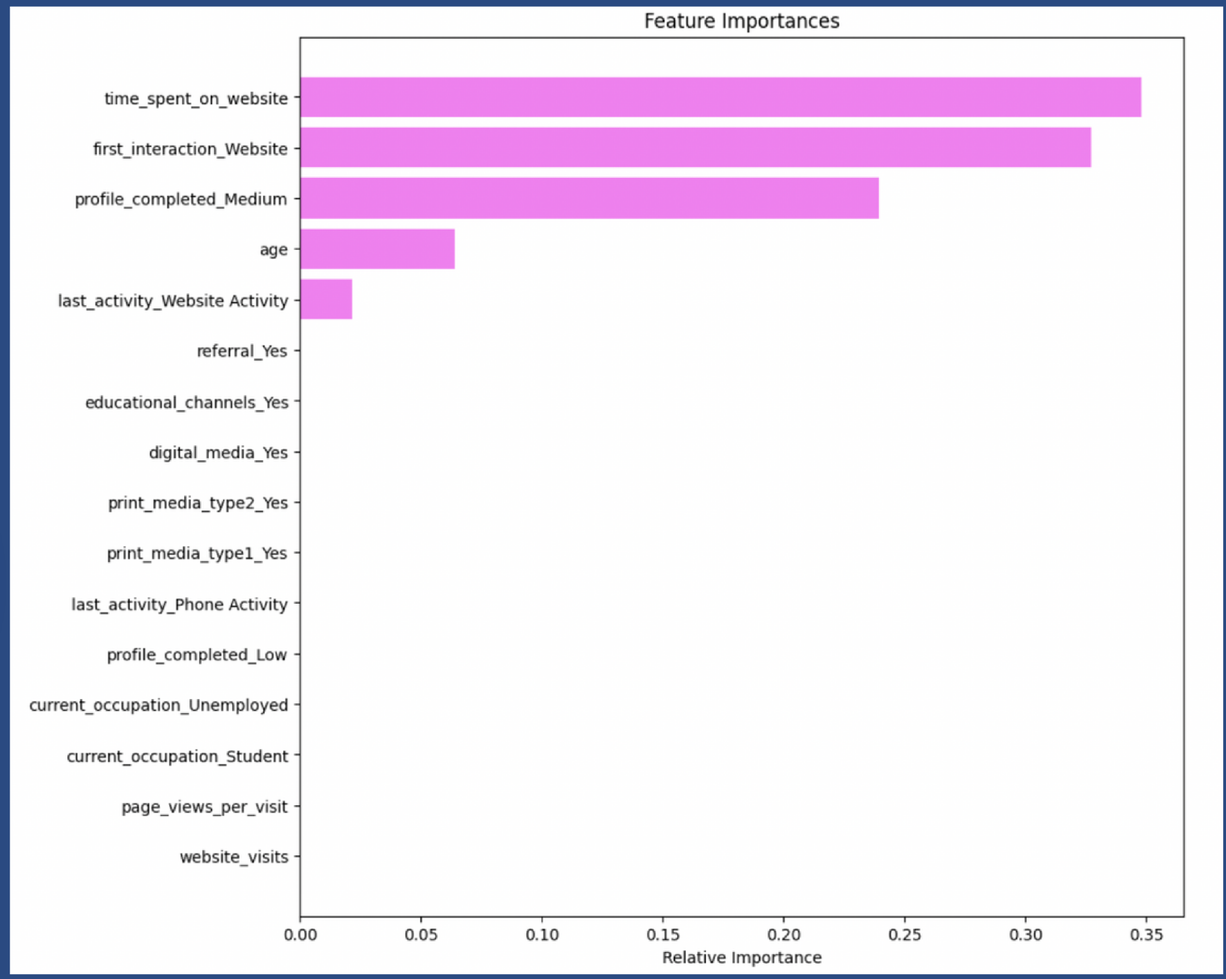
Let's check the model performance on the test data

```
# Checking performance on the test data
y_pred_test4 = rf_estimator_tuned.predict(X_test)

# Evaluate the model using the test data
metrics_score(y_test, y_pred_test4)
```

	precision	recall	f1-score	support
0	0.93	0.84	0.88	962
1	0.70	0.85	0.77	422
accuracy			0.84	1384
macro avg	0.81	0.85	0.83	1384
weighted avg	0.86	0.84	0.85	1384

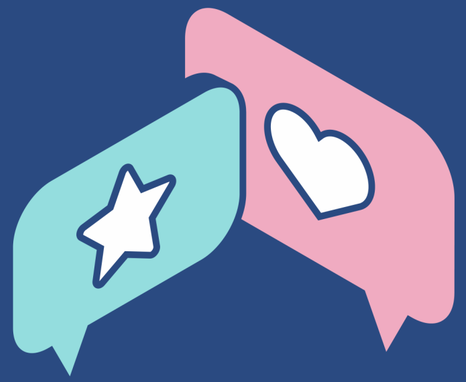
Conclusion and Recommendations

AFTER TUNING THE HYPER PARAMETERS AND GETTING THE RIGHT MODEL, WE FOUND THAT TIME SPENT ON THE WEBSITE AND FIRST INTERACTIONS LED TO CONVERSION.



BUSINESS SUGGESTION:

- PROMPT FOR EMAIL INTERACTIONS
- GET A CHAT POP UP ON MOBILE AND WEB DEVICES TO ANSWER QUERIES QUICKLY
- TARGET HIGH PROFILE COMPLETION & MEDIAN AGE GROUPS



Thank You

Massachusetts Institute of Technology

IDSS
MIT INSTITUTE FOR DATA, SYSTEMS, AND SOCIETY

Great Learning
POWER AHEAD