# *User Activity Logs*

Capstone project of Machine Learning Track

فنتك السعودية
FintechSaudi

**Presented by:**
**Desert Ninjas**

**Date Presented:**
Nov 20th, 2022

# Outline

فنتك السعودية
FintechSaudi

# *Team Members*

فنتك السعودية
FintechSaudi

*Desert Ninjas*

Team under big data and artificial intelligence bootcamp.

**Reema Alaswad**
**Raghad Aleisa**
**Eman Aldosari**
**Maha Alhazzani**
**Aljohara Alkanhal**

# Introduction

# Company Overview

Established since 2020

Listed under FinTech lab initiative sandbox, supervised by the Capital Market Authority

Highest growing FinTech startup in 2 years

Contributes to the whole Saudi economic growth and 2030 vision (i.e., GDP, job opportunities)

Highest consumers retention rates, amongst other Saudi FinTech's

# Problem Statement

فنتك السعودية
FintechSaudi

A startup **FinTech** company named X is interested in knowing its customers' behaviors and **whether they're going to invest** based on their users activity logs

## Challenges

- The number of users is unknown to us
- No useful features
- No users demographics
- Huge preprocessing time

# Dataset Overview

Sep 8th → Sep 15th
2022

User Activity Logs

95K Rows

3K Visitors

5th Path is Longest Path

# Objective

## Main Objectives

1. Predict customer behavior and activity logs to see whether the customer would invest in the company.

2. Predicting the potential investors to target them with marketing strategies.

# Questions

1. **What kind of data does their website collect from users?**
2. **What is the path that gets visited by users usually? And how much time do users spend on this path?**
3. **Does the average time spent on a page differ based on the user type?**
4. **Which path has the maximum time?  Is this the path that leads to a successful transaction (investment)?**

# Data Preprocessing

# Data preprocessing

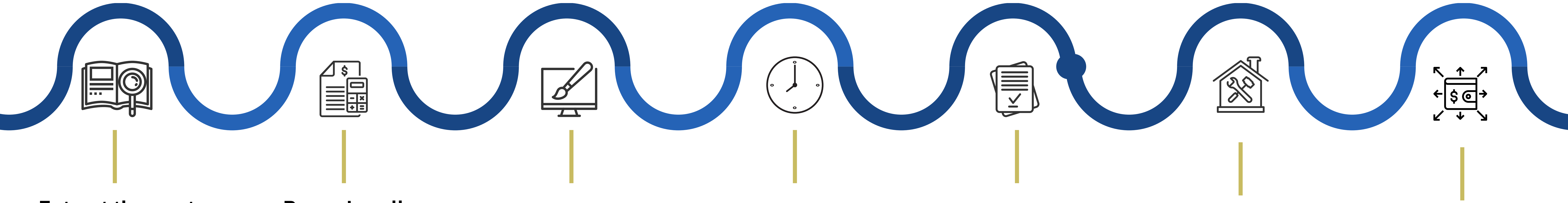
## Before data preprocessing process:

{"$os":"Windows","$browser":"Chrome","$device_type":"Desktop","$current_url":"https://company.sa/investor/dashboard","$host":"company.sa","$pathname":"/investor/dashboard","$browser_version":105,"$screen_height":864,"$screen_width":1536,"$viewport_height":714,"$viewport_width":1536,"$lib":"web","$lib_version":"1.26.0","$insert_id":"ryqkpj2yyv4ob7iq","$time":1662659942.785,"distinct_id":"1831e41502a288-0aadb2e8522fed-26021c51-144000-1831e41502b504","$device_id":"1831e41502a288-0aadb2e8522fed-26021c51-144000-1831e41502b504","$referrer":"$direct","$referring_domain":"$direct","$active_feature_flags":[],"$event_type":"click","$ce_version":1,"token":"phc_TfMQhNNAvw1adnHEWilG1LMpLeszOjUV5y1X6EXAqwR","$session_id":"1831e4150303a0-04a83b401e337e-26021c51-144000-1831e4150316d8","$window_id":"1831e4150327d1-0f02fd620ee793-26021c51-144000-1831e4150338f5","$set_once":{"$initial_os":"Windows","$initial_browser":"Chrome","$initial_device_type":"Desktop","$initial_current_url":"https://company.sa/investor/dashboard","$initial_pathname":"/investor/dashboard","$initial_browser_version":105,"$initial_referrer":"$direct","$initial_referring_domain":"$direct"},"$

## After data preprocessing process:

| session_id | Type | window_id | browser | device_type | os | host | current_url |
|---|---|---|---|---|---|---|---|
| 1831e41503 | pageview | 1831e41503 | Chrome | Desktop | Windows | company.sa | https//comp |
| 1831fa67f29 | pageview | 1831fa67f2b | Chrome | Desktop | Windows | company.sa | https//comp |
| 18327baf015 | pageview | 18327e6980 | Chrome | Desktop | Windows | company.sa | https//comp |
| 1832bff5050 | pageview | 1832bff5053 | Chrome | Desktop | Windows | company.sa | https//comp |
| 1832ce7991 | pageview | 1832ce7991 | Chrome | Desktop | Mac OS X | company.sa | https//comp |
| 18330ae004 | pageview | 18330ae004 | Mobile Safar | Mobile | iOS | company.sa | https//comp |
| 18330ae082 | pageview | 18330ae082 | Mobile Safar | Mobile | iOS | company.sa | https//comp |
| 18330ae40b | pageview | 18330ae40b | Microsoft Ed | Desktop | Windows | company.sa | https//comp |
| 18330aeb0c | pageview | 18330aeb0c | Microsoft Ed | Desktop | Windows | company.sa | https//comp |
| 18330b065d | pageview | 18330b065d | Mobile Safar | Mobile | iOS | company.sa | https//comp |

# *Data preprocessing*



Extract the most important columns.

Removing all unwanted symbols/signs.

Columns type conversion

Handling Time column

Generate new columns

Handling missing values

Labeling the dataset

# 1. Extract the Most Important Columns.

```
def Extract_column(data , column):
  if(column in data):
    data = str(data)
    data = data.split(column)[1]
    data = data.split(',')[0]
    data = data.replace('$','')
    data = data.replace(':','')
    data = data.replace('"','')
    return data

  for column in columns:
    df2[column] = df2['Raw data'].apply(lambda data : Extract_column(data, column))
```

# 1. Extract the Most Important Columns.

## Before

| Raw data |
| --- |
| 'os ":" Windows ", "browser":"Chrome","$device_.. |
| 'os ":" Windows ", "browser":"Chrome","$device_.. |
| 'os ":" Windows ", "browser":"Chrome","$device_.. |
| 'os ":" Windows ", "browser":"Chrome","$device_.. |
| 'os ":" Windows ", "browser":"Chrome","$device_.. |

## After

| os | browser | |
| --- | --- | --- |
| {"$os":"Windows" | $browser:"Chrome" | $current_url:"https://compa |
| {"$os":"Windows" | $browser:"Chrome" | $current_url:"https://comp |
| {"$os":"Windows" | $browser:"Chrome" | $current_url:"https://comp |
| {"$os":"Windows" | $browser:"Chrome" | $current_url:"https://comp |

# 2. Removing All Unwanted Symbols/Signs.

```python
def Remove_signs(x):
    x = str(x)
    name = x.split(':')[0]
    x = x.replace(name,'')
    x = x.replace('$','') #Remove $ symbol
    x = x.replace(':','') #Remove : symbol
    x = x.replace('"','') #Remove " symbol
    return x
```

# 2. Removing All Unwanted Symbols/Signs.

**Before**

| os | browser | current_url |
|---|---|---|
| {"$os":"Windows" | $browser:"Chrome" | $current_url:"https://company.sa/investor/dash... |
| {"$os":"Windows" | $browser:"Chrome" | $current_url:"https://company.sa/investor/inve... |
| {"$os":"Windows" | $browser:"Chrome" | $current_url:"https://company.sa/investor/inve... |
| {"$os":"Windows" | $browser:"Chrome" | $current_url:"https://company.sa/investor/inve... |
| {"$os":"Windows" | $browser:"Chrome" | $current_url:"https://company.sa/investor/inve... |

**After**

| os | browser | current_url | host |
|---|---|---|---|
| Windows | Chrome | https//company.sa/investor/dashboard | company.sa |
| Windows | Chrome | https//company.sa/investor/investment-portfolio | company.sa |
| Windows | Chrome | https//company.sa/investor/investment-portfolio | company.sa |
| Windows | Chrome | https//company.sa/investor/investment-portfolio | company.sa |
| Windows | Chrome | https//company.sa/investor/investment-portfolio | company.sa |

# 3. Columns Type Conversion

```
df[column]= pd.to_numeric(df.column, errors='coerce')
df[column].fillna(df[column].mean(), inplace=True)
```

# 3. Columns Type Conversion

*Before*

| | | | | |
|---|---|---|---|---|
| 6 | browser_version | 95818 | non-null | object |
| 7 | screen_height | 95818 | non-null | object |
| 8 | screen_width | 95818 | non-null | object |
| 9 | viewport_height | 95818 | non-null | object |
| 10 | viewport_width | 95818 | non-null | object |

*After*

| | | | | |
|---|---|---|---|---|
| 27 | browser_version | 2949 | non-null | float64 |
| 28 | screen_height | 2949 | non-null | int64 |
| 29 | screen_width | 2949 | non-null | int64 |
| 30 | viewport_height | 2949 | non-null | float64 |
| 31 | viewport_width | 2949 | non-null | float64 |

# 4. Handling Time column

## 4.1 Timestamp conversion:

- Currently, The time format in "Time" column is in the float representing a Unix epoch in units of seconds.

```
def convert_to_timestamp(x):
    x = float(x)
    x = pd.Timestamp(x, unit='s')
    return x
```

# 4. Handling Time column

| Before | | After | |
|---|---|---|---|
| 0 | 1662659942.785 | 0 | 2022-09-08 17:59:02.785000086 |
| 1 | 1662660028.357 | 1 | 2022-09-08 18:00:28.357000113 |
| 2 | 1662660365.581 | 2 | 2022-09-08 18:06:05.581000090 |
| 3 | 1662660346.708 | 3 | 2022-09-08 18:05:46.707999945 |
| 4 | 1662660067.629 | 4 | 2022-09-08 18:01:07.628999949 |

# 4. Handling Time column

## 4.2 Remove the sub-seconds "parts of a second” from “Time” column:

17:59:02.785000

```
def remove_micro_seconds(time):
    time=str(time).split('.')[0]
    return time
```

# 4. Handling Time column

**Before**

| | |
|---|---|
| 0 | 17:59:02.785000 |
| 1 | 18:00:28.357000 |
| 2 | 18:06:05.581000 |

**After**

| | |
|---|---|
| 0 | 17:59:02 |
| 1 | 18:00:28 |
| 2 | 18:06:05 |

# 5. Generate New Columns

## 5.1 Generate new columns by splitting the pathname into different columns

- Max path length is 5.

```
df[['path1', 'path2', 'path3', 'path4', 'path5']] = df_concat['pathname'].str.split('/' ,
                                                                                expand= True)
```

# 5. Generate New Columns

## The Result:

| pathname | path1 | path2 | path3 | path4 | path5 |
|---|---|---|---|---|---|
| /investor/opportunity/WX0 | investor | opportunity | WX0 | blank | blank |

# *5. Generate New Columns*

## 5.2 Generate "Number of Pages" column

- Calculate the pages based on the number of backslashes "/" in each pathname.

```
#This function will calculate how many pages have the user entered using the "/" symbol.
def count_pages (path):
  number_of_pages = path.count('/')
  return number_of_pages
```

# 5. Generate New Columns

**The Result:**

| path1 | path2 | path3 | path4 | path5 | number_of_pages |
|---|---|---|---|---|---|
| investor | dashboard | blank | blank | blank | 2 |
| investor | investment-portfolio | blank | blank | blank | 2 |

# 5. Generate New Columns

## 5.3 Separate the date and time into two new and different columns.

```
#Separate the date and time into two new and different columns.
df['new_date'] = [d.date() for d in df['time']]
df['new_time'] = [d.time() for d in df['time']]
```

# 5. Generate New Columns

**The Result:**

| | time |
|---|---|
| | 2022-09-08 17:59:02.785000086 |
| | 2022-09-08 18:00:28.357000113 |

| date | time |
|---|---|
| 2022-09-08 | 17:59:02.785000 |
| 2022-09-08 | 18:00:28.357000 |

# 5. Generate New Columns

## 5.4 Extract the year, month, and day from "date" column.

```
#Extract the year, month and day from "date" column.
df['year'] = pd.DatetimeIndex(df['date']).year
df['month'] = pd.DatetimeIndex(df['date']).month
df['day'] = pd.DatetimeIndex(df['date']).day
```

# 5. Generate New Columns

**The Result:**

| date | time | year | month | day |
|---|---|---|---|---|
| 2022-09-08 | 17:59:02.785000 | 2022 | 9 | 8 |
| 2022-09-08 | 18:00:28.357000 | 2022 | 9 | 8 |

# 5. Generate New Columns

## 5.4 Extract month name and day name from "date" column.

```python
#This function will extract the name of the month "not the number". e.g. September.
def get_month_name(date):
    d = pd.Timestamp(date)
    return d.month_name()
#This function will extract the name of the day "not the number". e.g. Thursday.
def get_day_name(date):
    d = pd.Timestamp(date)
    return d.day_name()
```

# *5. Generate New Columns*

**The Result:**

| year | month | day | month_name | day_name |
|---|---|---|---|---|
| 2022 | 9 | 8 | September | Thursday |
| 2022 | 9 | 8 | September | Thursday |

# *5. Generate New Columns*

**5.5 Classify the days as weekend or as weekday.**

- Sunday, Monday, Tuesday, Wednesday, Thursday → Weekday
- Friday, Saturday → Weekend

# 5. Generate New Columns

**The Result:**

| year | month | day | month_name | day_name | week_label |
|---|---|---|---|---|---|
| 2022 | 9 | 8 | September | Thursday | Weekday |
| 2022 | 9 | 8 | September | Thursday | Weekday |

# 5. Generate New Columns

**5.6 Categorize time into different categorical classes based on the week label.**

In **weekdays**:

1. From 0:00 to 7:59 ➔ Morning.
2. From 8:00 to 11:59 and from 13:00 to 16:59 ➔ During work.
3. From 12:00 till 12:59 ➔ Launch hour.
4. Otherwise ➔ Night.

In **weekend**:

1. From 4:00 to 11:59 ➔ Morning.
2. From 12:00 to 18:59 ➔ Evening.
3. Otherwise ➔ Night.

# 5. Generate New Columns

## The Result:

| time | year | month | day | month_name | day_name | week_label | day_parts |
|---|---|---|---|---|---|---|---|
| 17:59:02 | 2022 | 9 | 8 | September | Thursday | Weekday | Night |
| 18:00:28 | 2022 | 9 | 8 | September | Thursday | Weekday | Night |
| 18:06:05 | 2022 | 9 | 8 | September | Thursday | Weekday | Night |
| 18:05:46 | 2022 | 9 | 8 | September | Thursday | Weekday | Night |

# 5. Generate New Columns

## 5.7 Calculate the time duration of each session in hours, minutes and seconds.

- In Python, timedelta denotes a span of time. It’s the difference between two date, time, or datetime objects.

```
def time_duration_hours(end_time, start_time):
    from datetime import datetime
    # the string is changed to the DateTime object
    end_time = datetime.strptime(end_time, "%H:%M:%S")
    start_time = datetime.strptime(start_time, "%H:%M:%S")
    delta = end_time - start_time
    # get difference in seconds
    sec = delta.total_seconds()
    # get difference in min
    min = sec / 60
    # get difference in hours
    hours = sec / (60 * 60)
    # round the hours time
    return round(hours, 2)
```

# 5. Generate New Columns

## The Result:

| session_id | min_time | max_time | duration_hours | duration_minutes | duration_seconds |
|---|---|---|---|---|---|
| 1831e4150303a0-04a83b401e337e-26021c51-144000-1831e4150316d8 | 17:59:00 | 18:21:10 | 0.37 | 22.17 | 1330.0 |
| 1831fa67f29acf-03acc8ab0c94e8-26021c51-144000-1831fa67f2a10ea | 00:29:08 | 00:39:01 | 0.16 | 9.88 | 593.0 |
| 18327baf015d6c-01519b84418c49-26021c51-144000-18327baf016e38 | 14:08:26 | 14:56:09 | 0.80 | 47.72 | 2863.0 |
| 1832bff5050393-0f084138f6836c-26021c51-1fa400-1832bff50529e | 10:01:36 | 10:43:55 | 0.71 | 42.32 | 2539.0 |
| 1832ce7991833a-0fd268fb672d8b-1b525635-fa000-1832ce7991993a | 14:15:19 | 14:15:56 | 0.01 | 0.62 | 37.0 |

# *6. Handling Missing Values*

## 4.1 Handling Missing Values in Event Type column;

- There are 35K missing values out of 95K.
- Possibilities were: Click, Change and Submit
- Technically, If the user was just visiting the website without doing anything ➔ they did not do any action.

Filling the NaN's with "**No Action**"

# 6. Handling Missing Values

**The Result:**

| | |
|---|---|
| click | 49033 |
| no action | 35414 |
| change | 10990 |
| submit | 381 |

# *6. Handling Missing Values*

## 4.2 Handling Missing Values in the paths;

- There are some missing values in path 3, 4 and 5.

Filling the NaN's with "**Blank**"

```
df["path3"].fillna('blank', inplace=True)
df["path4"].fillna('blank', inplace=True)
df["path5"].fillna('blank', inplace=True)
```

# 6. Handling Missing Values

**The Result:**

| path1 | path2 | path3 | path4 | path5 |
|---|---|---|---|---|
| investor | dashboard | blank | blank | blank |
| investor | investment-portfolio | blank | blank | blank |
| investor | investment-portfolio | blank | blank | blank |

# 7. Labeling the Dataset

```
#Based on our research on the website and on the information given to us by the development
#team who developed the website, this is the path that leads to investment.
df_investor=df_concat.loc[(df_concat.path1 == "investor")&(df_concat.path2 == "transactions")
                          & (df_concat.path3 == "detail")]
```

```
#Here we found out that those who entered the transactions of investment page are diveded
#into two parts:
#788 of them "clicked" --> investors
#1550 did not do any action --> may invest "potential investors"
df_investor['event_type'].value_counts()
```

```
no action    1550
click         788
Name: event_type, dtype: int64
```

- **Investor condition:**

Users who entered the /investor/transactions/detail and they clicked.

path1 == "investor" **AND** path2 == "transactions" **AND**

path3 == "detail" **AND** event_type == '**click**'

# *7. Labeling the Dataset*

- **Potential Investor condition:**

1. Users who entered the /investor/transactions/detail and they did not do any action.

path1 == "investor" **AND** path2 == "transactions" **AND**

path3 == "detail" **AND** event_type == '**no action**'

2. Users who entered /investor/investment/..../form-step1 and they filled and submitted the investment form.

path1 == "investor" **AND** path2 == "investment" **AND**

path4 == "form-step1" **AND** event_type == '**submit**'

# 7. Labeling the Dataset

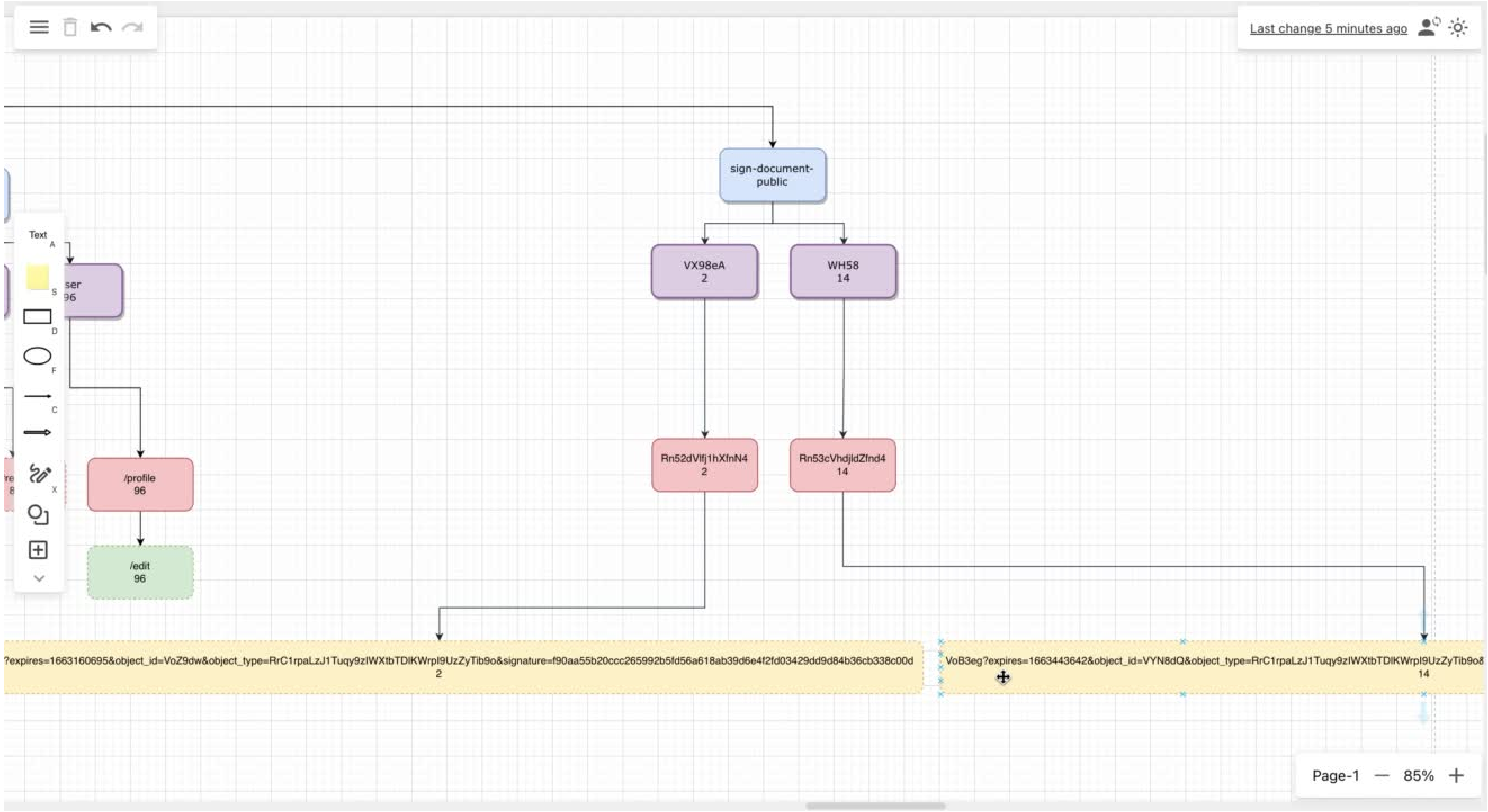
## The Result:

| duration_hours | duration_minutes | duration_seconds | invest |
| --- | --- | --- | --- |
| 0.37 | 22.17 | 1330.0 | No |
| 0.16 | 9.88 | 593.0 | No |
| 0.80 | 47.72 | 2863.0 | No |

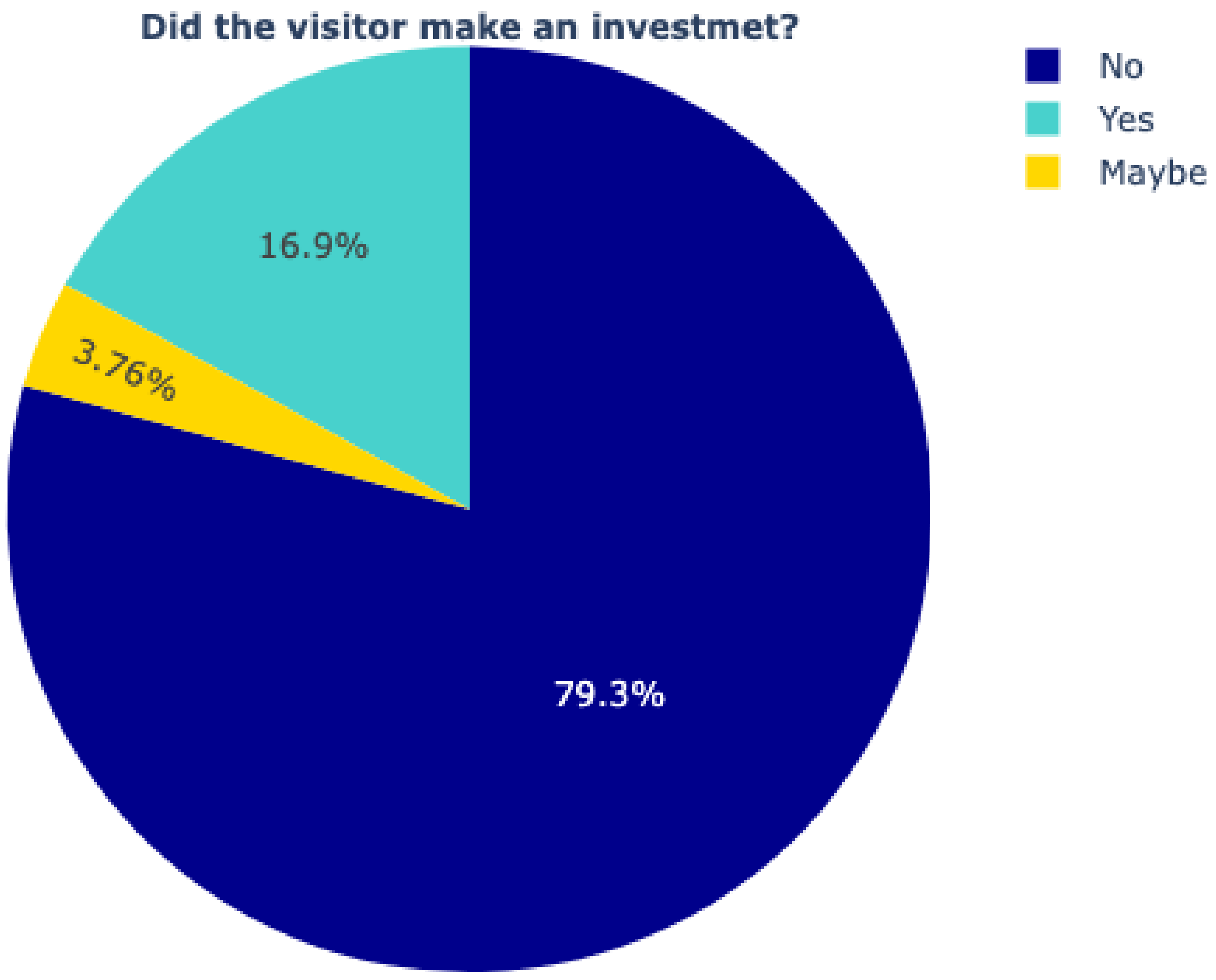
| | |
| --- | --- |
| No | 2339 |
| Yes | 499 |
| Maybe | 111 |

# Exploratory Data Analysis

# Paths visual tree



# What is the type of website visitation?

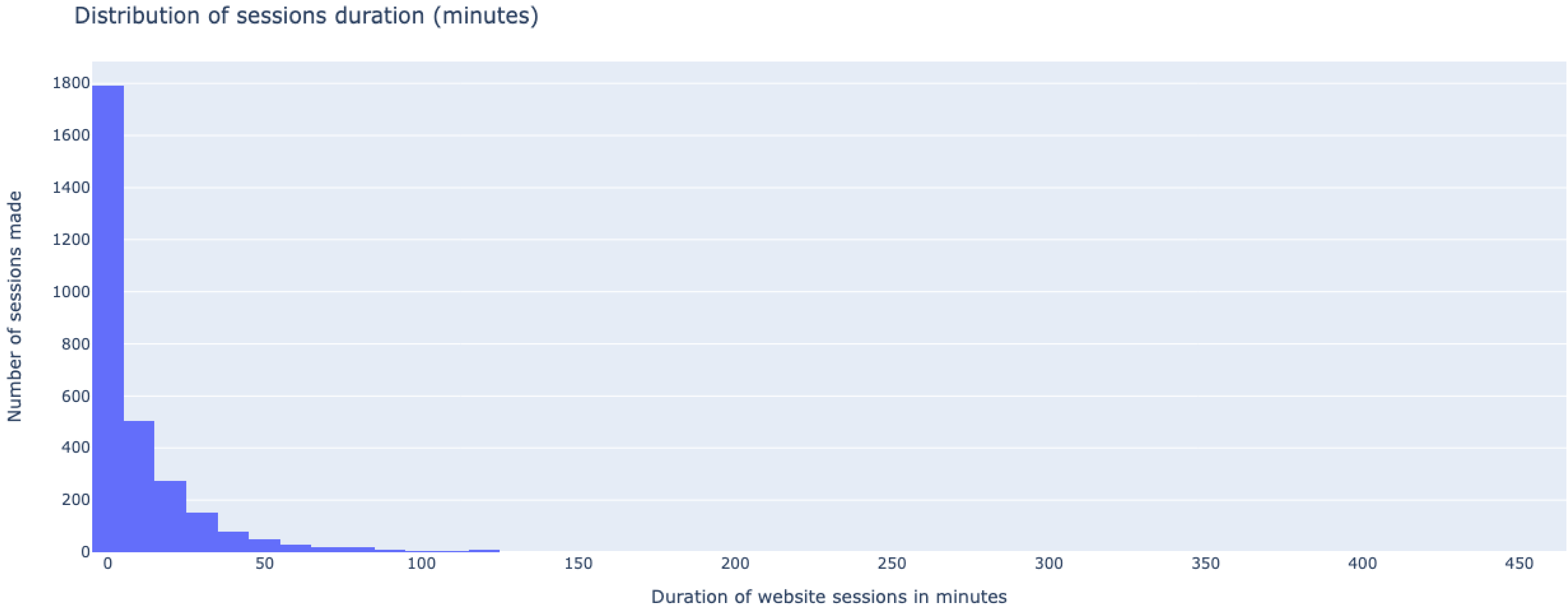



**Key insights:**

- **Majority of visitations** on the period on the analysis **did not make an investment**, where they represent **79%**
- **17% represents visitors who made a successful investment**
- While **4% represents visitors who tying to initiate an investment,** however they dropped out where we will call them the (Potential investors)

*Source: EDA notebook, 3.1 Visitation distribution*

# What was the duration of the visits?

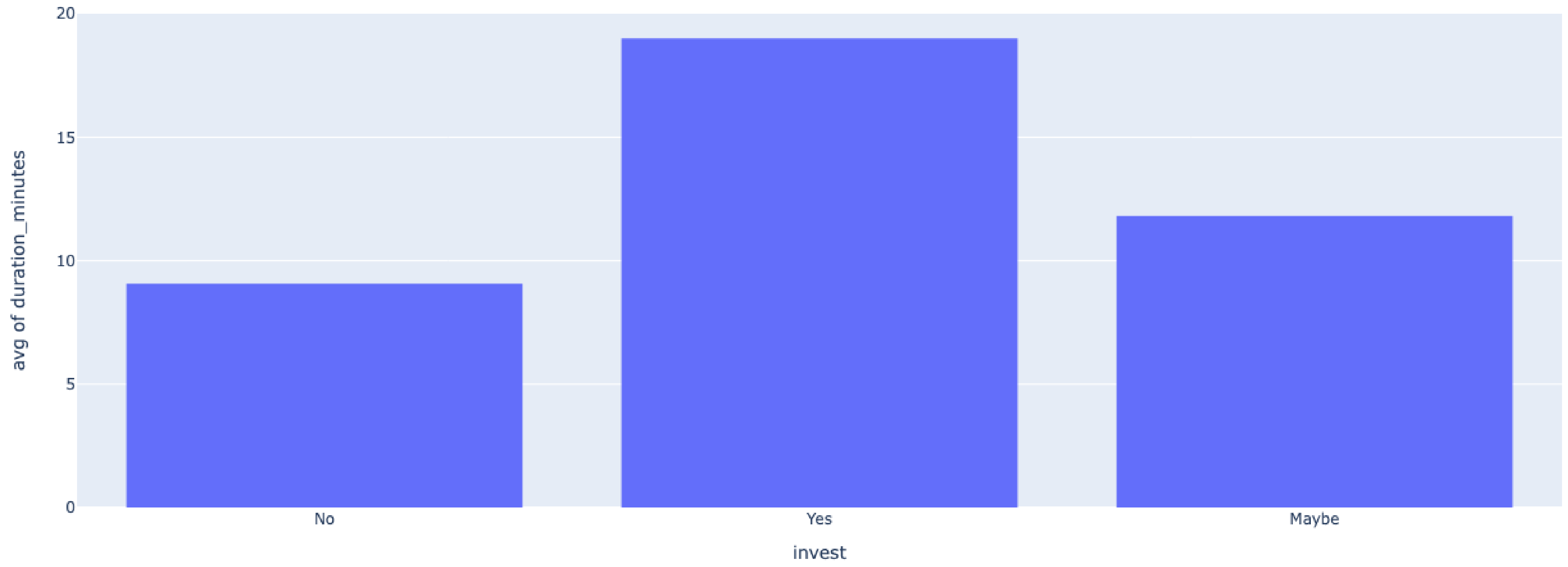



**Key insights:**

- **Majority of website visits remained for less than 5 minutes**

*Source: EDA notebook, 3.2 Sessions duration of visits (minutes)*

# In-depth view: Does investors stay less or more duration wise?




**Key insights:**

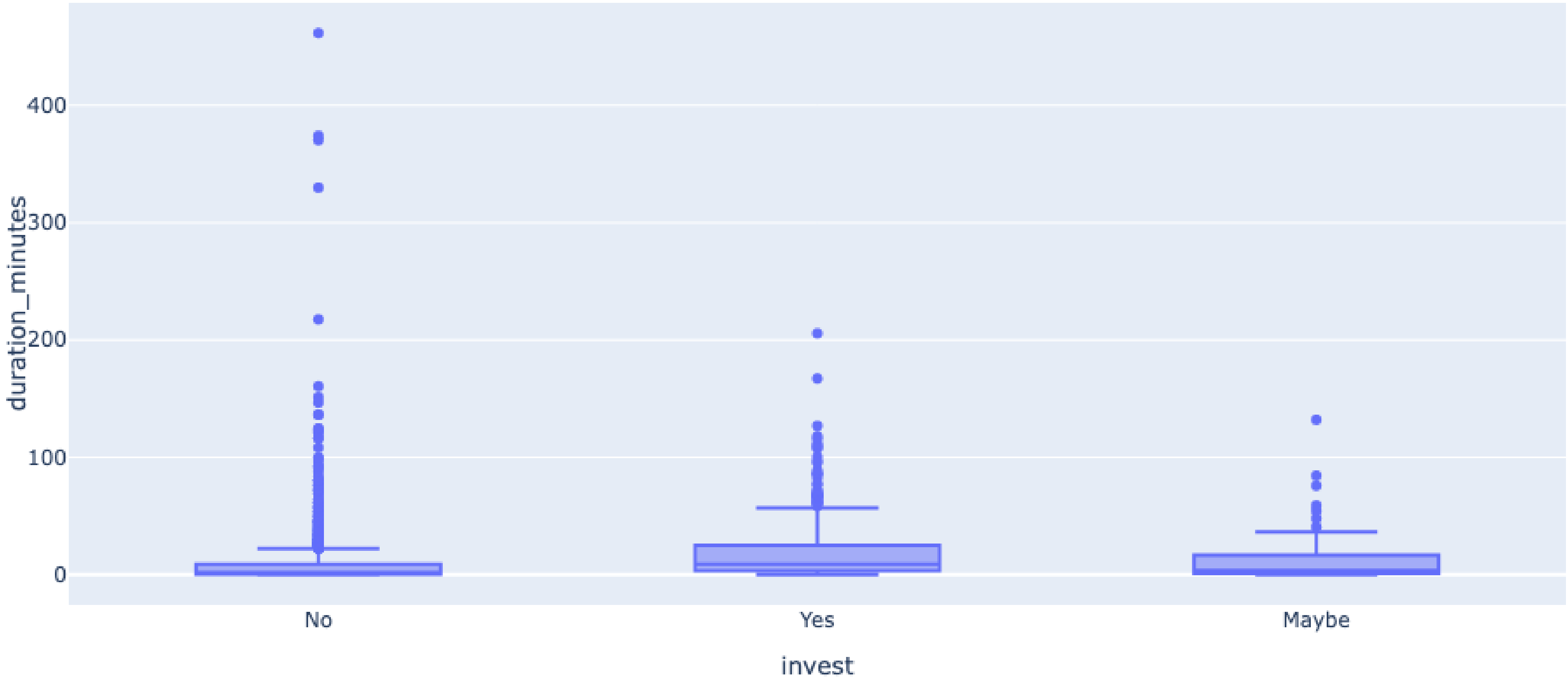
- **Investors have the highest average of** sessions duration staying on average **19 minutes**
- Followed by the **Potential investors visitors with ~ 11 minutes.**
- The **non-investors** visitors staying on the website for around **9 minutes on average.**

*Source: EDA notebook, 3.2 Sessions duration of visits (minutes)*

# In-depth view: Does investors stay less or more duration wise (excluding outliers)

FintechSaudi فنتك السعودية

Distribution based on sessions duration and investment




**Investors**

9 mins

**Potential investors**

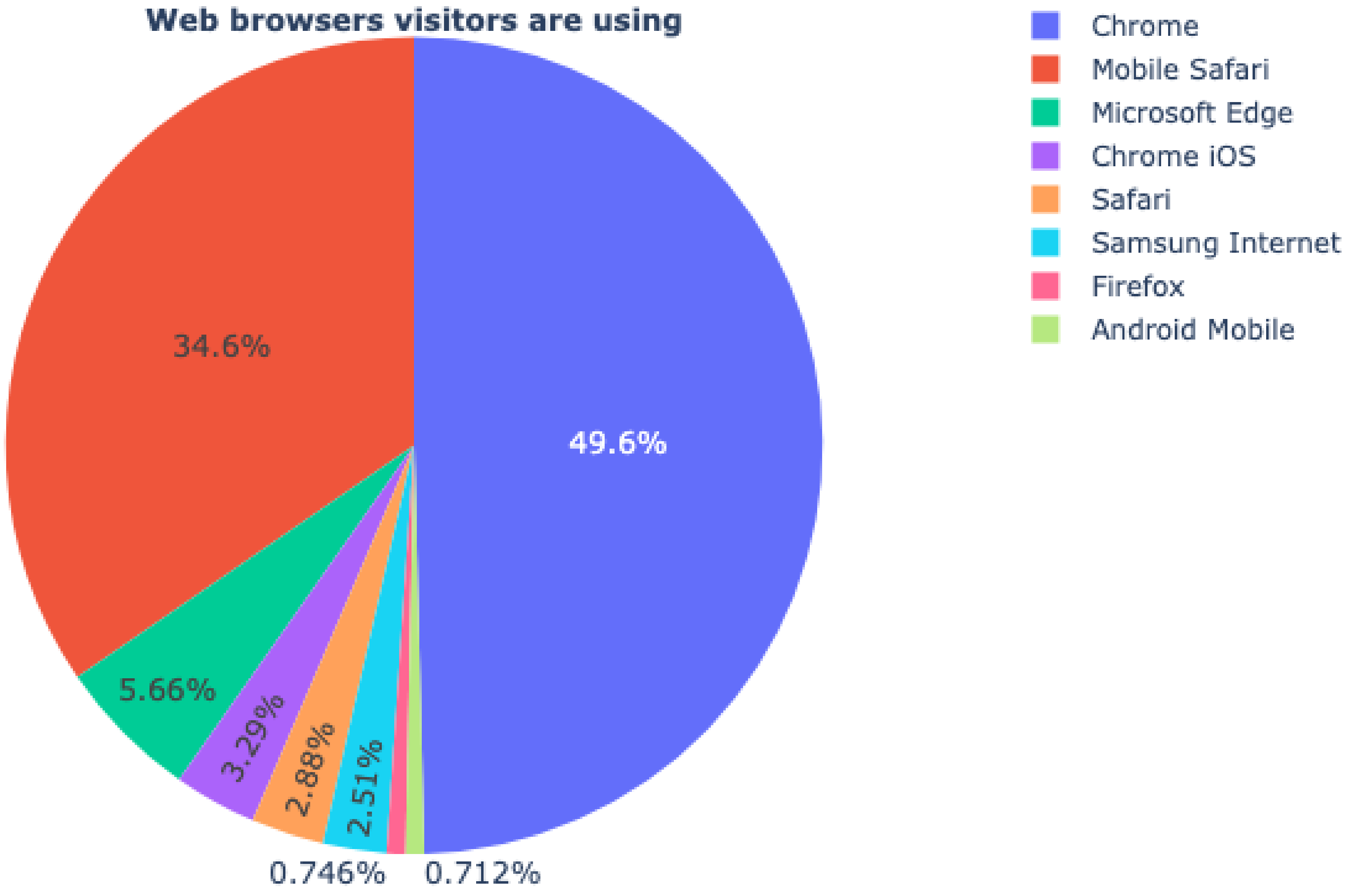
4 mins

**Non-Investors**

2 mins

*Source: EDA notebook, 3.2 Sessions duration of visits (minutes)*

# Browsers visitors are using through their website visitaion?



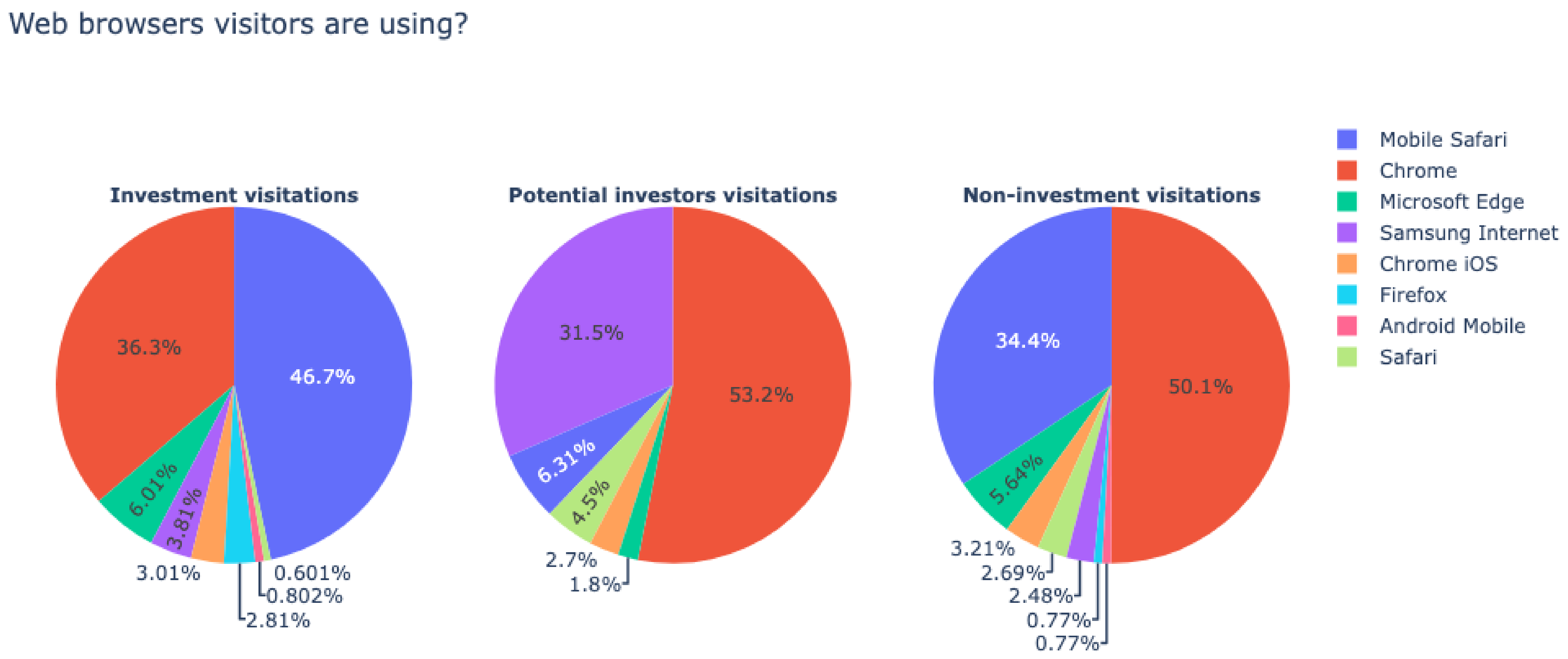

**Key insights:**

- **Half of the website visitations were using Chrome** as their web browser when browsing the website
- Followed by **Safari users counting for 34%** of the visitations

*Source: EDA notebook, 3.3*

# In-depth view: Browsers visitors are using through their website visitation?

فنتك السعودية
FintechSaudi




**Key insights:**

- **~47% from the investment visitations where using Mobile Safari**, followed by 36% using Chrome browser
- **More than half of the potential investors** visitations where **using Chrome** as their browser, followed by 32% Samsung internet browser users
- **Half of the non-investment visitations where using Chrome as their browser**, followed by Mobile Safari users counting for 35% of users

*Source: EDA notebook, 3.3*

# Devices users are using when visiting the website?

**74%**

Desktop

**26%**

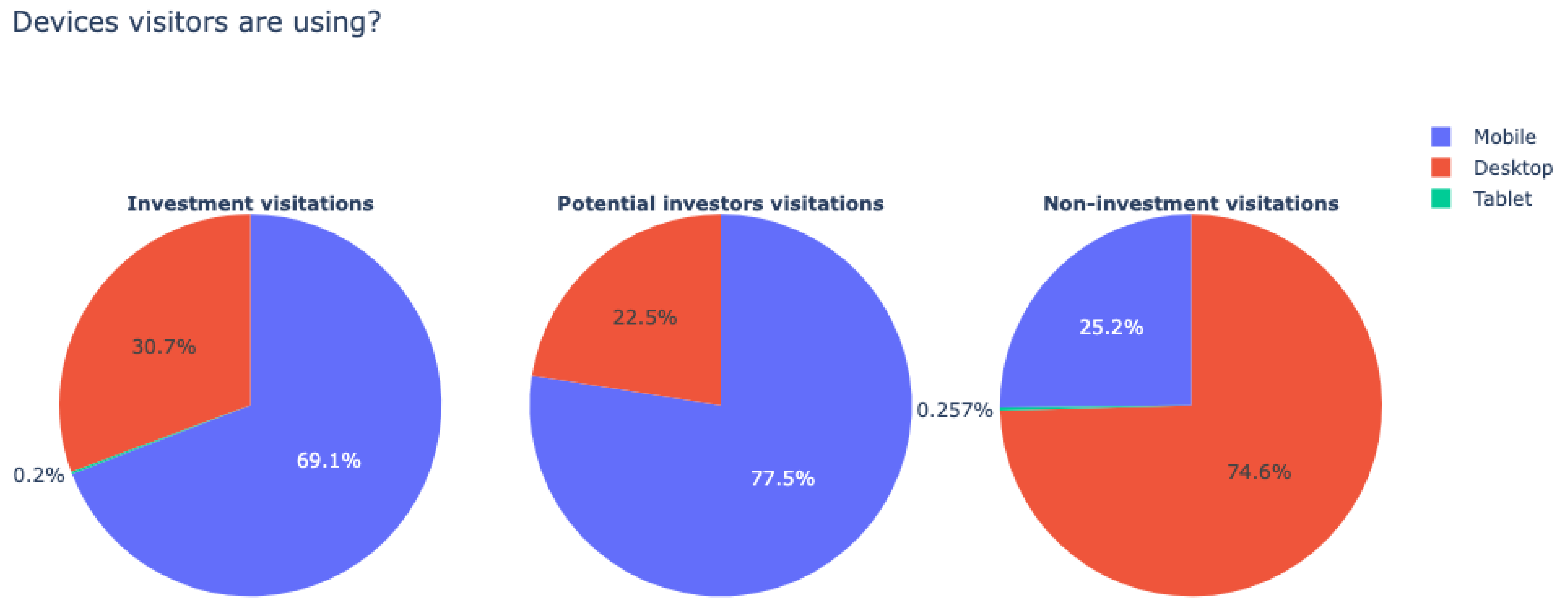
Moblie

**0.2%**

Tablet

*Source: EDA notebook, 3.4*

# In-depth view: Devices users are using when visiting the website?

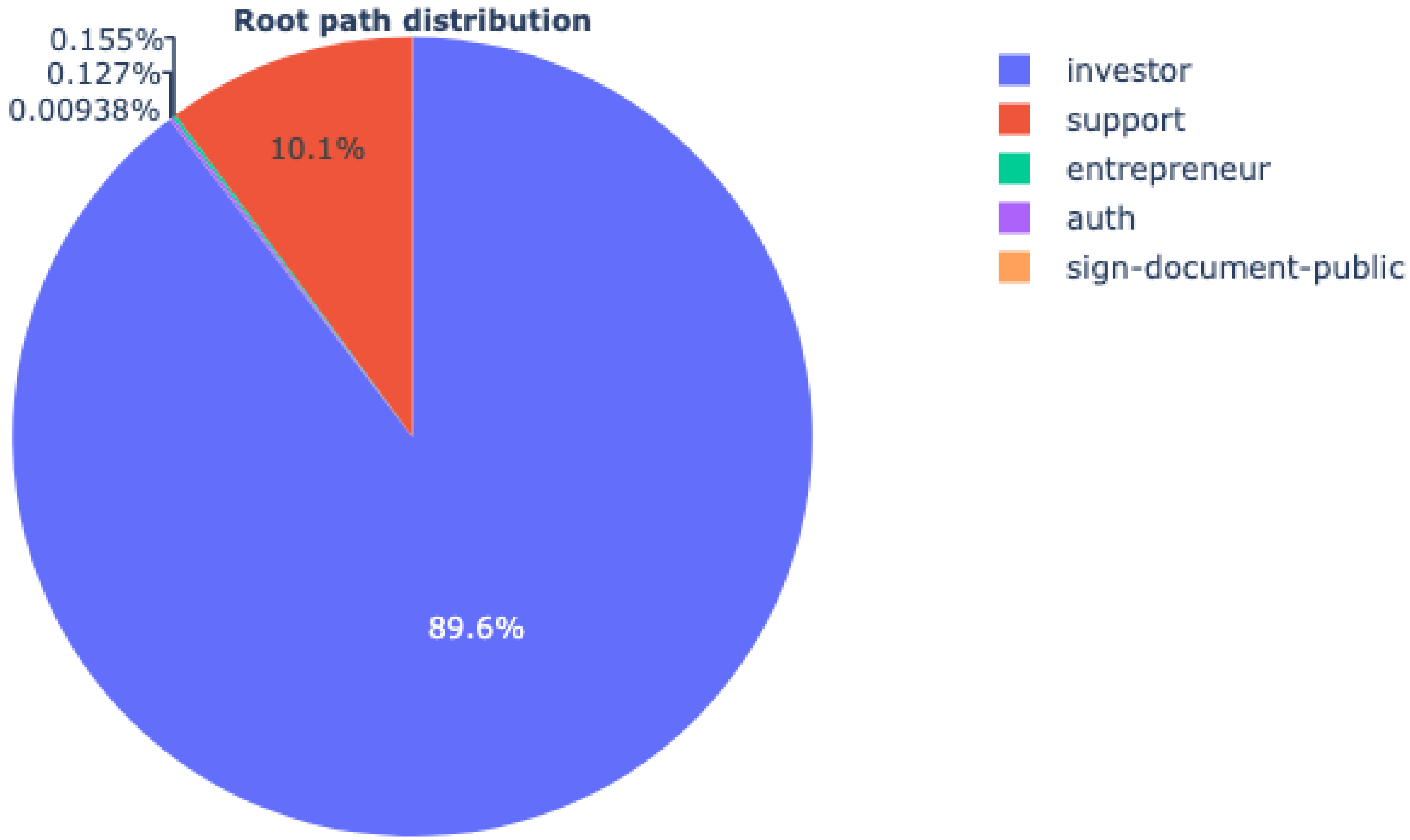



*Source: EDA notebook, 3.4*

**Key insights:**

- The **majority of Investors & potential investors visitations where using their Mobiles**
- 75% **of the Non-investment visitations** they where **using** their **Desktops**

# Root path distribution

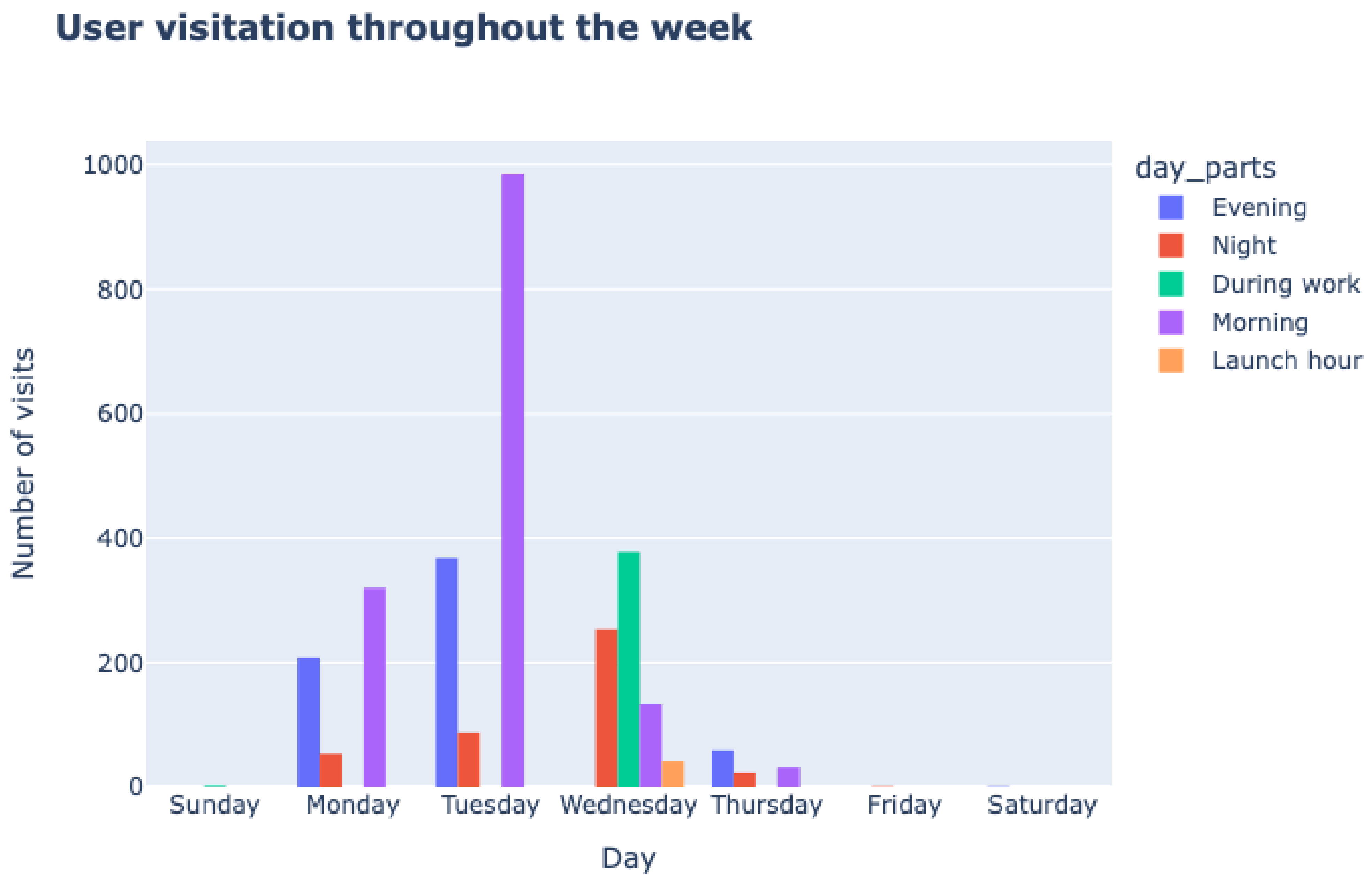



**Key insights:**

- **Majority of visits where at the investor web path 90%** (i.e., users looking at their investment profiles, wallets and investment opportunities)
- **Followed by the support web path at 10%** (i.e., where issues are raised, following up with previous issues raised)

*Source: EDA notebook, 3.8*

# Visitations around the week and by the part of the day




*Source: EDA notebook, 3.11*

**Key insights:**

- **Thursday morning had the highest visitation counting for around 1000 visits**
- **Weekends had the lowest visitation, nearly no visitations were counted**

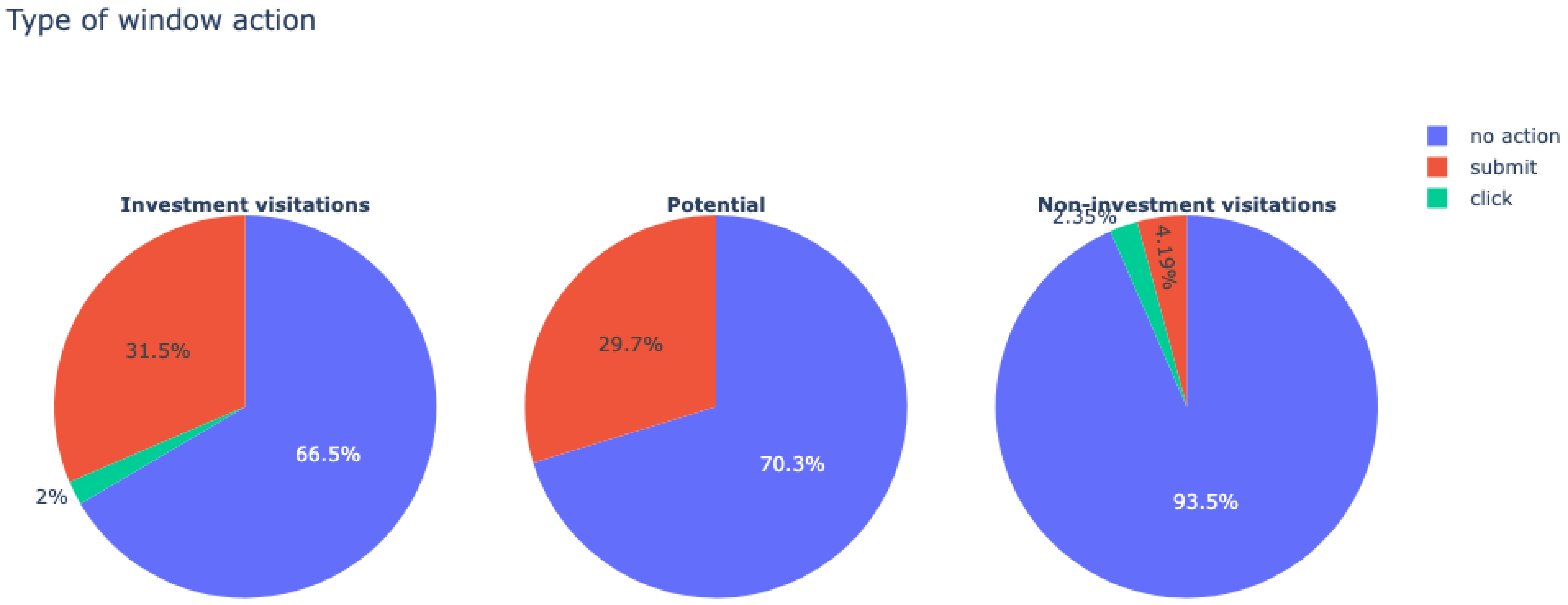
# Type of window action

فنتك السعودية
FintechSaudi

| 88% | 8% | 4% |
|---|---|---|
| No actions | Submit | Click |

*Source: EDA notebook, 3.12*

# In-depth view: Type of window action?




**Key insights:**

- Nearly **all of the non-investment visitations had no window action**.
- **Investment and potential investors** visitations had **submit window action as the majority**.

*Source: EDA notebook, 3.12*

# Average number of clicks by type of visit

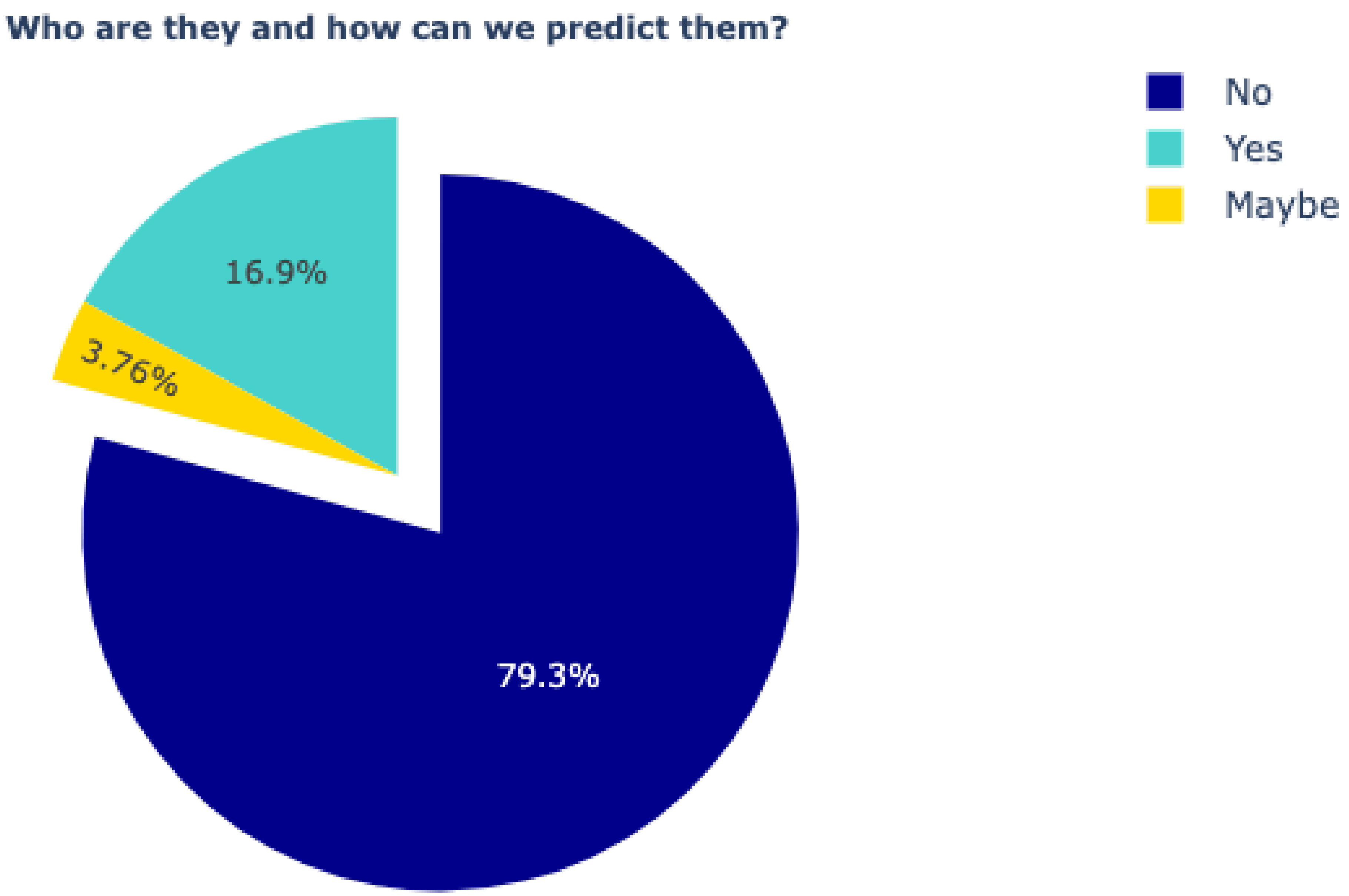
**Investors**
**80 clicks on average**

**Potential investors**
**47 clicks on average**

**Non-Investors**
**21 clicks on average**

*Source: EDA notebook, 3.13*

# Who are the investors & potential investors and how can we predict them?




*Source: EDA notebook, 3.14*

# Dashboards

فنتك السعودية
FintechSaudi

# Machine Learning Models

# Feature Engineering

فنتك السعودية
FintechSaudi

## Label Encoding

- week_label
- type
- browser
- device_type
- os
- day_parts
- event_type
- day_name
- path1
- path2
- path3

## Mapping

**Invest**

No → 0

Yes → 1

Maybe → 2

## Scaling

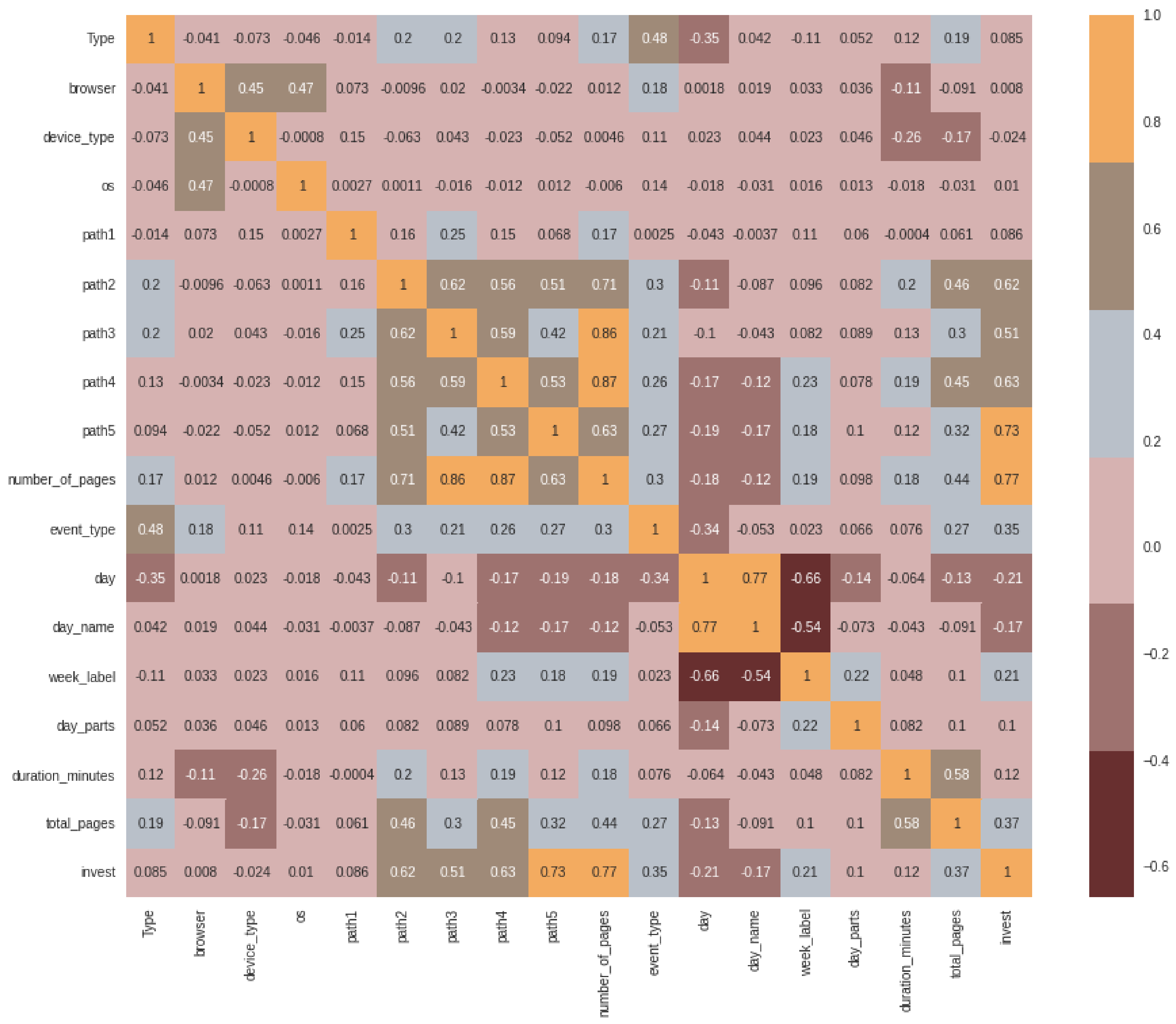
- duration_minutes
- total_pages
- day
- path2
- path3

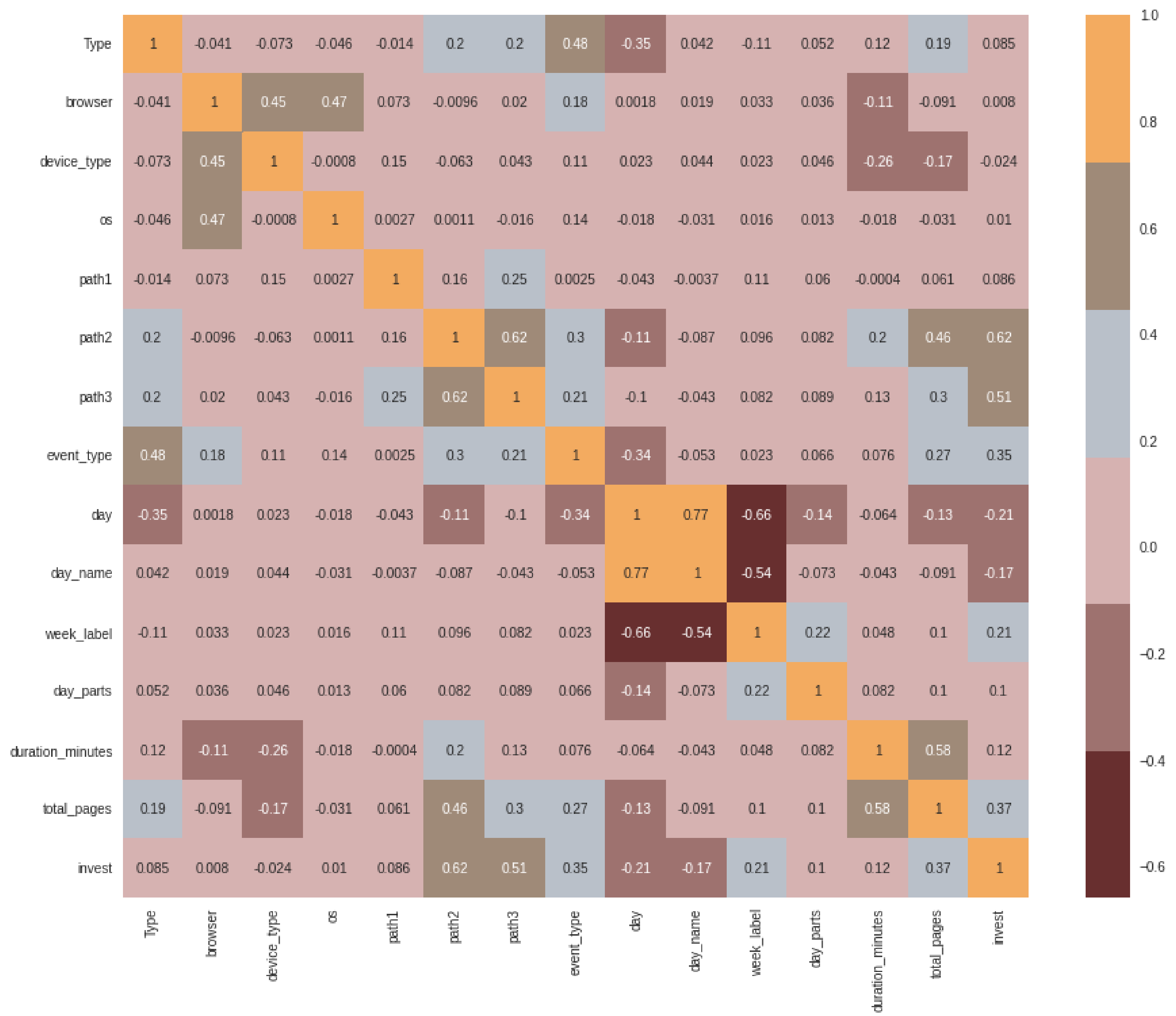
## Over-Sampling

| | Before | | After |
|---|---|---|---|
| 0 | 1892 | → | 1892 |
| 1 | 384 | → | 1892 |
| 2 | 83 | → | 1892 |

# Feature Selection

فنتك السعودية
FintechSaudi



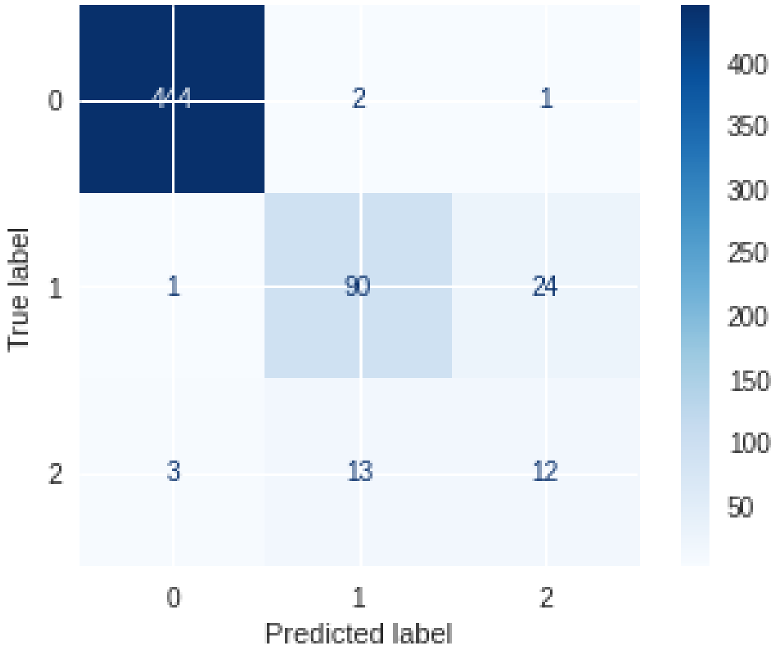
# ML Models

| | |
|---|---|
| 85% | Decision Trees |
| 86% | k-Nearest Neighbors |
| 90% | Support Vector |
| 92% | Gradient Boosting |
| 93% | Random Forest |
| 93% | XGBoost |

# Models Evaluation

فنتك السعودية
FintechSaudi

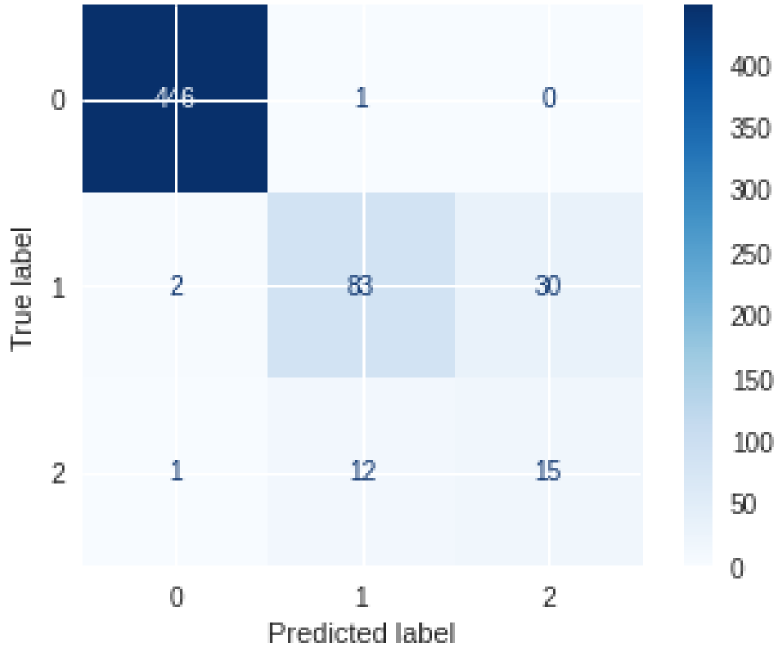
## Random Forest




Classification Report for Random Forest Classification model:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 0.99 | 0.99 | 447 |
| 1 | 0.86 | 0.78 | 0.82 | 115 |
| 2 | 0.32 | 0.43 | 0.37 | 28 |
| accuracy | | | 0.93 | 590 |
| macro avg | 0.72 | 0.73 | 0.73 | 590 |
| weighted avg | 0.93 | 0.93 | 0.93 | 590 |

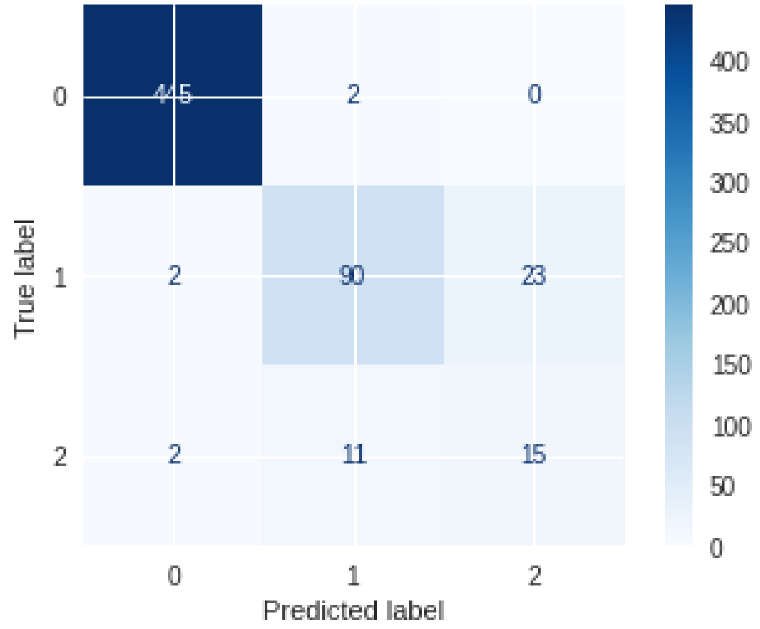
## Gradient Boosting




Classification Report for Gradient Boosting Classification model:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 1.00 | 1.00 | 447 |
| 1 | 0.86 | 0.72 | 0.79 | 115 |
| 2 | 0.33 | 0.54 | 0.41 | 28 |
| accuracy | | | 0.92 | 590 |
| macro avg | 0.73 | 0.75 | 0.73 | 590 |
| weighted avg | 0.94 | 0.92 | 0.93 | 590 |

## XGBoost




Classification Report for XGBoost Classification model:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 1.00 | 0.99 | 447 |
| 1 | 0.87 | 0.78 | 0.83 | 115 |
| 2 | 0.39 | 0.54 | 0.45 | 28 |
| accuracy | | | 0.93 | 590 |
| macro avg | 0.75 | 0.77 | 0.76 | 590 |
| weighted avg | 0.94 | 0.93 | 0.94 | 590 |

# Model Selection

- **XGBoost**

Classification Report for XGBoost Classification model:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 1.00 | 0.99 | 447 |
| 1 | 0.87 | 0.78 | 0.83 | 115 |
| 2 | 0.39 | 0.54 | 0.45 | 28 |
| accuracy | | | 0.93 | 590 |
| macro avg | 0.75 | 0.77 | 0.76 | 590 |
| weighted avg | 0.94 | 0.93 | 0.94 | 590 |

- **Baseline Model**

| | |
|---|---|
| No | 0.79315 |
| Yes | 0.16921 |
| Maybe | 0.03764 |

# Recommendations

**Potential investors' recommendation system**
Create an automated dashboard based on behavioral parameters, where potential investors are flagged, for marketing purposes to target the potential investors (e.g., Advertising via email, SMS, pop-up windows related to the undecided investment opportunities, with the investment important KPIs)

**User unique id**
Collect the user's unique id, as part of the user logs collected (i.e., user's frequency of visitation, user frequency of visitation for a specific investment opportunity)

**User's demographics**
Tie dataset with Google Analytics collected data and utilze it's benifits to provide better understanding of website users

**Attract new customers (Investors),**
Increase incoming new traffic, leaning from current investors characteristics to attract similar new investor

# *Future Work and Conclusion*

1. **Creating a package to handle the preprocessing of user activity logs datasets.**

2. **Create a more customizable model using the users ID and user demographics.**

# *Thank you*

For your attention

***Desert Ninjas***