WELLNESS TOURISM PACKAGE : Descriptive Analysis And New Segment Prediction

GROUP 4:

Anusha Ramesh

Prerana Das

Sanjana Santhanakrishnan

Sowmya Bhatraju

Swati Srivastava

AGENDA

- TEAM INTRODUCTION
- OBJECTIVE AND KEY TAKEAWAYS
- DATA ANALYSIS
  - DATA CLEANING
  - EXPLORATORY DESCRIPTIVE ANALYSIS OF USER-GROUPS
  - PRODUCT ADOPTION AND PREDICTION MODEL

MEET OUR TEAM!

Sanjana Santhanakrishnan

Prerana Das

Sowmya Bhatraju

Swati Srivastava

Anusha Ramesh

AGENDA

- TEAM INTRODUCTION
- OBJECTIVE AND KEY TAKEAWAYS
- DATA ANALYSIS
  - DATA CLEANING
  - EXPLORATORY DESCRIPTIVE ANALYSIS OF USER-GROUPS
  - PRODUCT ADOPTION AND PREDICTION MODEL

CASE OVERVIEW AND OBJECTIVE

PROBLEM STATEMENT : To analyze customers' data for "Trips & Travel.Com" company to provide consulting on segmentation and bolster strategic marketing for a new travel package, "Wellness Tourism", through a viable business model.

OBJECTIVE

- Determine the factors that drive the current customers' decision making
- Build an ML model to predict potential customers for the new product

DATASET

It consists of **4888** records and provides information of the customers, such as demographic information, type of contact, product pitched, duration of pitch, no of persons visited etc.

APPROACH

- Conduct descriptive analysis of the customer data and the variation in the variables based on the tourism package segments
- Use this analysis to understand the factors that drive preference for new product

THERE ARE 3 KEY TAKEAWAYS FROM THIS PROJECT

The **5 product packages**– Basic, Standard, Deluxe, Super Deluxe and King – chosen by customer groups who are **different in their product and pitch preference**, and demographics. Their preference is **<u>driven by their designation and income.</u>**

The **current product buyers** are the ones who belong to the following category:

- **age group of 15-30**
- **single/unmarried males**,
- more willing to **adopt a product based off sales pitch**,
- are contacted by company,
- belong to **tier 2 and 3 cities.**
- working as **Executive with a $15-30K salary**
- prefer **5-star hotels.**

Along with sophisticated techniques such as **random forest and XGBoost, basic exploratory analysis** also provides the most appropriate picture of real driving factors of a data. We created a **prediction model with 10 independent variables** with an accuracy of ~**90%**

AGENDA

- TEAM INTRODUCTION
- OBJECTIVE AND KEY TAKEAWAYS
- DATA ANALYSIS
  - DATA CLEANING
  - EXPLORATORY DESCRIPTIVE ANALYSIS OF USER-GROUPS
  - PRODUCT ADOPTION AND PREDICTION MODEL

OUR DATASET HAS 4888 ROWS AND 20 COLUMNS, IS FROM KAGGLE

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib
        import matplotlib.pyplot as plt
        import seaborn as sns

In [2]: import os
        path = input("enter directory")
        os.chdir(path)
        data = pd.read_csv("Travel.csv")

        enter directoryC:\Users\Sanjana\Desktop\Python Project

In [3]: data.head()
```

	CustomerID	ProdTaken	Age	TypeofContact	CityTier	DurationOfPitch	Occupation	Gender	NumberOfPersonVisiting	NumberOfFollowups	ProductPitched	PreferredPropertyStar	MaritalStatus	NumberOfTrips	Passport	PitchSatisfactionScore	OwnCar	NumberOfChildrenVisiting	Designation	MonthlyIncome
0	200000	1	41.0	Self Enquiry	3	6.0	Salaried	Female	3	3.0	Deluxe	3.0	Single	1.0	1	2	1	0.0	Manager	20993.0
1	200001	0	49.0	Company Invited	1	14.0	Salaried	Male	3	4.0	Deluxe	4.0	Divorced	2.0	0	3	1	2.0	Manager	20130.0
2	200002	1	37.0	Self Enquiry	1	8.0	Free Lancer	Male	3	4.0	Basic	3.0	Single	7.0	1	3	0	0.0	Executive	17090.0
3	200003	0	33.0	Company Invited	1	9.0	Salaried	Female	2	3.0	Basic	3.0	Divorced	2.0	1	5	1	1.0	Executive	17909.0
4	200004	0	NaN	Self Enquiry	1	8.0	Small Business	Male	2	3.0	Basic	4.0	Divorced	1.0	0	5	1	0.0	Executive	18468.0

BUSINESS QUESTION SOURCE – KAGGLE

DATASET TYPE – CSV

NO. OF ROWS – 4888
NO. OF COLUMNS – 20

LIBRARIES – PANDAS, SEABORN, MATPLOTLIB

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4888 entries, 0 to 4887
Data columns (total 20 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   CustomerID                4888 non-null   int64
 1   ProdTaken                 4888 non-null   int64
 2   Age                       4662 non-null   float64
 3   TypeofContact             4863 non-null   object
 4   CityTier                  4888 non-null   int64
 5   DurationOfPitch           4637 non-null   float64
 6   Occupation                4888 non-null   object
 7   Gender                    4888 non-null   object
 8   NumberOfPersonVisiting    4888 non-null   int64
 9   NumberOfFollowups         4843 non-null   float64
 10  ProductPitched            4888 non-null   object
 11  PreferredPropertyStar     4862 non-null   float64
 12  MaritalStatus             4888 non-null   object
 13  NumberOfTrips             4748 non-null   float64
 14  Passport                  4888 non-null   int64
 15  PitchSatisfactionScore    4888 non-null   int64
 16  OwnCar                    4888 non-null   int64
 17  NumberOfChildrenVisiting  4822 non-null   float64
 18  Designation               4888 non-null   object
 19  MonthlyIncome             4655 non-null   float64
dtypes: float64(7), int64(7), object(6)
memory usage: 763.9+ KB
```

TO BOOST THE MODEL'S DATA QUALITY, WE REPLACED THE BLANKS WITH MEAN AND MODE BASED OFF THE SEGMENTS AS PER THE DATA TYPE

DATA CLEANING ISSUE	DATA CLEANING SOLUTION
Upon checking the null values, we found that 8 variables have blank rows for some respondents	For data imputation, we reviewed the data type for each of these variables. As per missing data imputation rules, we imputed the – • Categorical Variables with their most likely value of the segment • Continuous Variables with the segment mean

```
In [7]: data.isna().sum()

Out[7]: CustomerID                  0
        ProdTaken                   0
        Age                       226
        TypeofContact              25
        CityTier                    0
        DurationOfPitch           251
        Occupation                  0
        Gender                      0
        NumberOfPersonVisiting      0
        NumberOfFollowups          45
        ProductPitched              0
        PreferredPropertyStar      26
        MaritalStatus               0
        NumberOfTrips             140
        Passport                    0
        PitchSatisfactionScore      0
        OwnCar                      0
        NumberOfChildrenVisiting   66
        Designation                 0
        MonthlyIncome             233
        dtype: int64
```

```
for prod_type in df['ProductPitched'].unique():
    df.loc[((df['ProductPitched']==prod_type)&(df['Age'].isna())), 'Age'] =
    int(df[df['ProductPitched']==prod_type]['Age'].mean())

    df.loc[((df['ProductPitched']==prod_type)&(df['DurationOfPitch'].isna())), 'DurationOfPitch'] =
    df[df['ProductPitched']==prod_type]['DurationOfPitch'].mean()

    df.loc[((df['ProductPitched']==prod_type)&(df['NumberOfTrips'].isna())), 'NumberOfTrips'] =
    int(df[df['ProductPitched']==prod_type]['NumberOfTrips'].mean())

    df.loc[((df['ProductPitched']==prod_type)&(df['MonthlyIncome'].isna())), 'MonthlyIncome'] =
    df[df['ProductPitched']==prod_type]['MonthlyIncome'].mean()

    df.loc[((df['ProductPitched']==prod_type)&(df['NumberOfFollowups'].isna())), 'NumberOfFollowups'] =
    int(df[df['ProductPitched']==prod_type]['NumberOfFollowups'].mode()[0])

    df.loc[((df['ProductPitched']==prod_type)&(df['PreferredPropertyStar'].isna())), 'PreferredPropertyStar'] =
    int(df[df['ProductPitched']==prod_type]['PreferredPropertyStar'].mode()[0])

    df.loc[((df['ProductPitched']==prod_type)&(df['NumberOfChildrenVisiting'].isna())), 'NumberOfChildrenVisiting'] =
    int(df[df['ProductPitched']==prod_type]['NumberOfChildrenVisiting'].mode()[0])

df['TypeofContact'].fillna("NA", inplace=True)
```

LOOKING CLOSELY TO THE DATA, WE FURTHER CREATED NEWER VARIABLES FOR BETTER ANALYSIS RESULTS

DATA ISSUES	DATA MANIPULATION SOLUTION (CODE)
1. One value of the variable Gender was mis-spelled as "Fe Male"	1. Updated Fe Male to Female `df['Gender'].replace('Fe Male', 'Female', inplace=True)`

2. 3 Variables with 0-1 categories needed to work as categorical variables	2. Updated 0-1 to yes-no categories
	`df["ProductTaken"] = df['ProdTaken'].replace({1:"Yes", 0:"No"})`
	`df["CarOwned"] = df['OwnCar'].replace({1:"Yes", 0:"No"})`
	`df["HavePassport"] = df['Passport'].replace({1:"Yes", 0:"No"})`

3. Not much differentiation was observed in the data of age, income and pitch satisfaction scores	3. Created buckets for age, income and pitch satisfaction scores to find better differentiation among segments (code below)

```
age_groups = groupings.groupby(["ProductTaken", "AgeGroup"])["AgeGroup"].count().unstack().fillna(0)
age_groups = age_groups.div(age_groups.sum(axis=1), axis=0)*100
age_groups.plot(kind='bar', stacked=True, figsize=(9,7),width=0.24,
                color=['turquoise','grey','lightblue','darkblue','silver'])
plt.title('Age Groups vs Product Taken or Not', fontsize=13)
plt.xlabel('Product Taken or Not', fontsize=12)
plt.ylabel('Percentage of Age Groups', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(age_groups.columns, fontsize=12)
```

THE DATA ALSO UNDERWENT OUTLIER TREATMENT. WE USED INTERQUARTILE METHOD TO TREAT THE OUTLIERS

DATA MANIPULATION ISSUE	DATA MANIPULATION SOLUTION (CODE)
4. We found Outliers in three continuous variables that were misleading the range and mean values – i. Monthly Income ii. Duration of pitch iii. Number of trips	4. Used interquartile outlier treatment method for replacing the outlier data with [Q3 + 1.5*(Q3-Q1)]

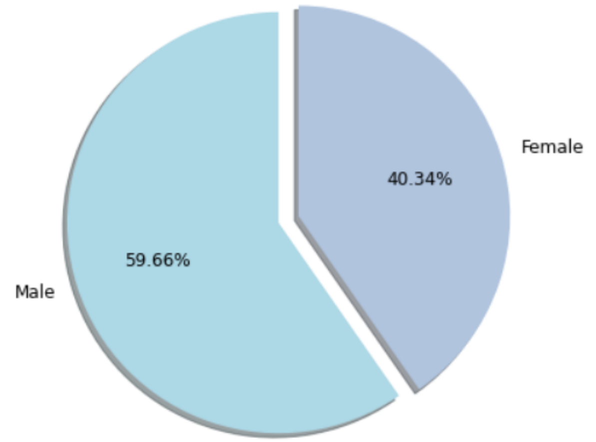
```python
for prod_type in df['ProductPitched'].unique():

    income_q3 = np.percentile(df.loc[df['ProductPitched']==prod_type, "MonthlyIncome"],
                              75, interpolation = 'midpoint')
    duration_q3 = np.percentile(df.loc[df['ProductPitched']==prod_type, "DurationOfPitch"],
                                75, interpolation = 'midpoint')
    numberoftrips_q3 = np.percentile(df.loc[df['ProductPitched']==prod_type, "NumberOfTrips"],
                                     75, interpolation = 'midpoint')

    iqr_income = stats.iqr(df.loc[df['ProductPitched']==prod_type, "MonthlyIncome"],
                           interpolation = 'midpoint')
    iqr_durationofpitch = stats.iqr(df.loc[df['ProductPitched']==prod_type, "DurationOfPitch"],
                                    interpolation = 'midpoint')
    iqr_numberoftrips = stats.iqr(df.loc[df['ProductPitched']==prod_type, "NumberOfTrips"],
                                  interpolation = 'midpoint')

    df.loc[((df['ProductPitched']==prod_type) & (df['MonthlyIncome']>(income_q3+(1.5*iqr_income)))),
           "MonthlyIncome"] = income_q3+(1.5*iqr_income)
    df.loc[((df['ProductPitched']==prod_type) & (df['DurationOfPitch']>(duration_q3+(1.5*iqr_durationofpitch)))),
           "DurationOfPitch"] = duration_q3+(1.5*iqr_durationofpitch)
    df.loc[((df['ProductPitched']==prod_type) & (df['NumberOfTrips']>(numberoftrips_q3+(1.5*iqr_numberoftrips)))),
           "NumberOfTrips"] = numberoftrips_q3+(1.5*iqr_numberoftrips)
```

AGENDA

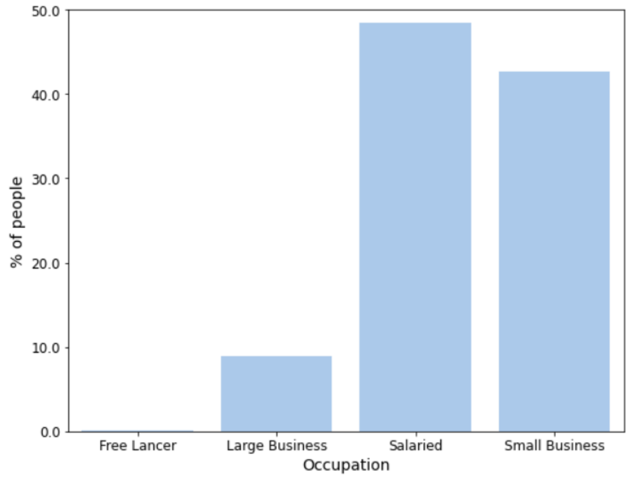
- TEAM INTRODUCTION
- OBJECTIVE AND KEY TAKEAWAYS
- DATA ANALYSIS
  - DATA CLEANING
  - EXPLORATORY DESCRIPTIVE ANALYSIS OF USER-GROUPS
  - PRODUCT ADOPTION AND PREDICTION MODEL

DEMOGRAPHIC DATA SHOWS THAT THE POPULATION HAS HIGHER % OF TIER 1, SALARIED, MARRIED, MALE; MORE LIKELY TO HAVE CAR, NOT PASSPORT

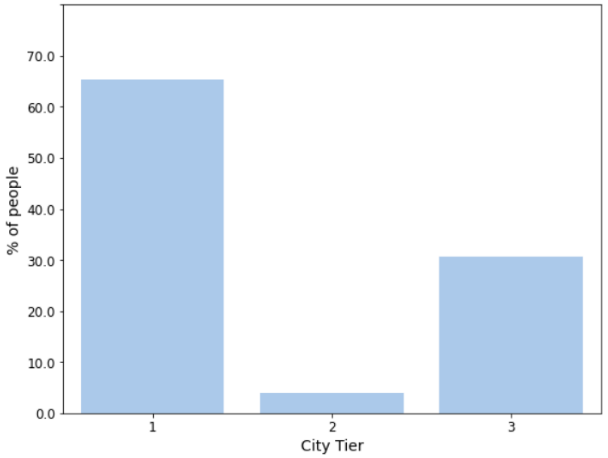
GENDER



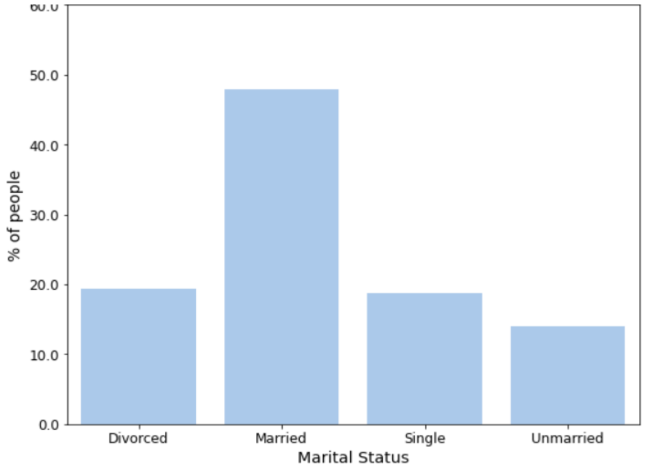
OCCUPATION



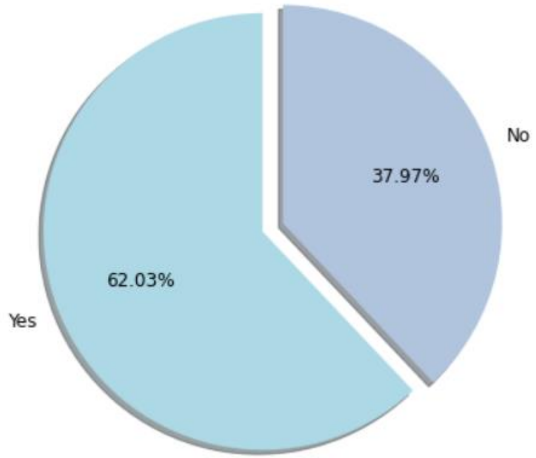
CITY TIER



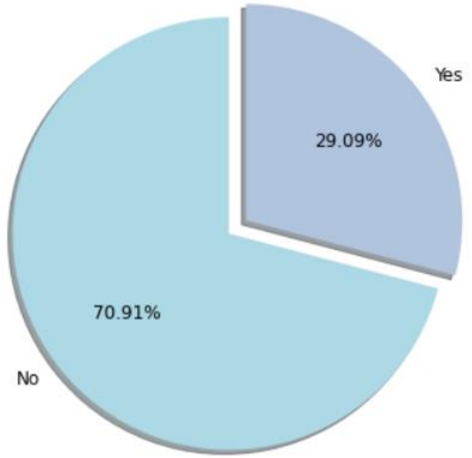
MARITAL STATUS



CAR OWNERSHIP

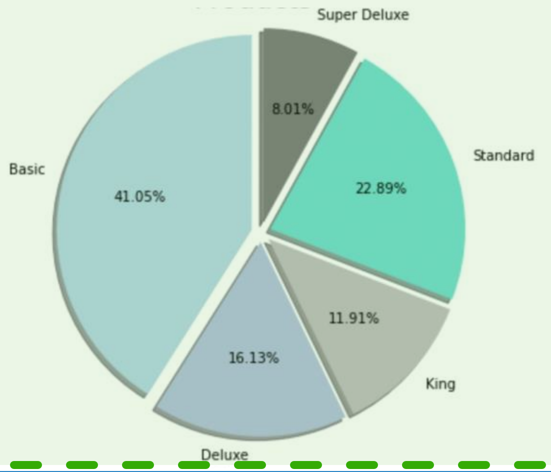


PASSPORT STATUS

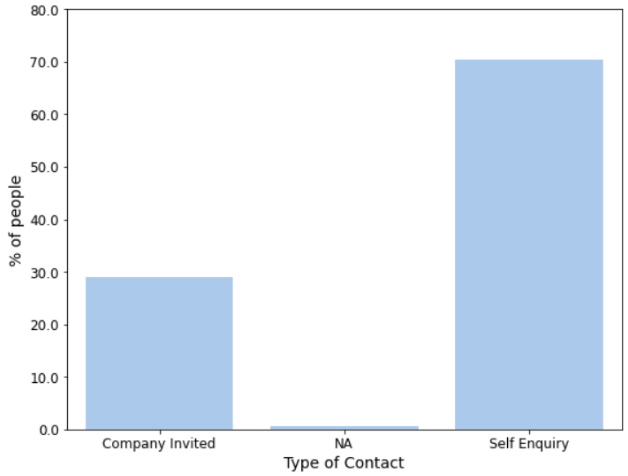


PRODUCT PREFERENCE DATA SHOWS HIGHER % OF BASIC USERS, WHO SELF-ENQUIRED, PREFER 3 STARS, HAVING 3 TRIPS

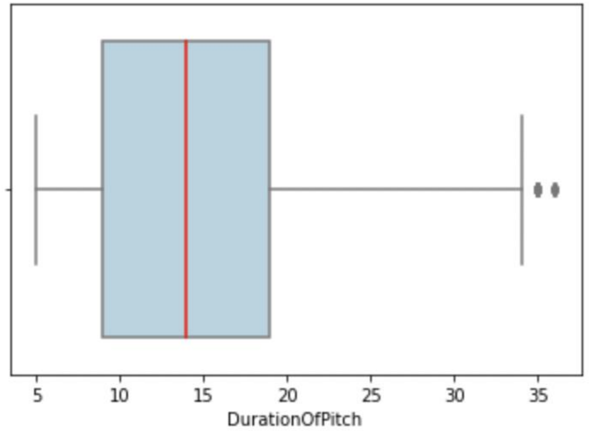
% PRODUCT USERS



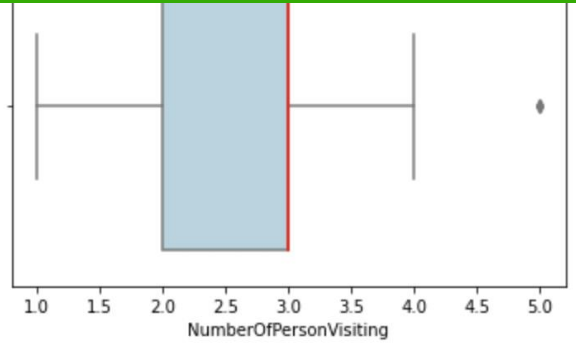
TYPE OF CONTACT



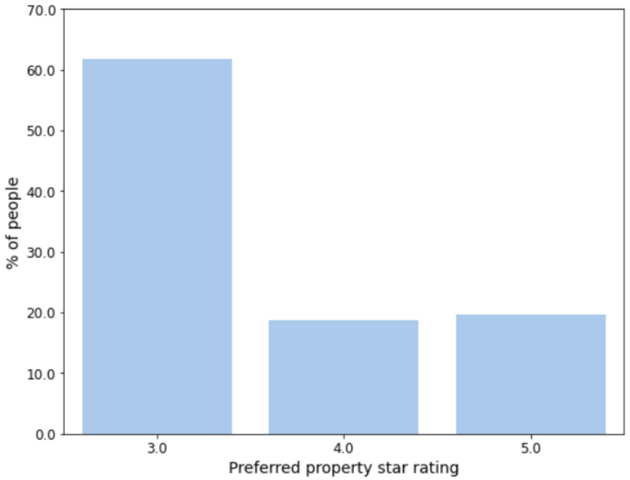
DURATION OF PITCH



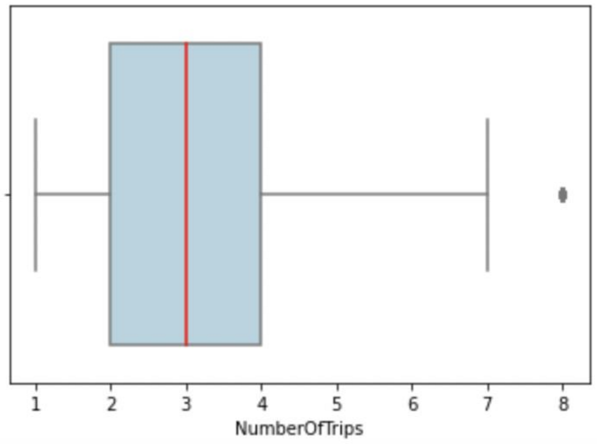
We will explore these segments in detail



PREFERRED PROPERTY STAR RATING

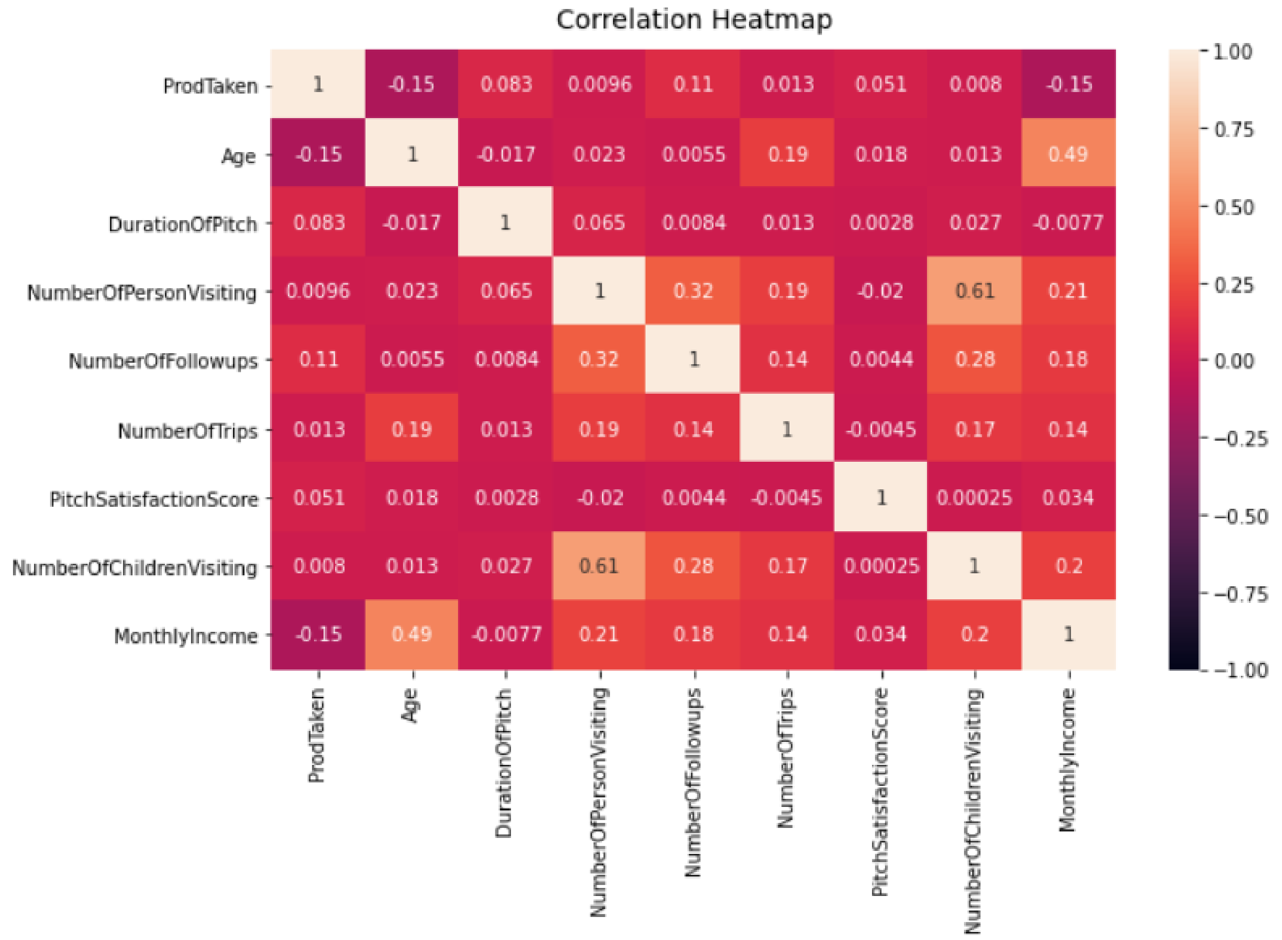


NUMBER OF TRIPS



WHAT KIND OF CORRELATION TRENDS ARE FOUND IN THE DATA?

```python
plt.figure(figsize=(10, 6))
heatmap = sns.heatmap(df.corr(), vmin=-1, vmax=1, annot=True)
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':14}, pad=12);
```



INSIGHTS:

We can see some obvious trends in the correlation analysis that is proving the good quality of the data –

- Number of persons visiting and number of children visiting = 0.61
- Age and Monthly Income = 0.49

BEFORE MOVING AHEAD, LET'S DISCUSS THE BUSINESS QUESTIONS WE WILL ADDRESS TODAY!

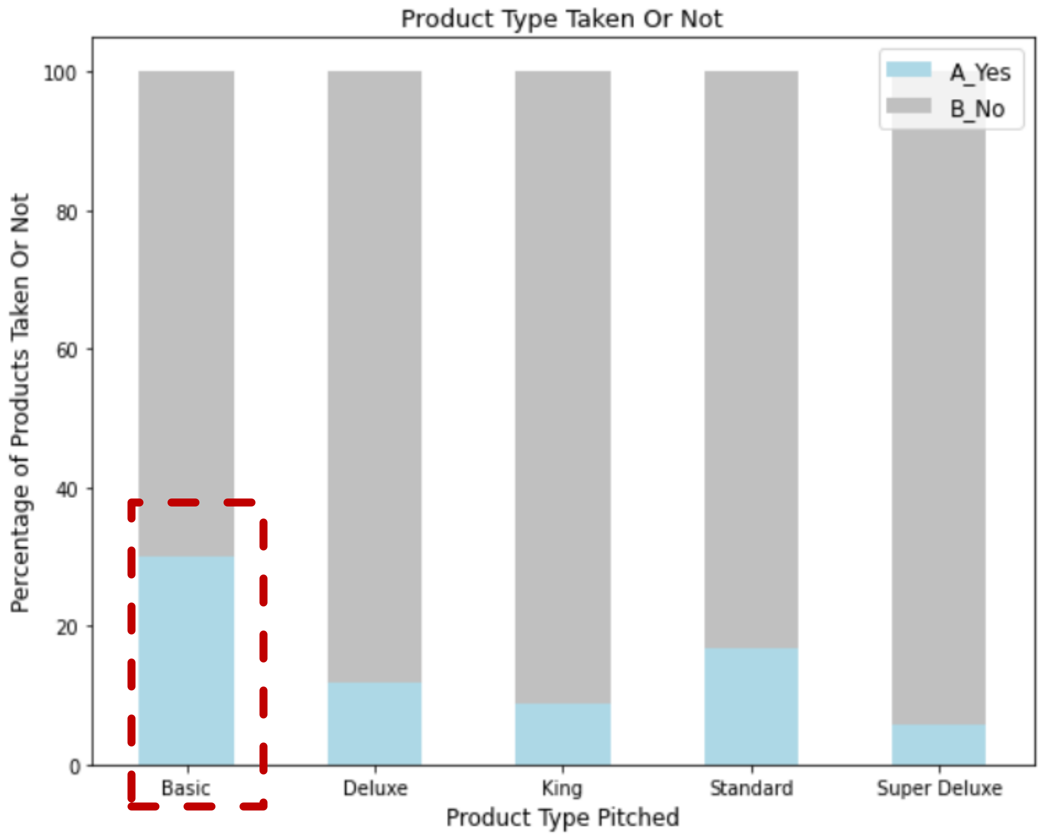
BUSINESS QUESTIONS:

1. What are the trends observed in product preference and demographics within the 5-user segments?
2. What is the story of the users a.k.a. product buyers who are willing to take a product based on product pitched?
3. What are the top 10 predicting variables or factors driving the willingness to take a new product?
4. What kind of model can we create to measure adoption of a new product? What will be its accuracy?

As business consultants, the aim of our study is to understand detailed insights of these business questions and present the solution to the client "Trips and Travels .com"

1. What are the trends observed in product preference and demographics within the 5-user segments?

WHICH SEGMENT IS MOST WILLING TO TAKE THE NEW PRODUCT?



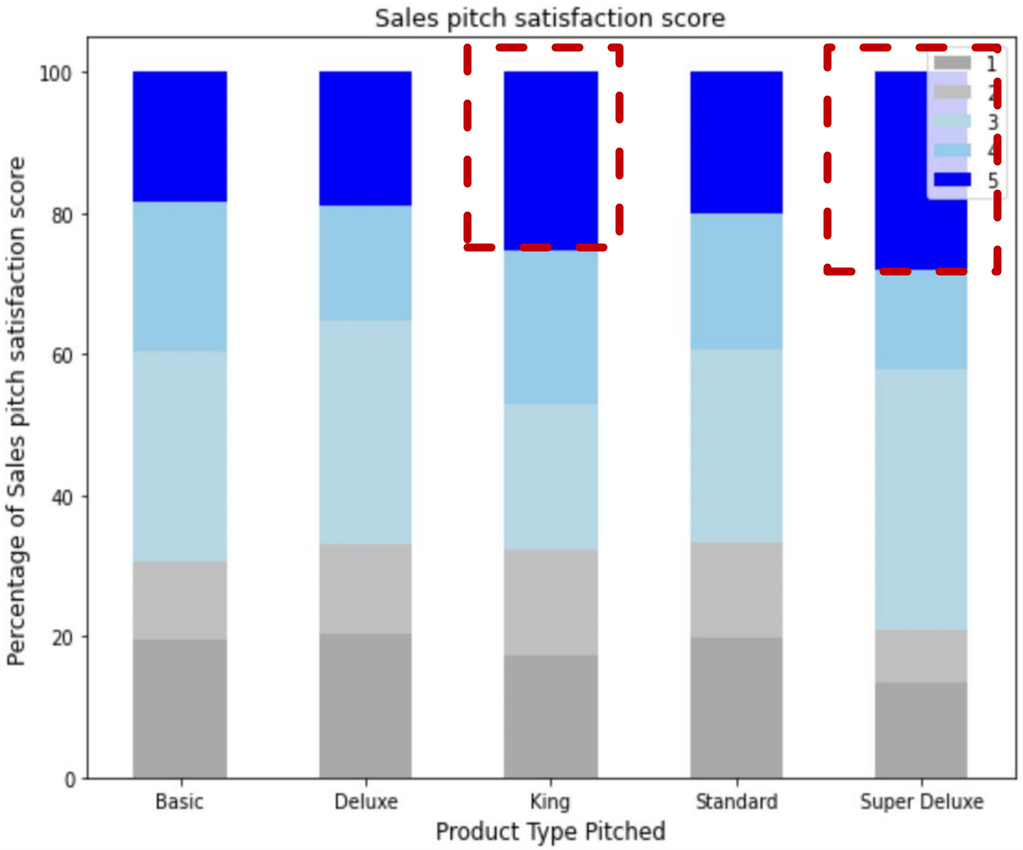
PYTHON CODE:

```
prod_taken = df.groupby(["ProductPitched", "ProductTaken"])["ProductTaken"].count().unstack().fillna(0)
print(prod_taken)
prod_taken = prod_taken.div(prod_taken.sum(axis=1), axis=0)*100
prod_taken.plot(kind='bar', stacked=True, figsize=(9,7),color=['lightblue', 'silver'])
plt.title('Product Type Taken Or Not', fontsize=13)
plt.xlabel('Product Type Pitched', fontsize=12)
plt.ylabel('Percentage of Products Taken Or Not', fontsize=12)
plt.xticks(rotation=0, ha='center')

plt.legend(prod_taken.columns, fontsize=12)
```

INSIGHTS:

- Basic product is the most sought-after product and has garnered greater customer base for the company amongst the 5 products.
- Supreme Deluxe and King products are the least likely to be selected in comparison to other segments

HOW SATISFIED IS EACH SEGMENT WITH THE SALES PITCH?



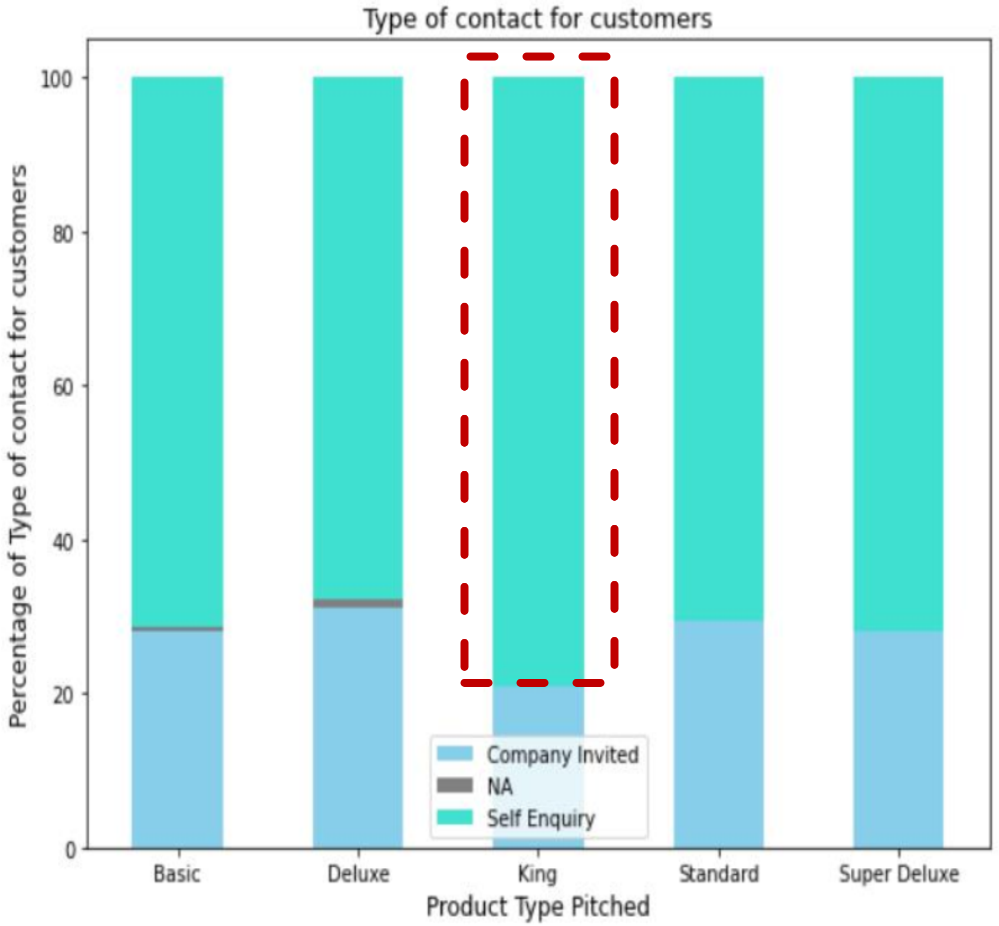
PYTHON CODE:

```
Pitch_Satisfaction_Score = df.groupby(["ProductPitched", "PitchSatisfactionScore"])["PitchSatisfactionScore"]\
.count().unstack().fillna(0)
Pitch_Satisfaction_Score = Pitch_Satisfaction_Score.div(Pitch_Satisfaction_Score.sum(axis=1), axis=0)*100
Pitch_Satisfaction_Score.plot(kind='bar', stacked=True, figsize=(9,7),\
                              color=['darkgrey','silver','lightblue','skyblue','blue'])
plt.title('Sales pitch satisfaction score', fontsize=13)
plt.xlabel('Product Type Pitched', fontsize=12)
plt.ylabel('Percentage of Sales pitch satisfaction score', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(Pitch_Satisfaction_Score.columns)
```

INSIGHTS:

- King and Super Deluxe user are most pitch satisfied and a higher % of these 2 segments have given a rating of 5 (in comparison to other segments)

ARE THESE PRODUCT USER SEGMENTS CONTACTED BY THE COMPANY?



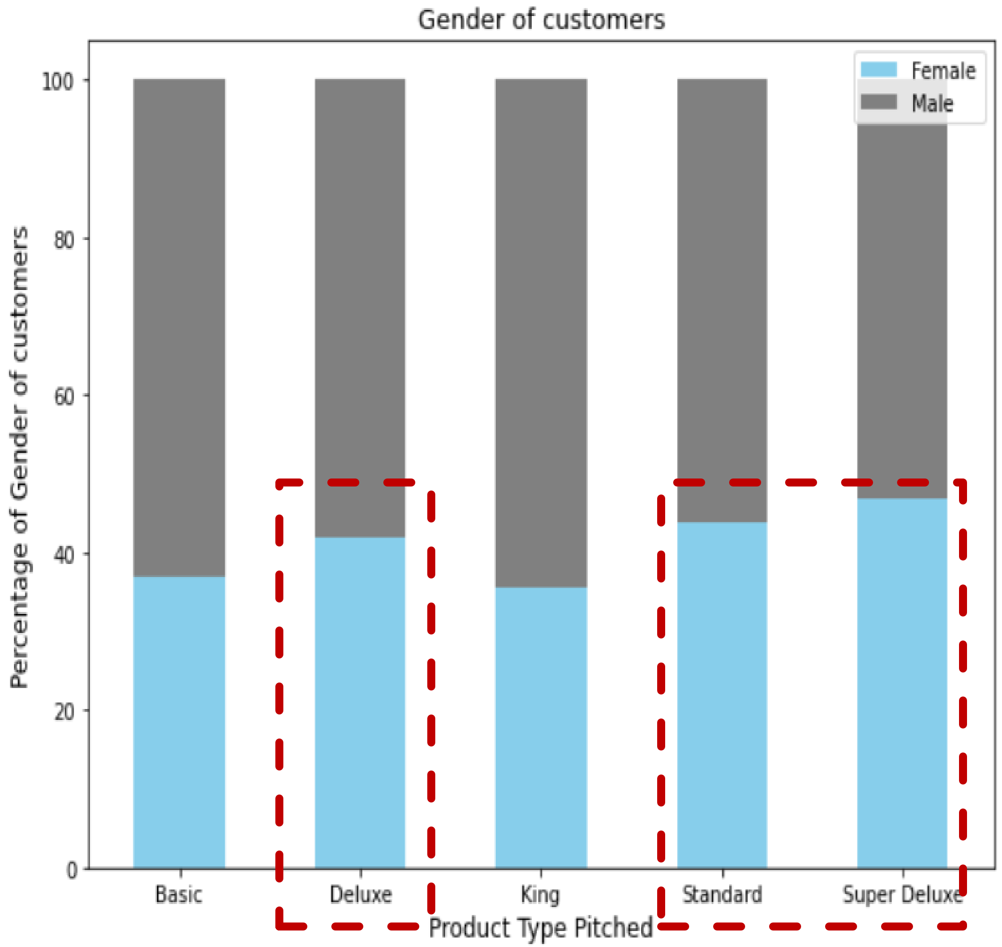
PYTHON CODE:

```
type_of_contact = df.groupby(["ProductPitched", "TypeofContact"])["TypeofContact"].count().unstack().fillna(0)
type_of_contact = type_of_contact.div(type_of_contact.sum(axis=1), axis=0)*100
type_of_contact.plot(kind='bar', stacked=True, figsize=(9,7),color=['skyblue','grey', 'turquoise'])
plt.title('Type of contact for customers', fontsize=13)
plt.xlabel('Product Type Pitched', fontsize=12)
plt.ylabel('Percentage of Type of contact for customers', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(type_of_contact.columns,loc='upper right')
```

INSIGHTS:

- Majority of the customers, irrespective of segments reach out through self enquiry
- However, King users are the least company invited customers (in comparison to other segments)

IS THERE A GENDER CATEGORIZATION AMONG THE SEGMENTS?



PYTHON CODE:

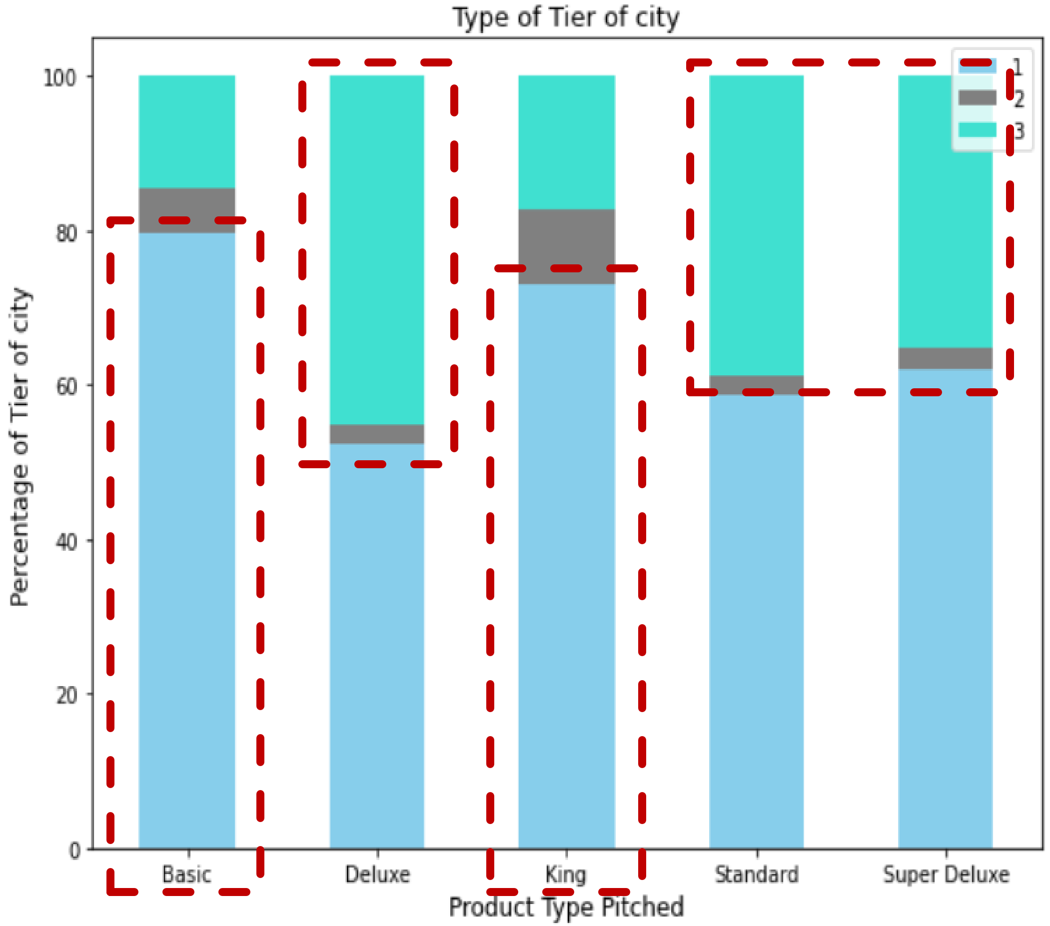
```
Gender = df.groupby(["ProductPitched", "Gender"])["Gender"].count().unstack().fillna(0)
Gender = Gender.div(Gender.sum(axis=1), axis=0)*100
Gender.plot(kind='bar', stacked=True, figsize=(9,7),color=['skyblue','grey'])
plt.title('Gender of customers', fontsize=13)
plt.xlabel('Product Type Pitched', fontsize=12)
plt.ylabel('Percentage of Gender of customers', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(Gender.columns)
```

INSIGHTS:

- All segments have relatively more males than females
- Super Deluxe, Deluxe and Standard have slightly higher proportion of females (in comparison to other segments)

1. What are the trends observed in product preference and demographics within the 5-user segments?

WHERE DO ALL THE SEGMENT BELONG?



PYTHON CODE:

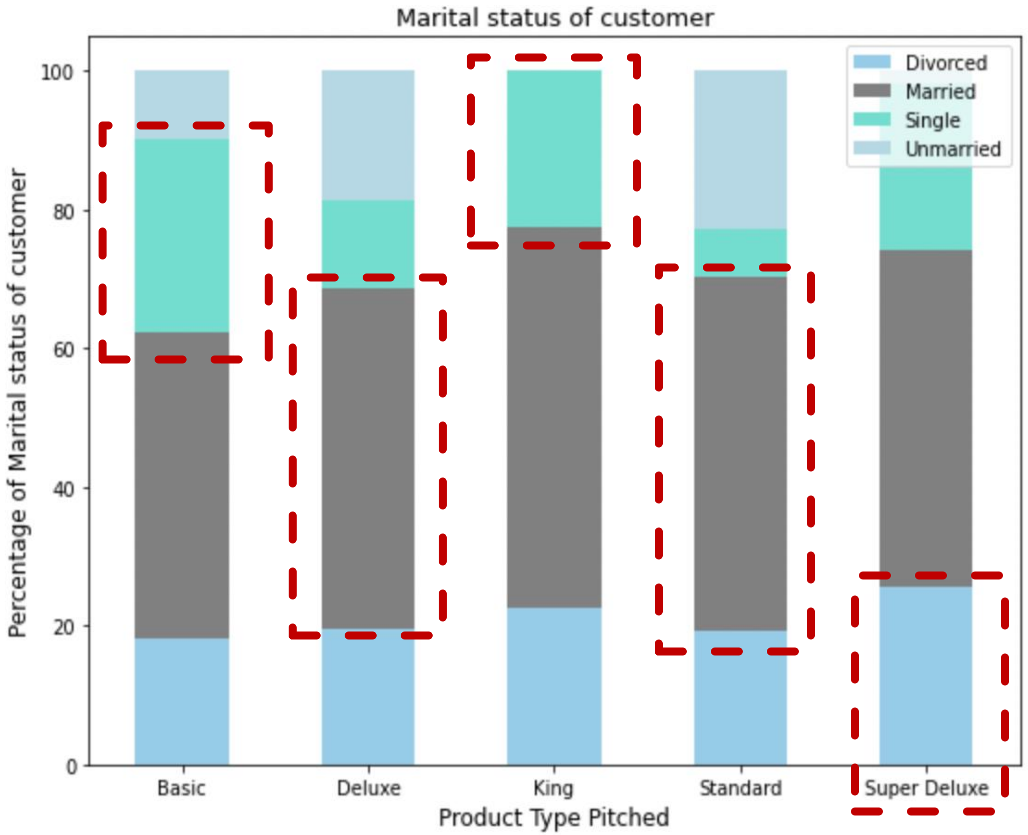
```
tier_of_city = df.groupby(["ProductPitched", "CityTier"])["CityTier"].count().unstack().fillna(0)
tier_of_city = tier_of_city.div(tier_of_city.sum(axis=1), axis=0)*100
tier_of_city.plot(kind='bar', stacked=True, figsize=(9,7),color=['skyblue','grey','turquoise'])
plt.title('Type of Tier of city', fontsize=13)
plt.xlabel('Product Type Pitched', fontsize=12)
plt.ylabel('Percentage of Tier of city', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(tier_of_city.columns)
```

INSIGHTS:

- Basic and King product users are more likely to be found in tier 1 and 2 cities (in comparison to other segments)
- Deluxe, followed by standard and super deluxe product users are more likely to be from tier 3 city types (in comparison to other segments)

1. What are the trends observed in product preference and demographics within the 5-user segments?

WHAT IS THE MARITAL STATUS OF THE PRODUCT USER SEGMENTS?



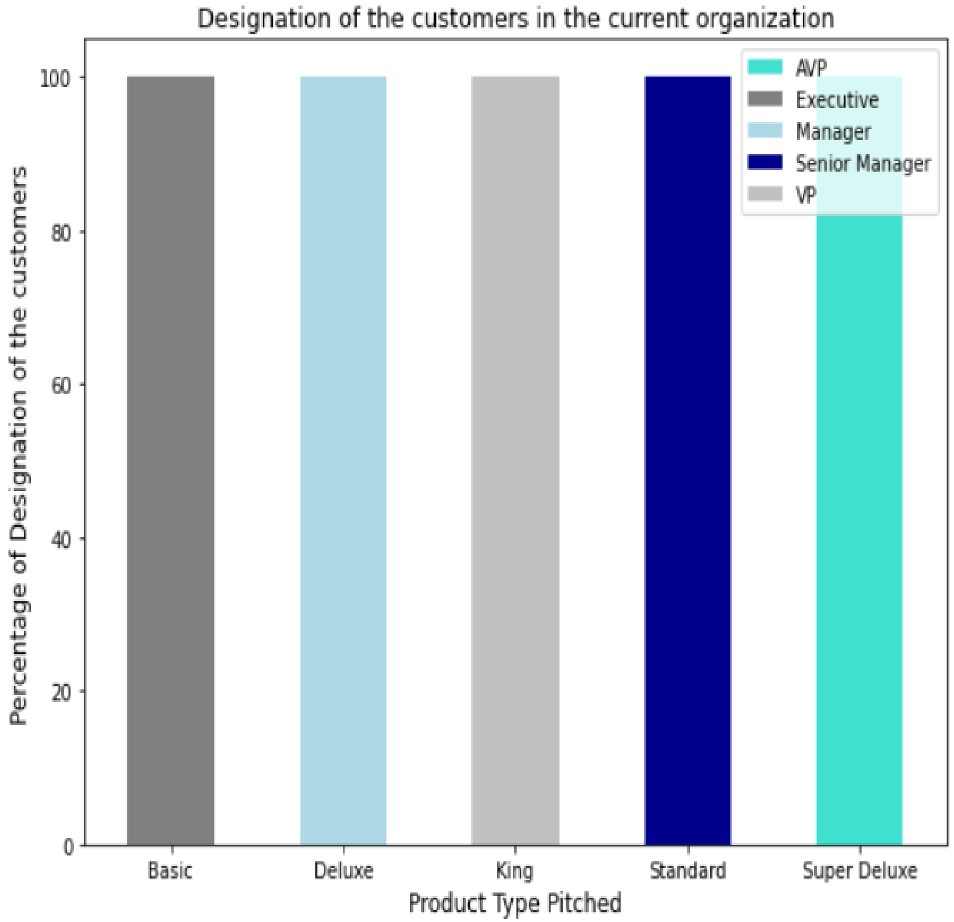
PYTHON CODE:

```
Marital_Status = df.groupby(["ProductPitched", "MaritalStatus"])["MaritalStatus"].count().unstack().fillna(0)
Marital_Status = Marital_Status.div(Marital_Status.sum(axis=1), axis=0)*100
Marital_Status.plot(kind='bar', stacked=True, figsize=(9,7),color=['skyblue','grey','turquoise','lightblue'])
plt.title('Marital status of customer', fontsize=13)
plt.xlabel('Product Type Pitched', fontsize=12)
plt.ylabel('Percentage of Marital status of customer', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(Marital_Status.columns,loc='upper right')
```

INSIGHTS:

- Basic and King product users are more likely to be single (in comparison to other segments)
- Standard and deluxe users are more likely to be unmarried couples (in comparison to other segments)
- Super Deluxe users are most likely to be either divorced or single (in comparison to other segments)

WHAT IS THE DESIGNATION OF EACH USER SEGMENT?



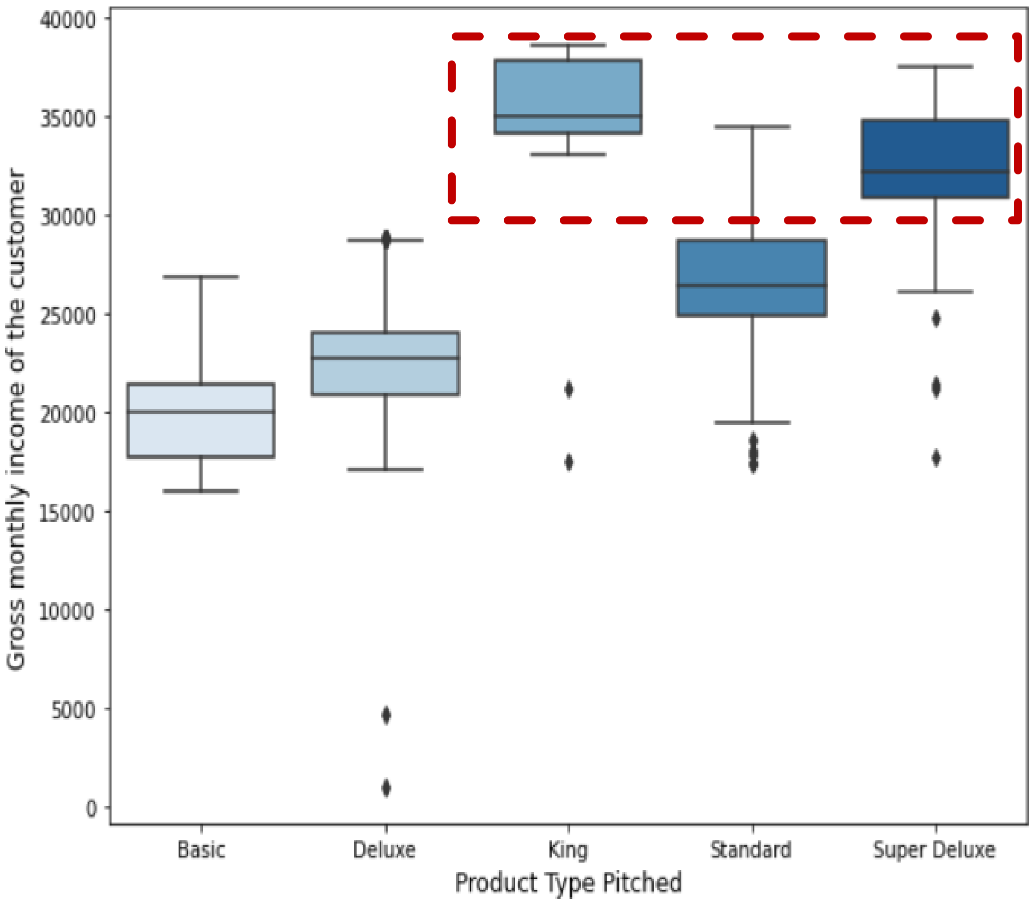
PYTHON CODE:

```
Designation = df.groupby(["ProductPitched", "Designation"])["Designation"].count().unstack().fillna(0)
Designation = Designation.div(Designation.sum(axis=1), axis=0)*100
Designation.plot(kind='bar', stacked=True, figsize=(9,7),\
                 color=['turquoise','grey','lightblue','darkblue','silver'])
plt.title('Designation of the customers in the current organization', fontsize=13)
plt.xlabel('Product Type Pitched', fontsize=12)
plt.ylabel('Percentage of Designation of the customers', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(Designation.columns);
```

INSIGHTS:

- Very interestingly, the data suggests that –
  - All Basic users are Executives,
  - All Deluxe user are Managers,
  - All Standard users are Senior Managers,
  - All Super Deluxe users are AVPs, and
  - All King users are VPs
- To probe further on this, we have checked their income levels in the next slide

WHAT IS THE AFFLUENCE LEVELS OF EACH PRODUCT USER GROUP?



PYTHON CODE:

```
fig2, ax2 = plt.subplots(figsize=(9,7))
ax2 = sns.boxplot(x="ProductPitched", y="MonthlyIncome", data=df, palette='Blues')
plt.ylabel('Gross monthly income of the customer', fontsize=12)
plt.xlabel('Product Type Pitched', fontsize=12)
plt.show()
```

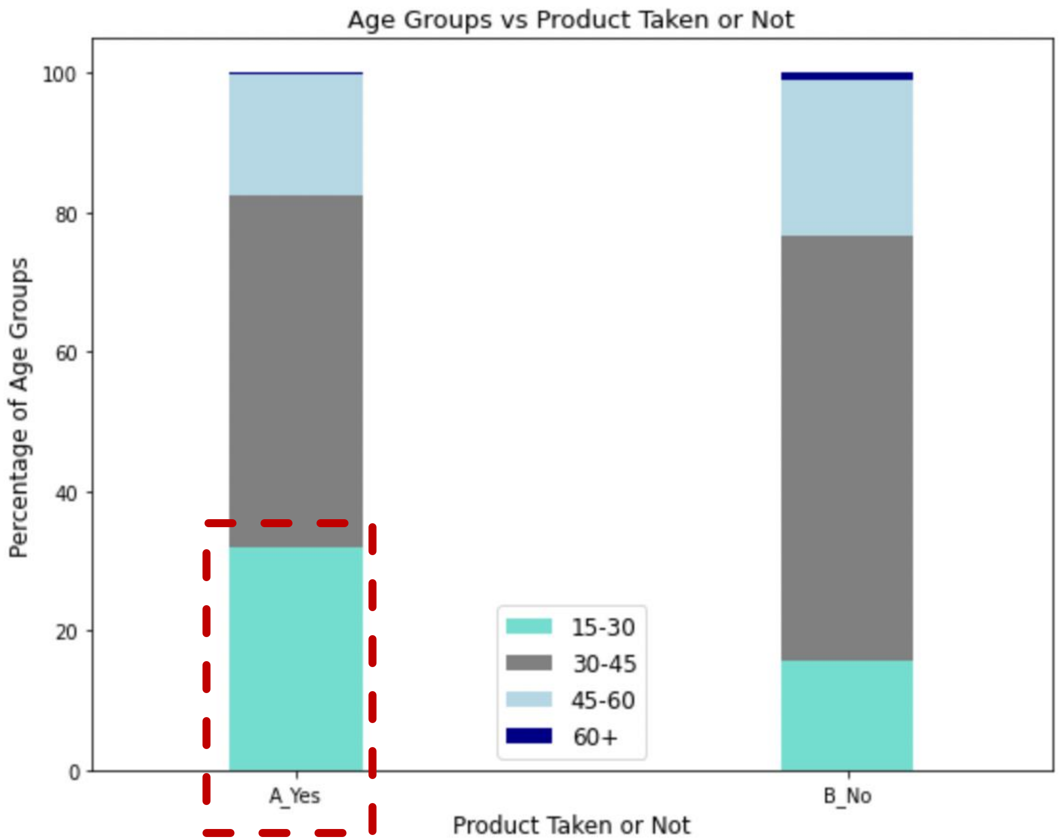
INSIGHTS:

- Based on the income and the prior understanding of designation, we find the results consistent
  - King and Super Deluxe users are VPs and AVPs. So, they have the highest income (in comparison to other segments)
  - They are followed by Standard users who are Senior Managers (in comparison to other segments)

AGENDA

- TEAM INTRODUCTION
- OBJECTIVE AND KEY TAKEAWAYS
- DATA ANALYSIS
  - DATA CLEANING
  - EXPLORATORY DESCRIPTIVE ANALYSIS OF USER-GROUPS
  - PRODUCT ADOPTION AND PREDICTION MODEL

WHAT AGE GROUP DO THE PRODUCT BUYERS BELONG TO?



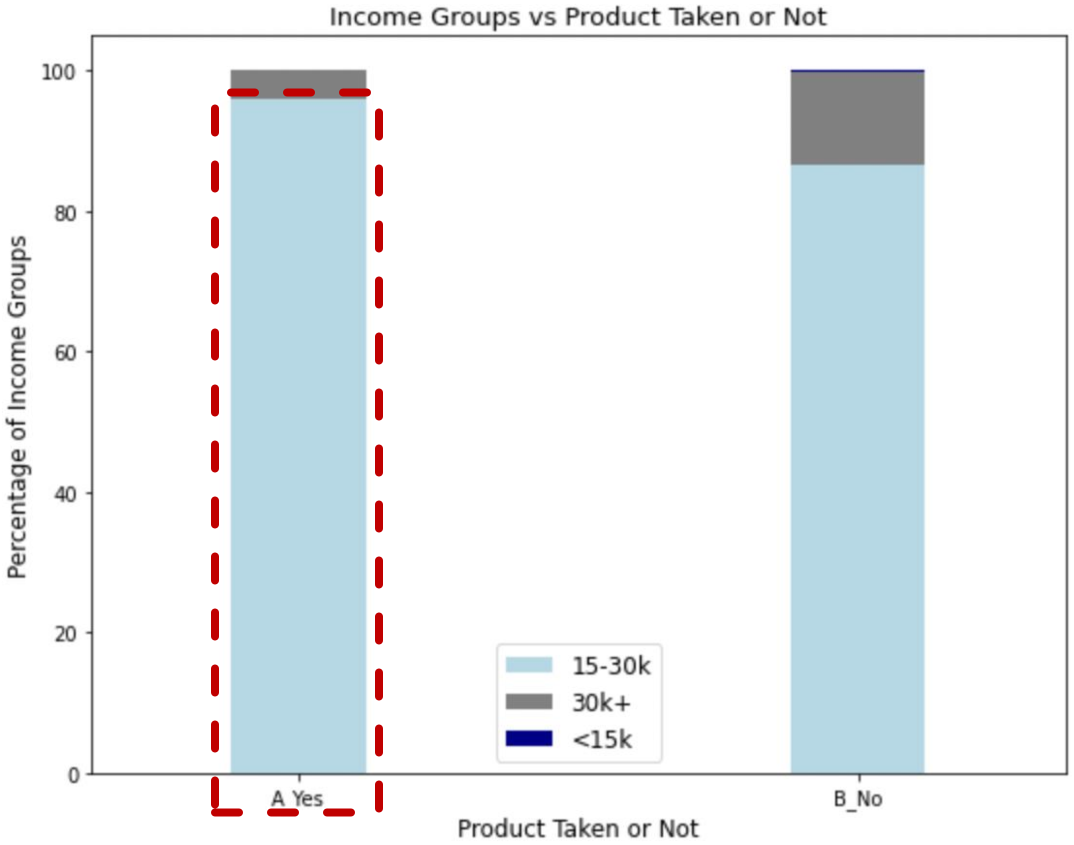
PYTHON CODE:

```
age_groups = groupings.groupby(["ProductTaken", "AgeGroup"])["AgeGroup"].count().unstack().fillna(0)
age_groups = age_groups.div(age_groups.sum(axis=1), axis=0)*100
age_groups.plot(kind='bar', stacked=True, figsize=(9,7),width=0.24,
                color=['turquoise','grey','lightblue','darkblue','silver'])
plt.title('Age Groups vs Product Taken or Not', fontsize=13)
plt.xlabel('Product Taken or Not', fontsize=12)
plt.ylabel('Percentage of Age Groups', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(age_groups.columns, fontsize=12)
```

NSIGHTS:

- It is evident that there is higher % of 15–30-year-old population among the product buyers (in comparison to non-buyers)

WHAT IS THE INCOME LEVEL OF THE CURRENT PRODUCT BUYERS?



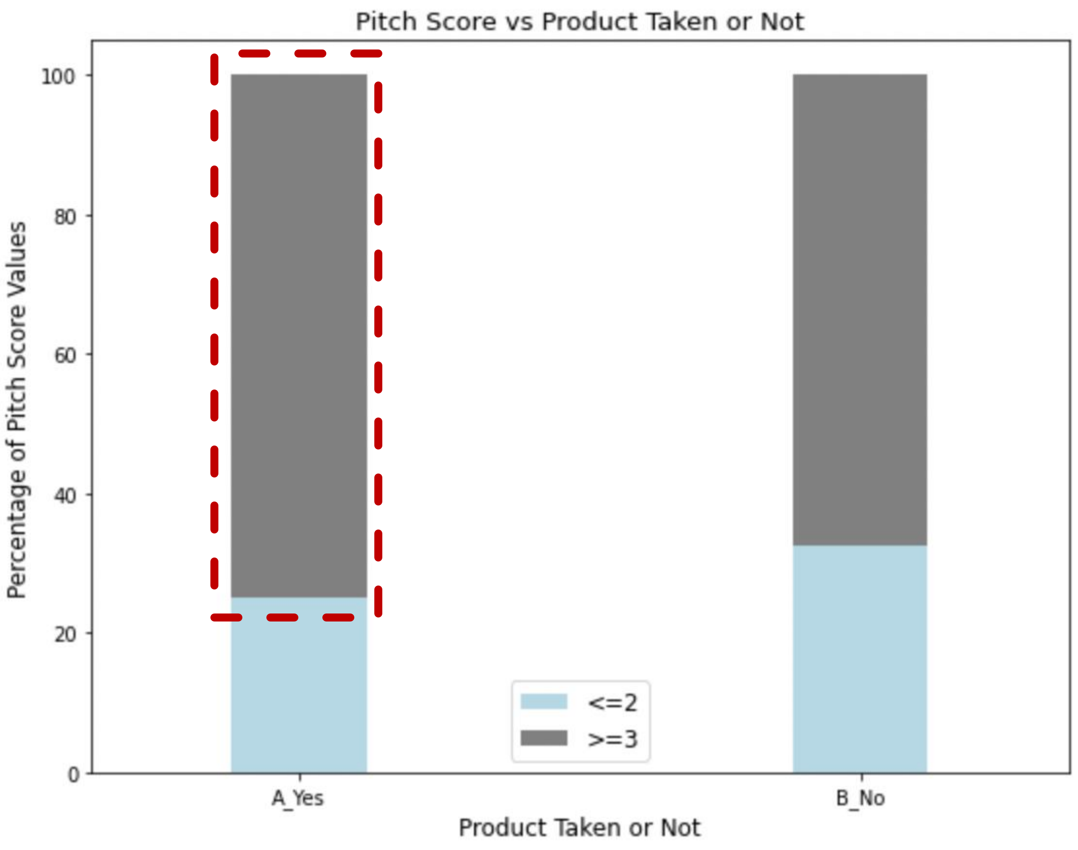
PYTHON CODE:

```
income_groups = groupings.groupby(["ProductTaken", "IncomeGroup"])["IncomeGroup"].count().unstack().fillna(0)
income_groups = income_groups.div(income_groups.sum(axis=1), axis=0)*100
income_groups.plot(kind='bar', stacked=True, figsize=(9,7),width=0.24,color=['lightblue','grey','darkblue'])
plt.title('Income Groups vs Product Taken or Not', fontsize=13)
plt.xlabel('Product Taken or Not', fontsize=12)
plt.ylabel('Percentage of Income Groups', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(income_groups.columns, fontsize=12)
```

INSIGHTS:

- Product buyers are more likely to belong to ~$15-30K income group (in comparison to non-buyers)

HOW ARE THE PRODUCT BUYERS REACTING TO SALES PITCH?



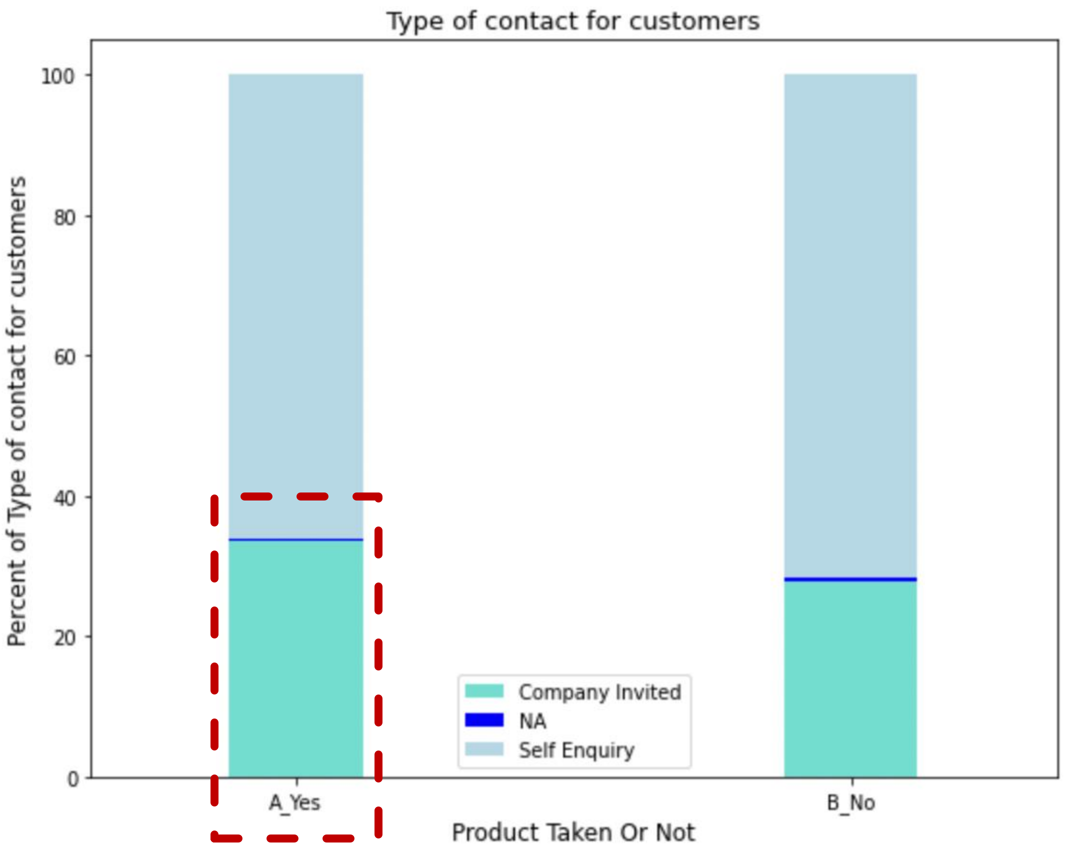
PYTHON CODE:

```
pitch_score_groups = groupings.groupby(["ProductTaken", "PitchScoreGroup"])["PitchScoreGroup"].
count().unstack().fillna(0)
pitch_score_groups = pitch_score_groups.div(pitch_score_groups.sum(axis=1), axis=0)*100
pitch_score_groups.plot(kind='bar', stacked=True, figsize=(9,7), width = 0.24, color=['lightblue','grey'])
plt.title('Pitch Score vs Product Taken or Not', fontsize=13)
plt.xlabel('Product Taken or Not', fontsize=12)
plt.ylabel('Percentage of Pitch Score Values', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(pitch_score_groups.columns, fontsize=12)
```

INSIGHTS:

- Product buyers are more willing to choose a product based on sales pitch. So, they are more likely to provide a higher score to the sales pitch satisfaction (in comparison to non-buyers)

HOW DID THE COMPANY BRING PRODUCT BUYERS ON BOARD?



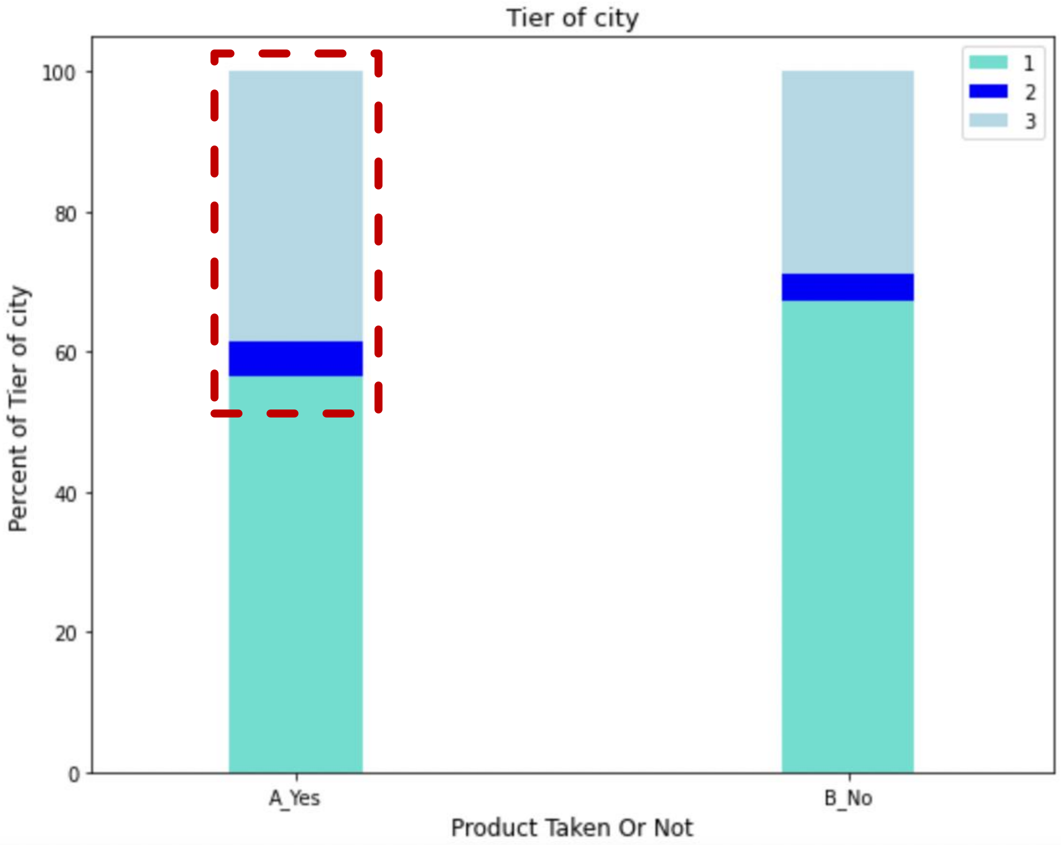
PYTHON CODE:

```
contact_type = df.groupby(["ProductTaken", "TypeofContact"])["TypeofContact"].count().unstack().fillna(0)
contact_type = contact_type.div(contact_type.sum(axis=1), axis=0)*100
# fig, ax = plt.subplots(figsize=(9,9))
contact_type.plot(kind='bar', stacked=True, figsize=(9,7), width= 0.24, color=['turquoise','blue','lightblue'])
plt.xticks(rotation=0, ha='center')
plt.title('Type of contact for customers', fontsize=13)
plt.ylabel('Percent of Type of contact for customers', fontsize=12)
plt.xlabel('Product Taken Or Not', fontsize=12)
plt.legend(contact_type.columns);
```

INSIGHTS:

- Product buyers are more likely to be company invited customers who liked the sales pitch and took the product (in comparison to non-buyers)

WHICH CITY TIERS DO THE CURRENT PRODUCT BUYERS BELONG TO?



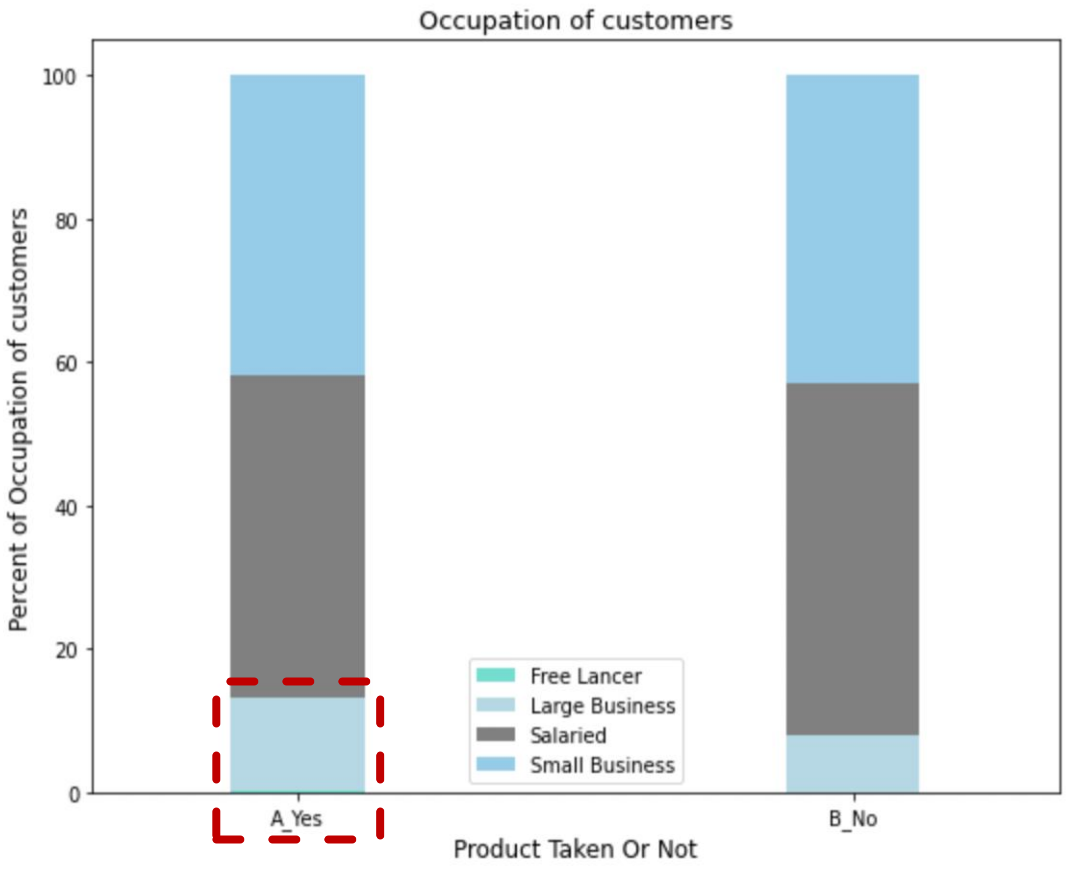
PYTHON CODE:

```
city_tier = df.groupby(["ProductTaken", "CityTier"])["CityTier"].count().unstack().fillna(0)
city_tier = city_tier.div(city_tier.sum(axis=1), axis=0)*100
# fig, ax = plt.subplots(figsize=(9,9))
city_tier.plot(kind='bar', stacked=True, figsize=(9,7), width= 0.24, color=['turquoise','blue','lightblue'])
plt.xticks(rotation=0, ha='center')
plt.title('Tier of city', fontsize=13)
plt.ylabel('Percent of Tier of city', fontsize=12)
plt.xlabel('Product Taken Or Not', fontsize=12)
plt.legend(city_tier.columns);
```

INSIGHTS:

- More Product Buyers are likely to be from Tier 1 and Tier 3 cities (in comparison to non-buyers)

WHAT IS THE OCCUPATION OF THE CURRENT PRODUCT BUYERS?



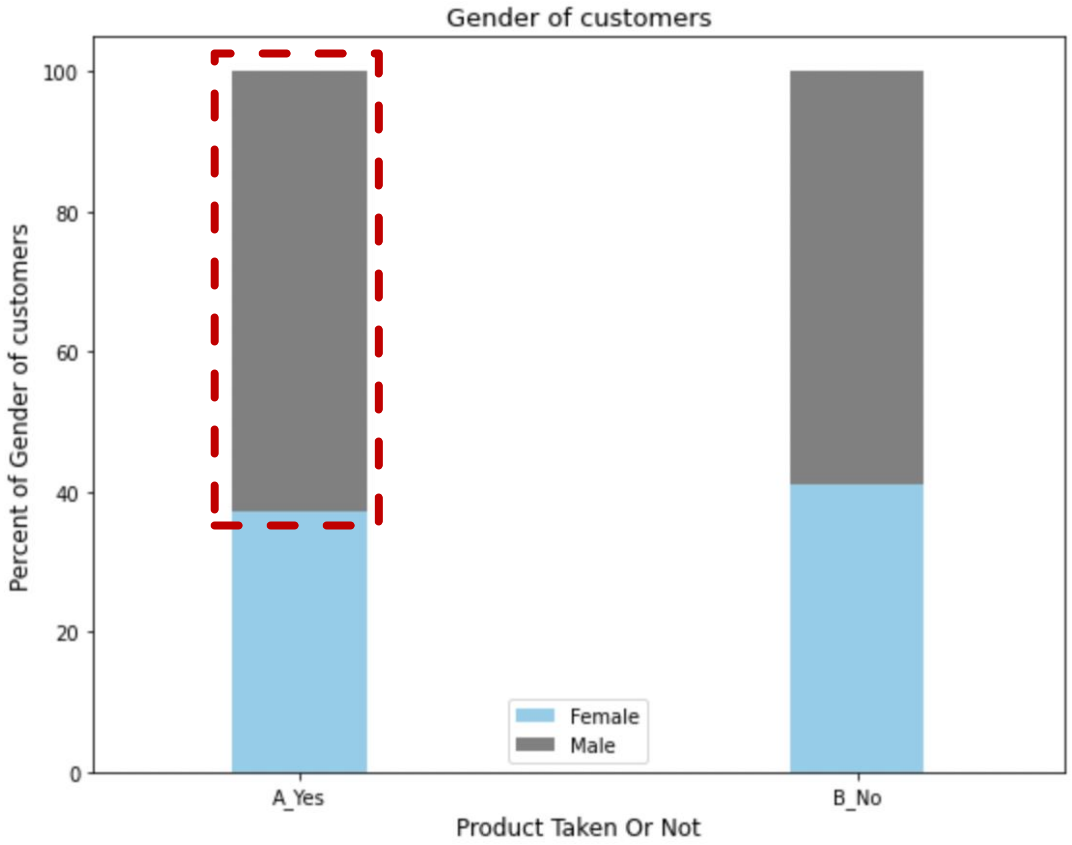
PYTHON CODE:

```
occupation = df.groupby(["ProductTaken", "Occupation"])["Occupation"].count().unstack().fillna(0)
occupation = occupation.div(occupation.sum(axis=1), axis=0)*100
# fig, ax = plt.subplots(figsize=(9,9))
occupation.plot(kind='bar', stacked=True, figsize=(9,7),width= 0.24, color=['turquoise','lightblue','grey','skyblue'])
plt.xticks(rotation=0, ha='center')
plt.title('Occupation of customers', fontsize=13)
plt.ylabel('Percent of Occupation of customers', fontsize=12)
plt.xlabel('Product Taken Or Not', fontsize=12)
plt.legend(occupation.columns);
```

INSIGHTS:

- Product buyers are more likely to be large businessmen (in comparison to non-buyers)

WHICH GENDER SHARE THE MAJOR CHUNK OF PRODUCT BUYERS?



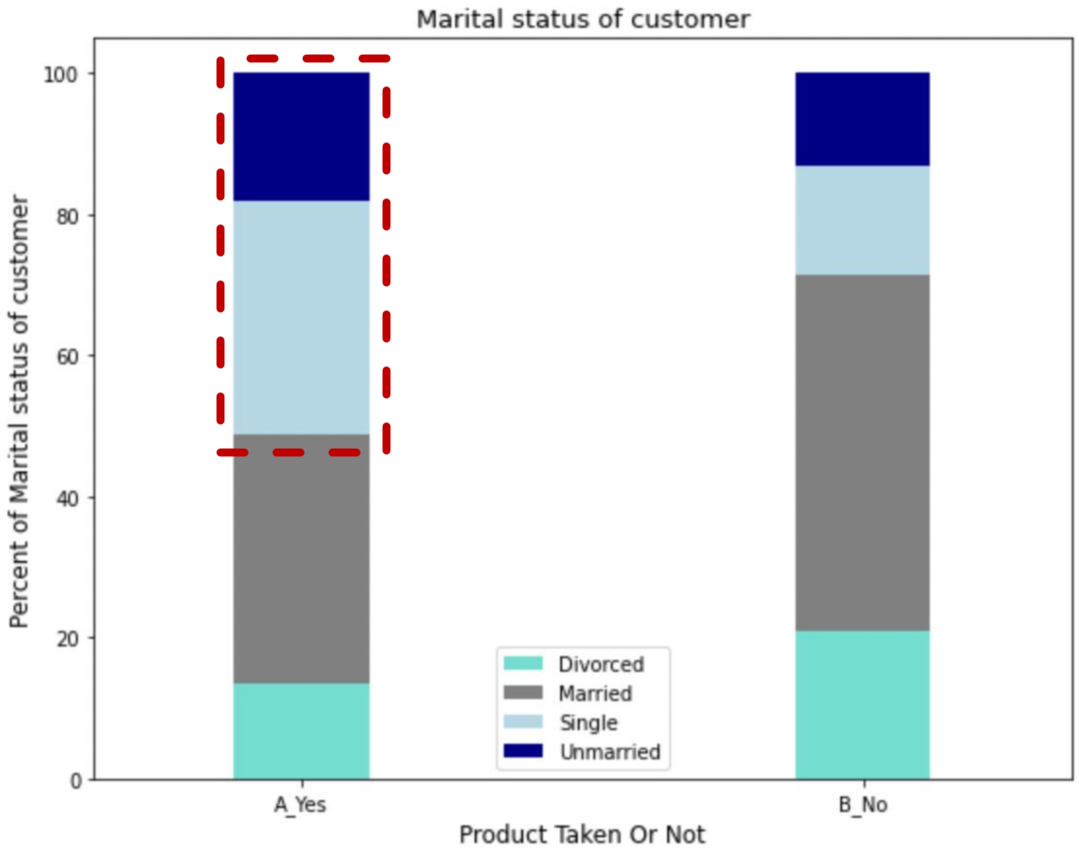
PYTHON CODE:

```
gender = df.groupby(["ProductTaken", "Gender"])["Gender"].count().unstack().fillna(0)
gender = gender.div(gender.sum(axis=1), axis=0)*100
# fig, ax = plt.subplots(figsize=(9,9))
gender.plot(kind='bar', stacked=True, figsize=(9,7),width= 0.24, color=['skyblue','grey'])
plt.xticks(rotation=0, ha='center')
plt.title('Gender of customers', fontsize=13)
plt.ylabel('Percent of Gender of customers', fontsize=12)
plt.xlabel('Product Taken Or Not', fontsize=12)
plt.legend(gender.columns);
```

INSIGHTS:

- Product buyers are more likely to be belonging to the male population (in comparison to non-buyers)

WHAT IS THE MARITAL STATUS OF THE CURRENT PRODUCT BUYERS?



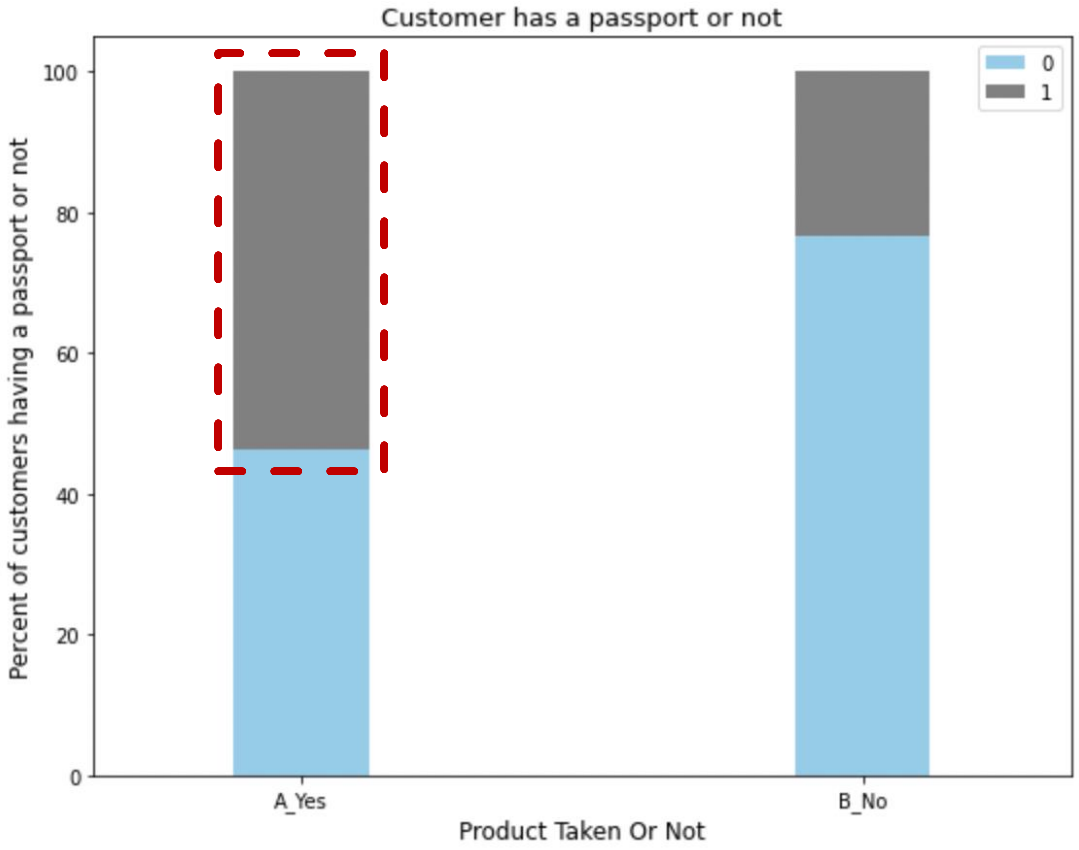
PYTHON CODE:

```
marital_status = df.groupby(["ProductTaken", "MaritalStatus"])["MaritalStatus"].count().unstack().fillna(0)
marital_status = marital_status.div(marital_status.sum(axis=1), axis=0)*100
# fig, ax = plt.subplots(figsize=(9,9))
marital_status.plot(kind='bar', stacked=True, figsize=(9,7),width= 0.24,
                    color=['turquoise','grey','lightblue','darkblue'])
plt.xticks(rotation=0, ha='center')
plt.title('Marital status of customer', fontsize=13)
plt.ylabel('Percent of Marital status of customer', fontsize=12)
plt.xlabel('Product Taken Or Not', fontsize=12)
plt.legend(marital_status.columns);
```

INSIGHTS:

- Product buyers are more likely to be unmarried or single (in comparison to non-buyers)
- Non-buyers are more likely to be married (in comparison to buyers)

DO THE CURRENT PRODUCT BUYERS HAVE A PASSPORT?



PYTHON CODE:

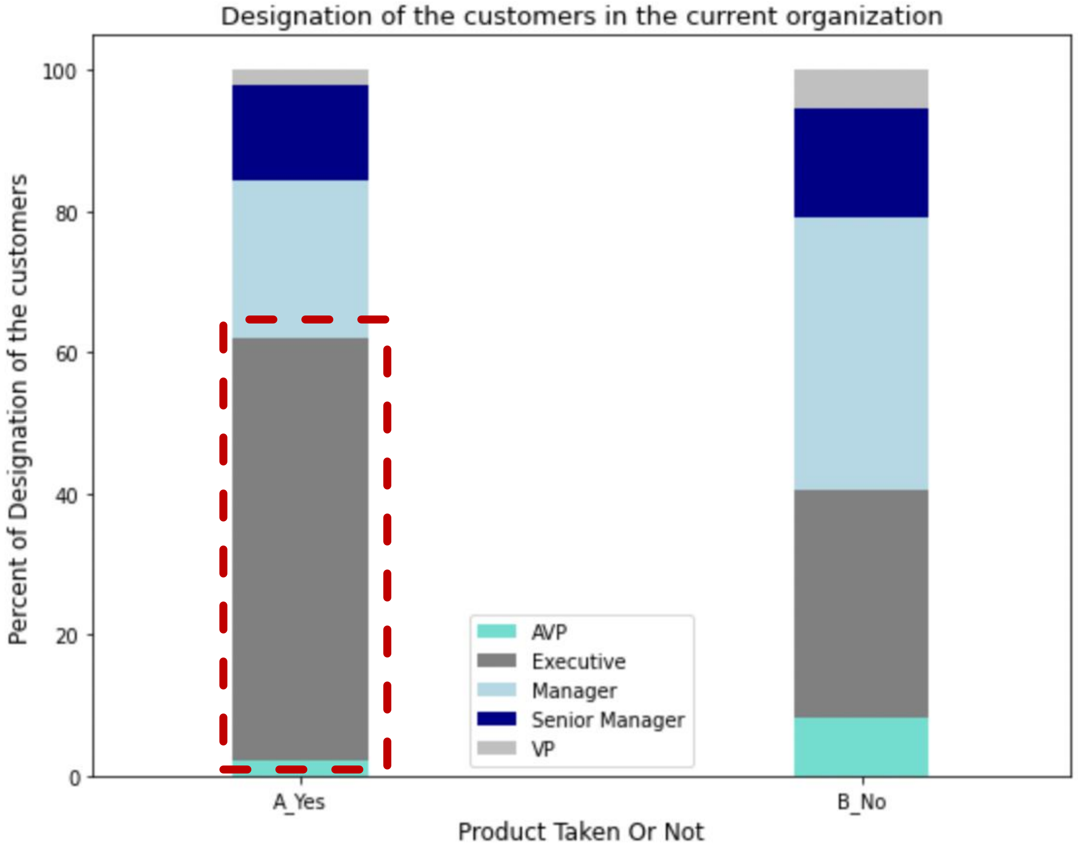
```
Passport = df.groupby(["ProductPitched", "Passport"])["Passport"].count().unstack().fillna(0)
Passport = Passport.div(Passport.sum(axis=1), axis=0)*100
Passport.plot(kind='bar', stacked=True, figsize=(9,7),color=['skyblue','grey'])
plt.title('Customer has a passport or not', fontsize=13)
plt.xlabel('Product Type Pitched', fontsize=12)
plt.ylabel('Percentage of Customer has a passport or not', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(Passport.columns,loc='upper right');
```

INSIGHTS:

- Product buyers generally own a passport (in comparison to non-buyers).This shows their willingness to travel more.

2. What is the story of the users who are willing to take a product based on product pitched?

WHAT IS THE DESIGNATION OF THE CURRENT PRODUCT BUYERS?



PYTHON CODE:

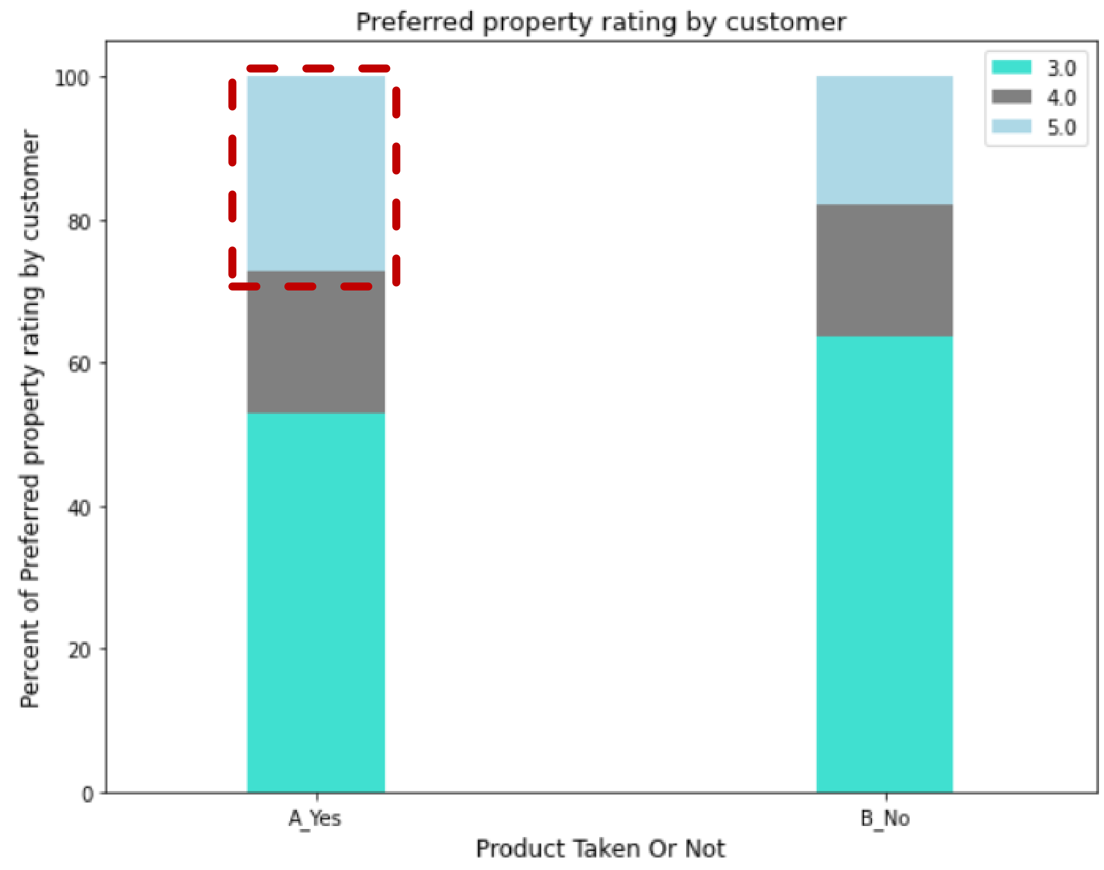
```
Designation = df.groupby(["ProductPitched", "Designation"])["Designation"].count().unstack().fillna(0)
Designation = Designation.div(Designation.sum(axis=1), axis=0)*100
Designation.plot(kind='bar', stacked=True, figsize=(9,7),color=['turquoise','grey','lightblue','darkblue','silver'])
plt.title('Designation of the customers in the current organization', fontsize=13)
plt.xlabel('Product Type Pitched', fontsize=12)
plt.ylabel('Percentage of Designation of the customers', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.legend(Designation.columns);
```

INSIGHTS:

- Product buyers are very likely to be Executives while non-buyers are more likely to be Managers or AVPs

2. What is the story of the users who are willing take a product based on product pitched?

WHICH KIND OF HOTEL PROPERTIES DO PRODUCT BUYERS PREFER?



PYTHON CODE:

```
property_star = df.groupby(["ProductTaken", "PreferredPropertyStar"])["PreferredPropertyStar"]\
.count().unstack().fillna(0)
property_star = property_star.div(property_star.sum(axis=1), axis=0)*100
property_star.plot(kind='bar', stacked=True, figsize=(9,7),width= 0.24, color=['turquoise','grey','lightblue'])
plt.xticks(rotation=0, ha='center')
plt.title('Preferred property rating by customer', fontsize=13)
plt.ylabel('Percent of Preferred property rating by customer', fontsize=12)
plt.xlabel('Product Taken Or Not', fontsize=12)
plt.legend(property_star.columns);
```

INSIGHTS:

- Product buyers are more likely to stay in a 5 Star rated hotel (in comparison to non-buyers), while non-buyers seem to choose 3-star hotels more

WE USED MULTIPLE TECHNIQUES TO CREATE THE PREDICTION MODEL

INPUTS

DEPENDENT VARIABLE (Y) –
PRODUCT TAKEN FLAG

INDEPENDENT VARIABLES (X_i) –
REMAINING 18 VARIABLES

TOP 15 RANDOM FOREST PREDICTORS	DIFFERENTIATION LEVEL IN EXPORATORY ANALYSIS
Monthly Income	MEDIUM
Age	HIGH
Duration of Pitch	LOW
Have Passport	HIGH
Number of Trips	LOW
Pitch Satisfaction Score	LOW
Marital Status	MEDIUM
Number of follow-ups	MEDIUM
Preferred property star	HIGH
Product pitched	HIGH
Occupation	HIGH
City Tier	MEDIUM
Designation	HIGH
Number of Children Visiting	LOW
Number of Person Visiting	LOW

We are choosing top 10 variables based off variable importance analyses from Random forest and distinct differentiation that is seen in exploratory analysis

4. What kind of model can we create to measure to take a new product? What will be its accuracy?

WE WERE ABLE TO CREATE A PREDICTION MODEL OF ~90% ACCURACY

FINALIZED TOP 10 VARIABLES USING R.F.

- Monthly Income
- Age
- Have Passport
- Marital Status
- # of follow-ups
- Preferred property
- product pitched
- Occupation
- City Tier
- Designation

R.F. CODE :

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OrdinalEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from xgboost import XGBClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.model_selection import cross_val_score
```

```
ordinalencoder = OrdinalEncoder()
X_train[:,0:10] = ordinalencoder.fit_transform(X_train[:,0:10])
X_test[:,0:10] = ordinalencoder.transform(X_test[:,0:10])
sc = StandardScaler()
X_train[:,10:] = sc.fit_transform(X_train[:,10:])
X_test[:,10:] = sc.transform(X_test[:,10:])
```

```
le = LabelEncoder()
y_train = le.fit_transform(y_train)
y_test = le.transform(y_test)
```

```
rf_classifier = RandomForestClassifier(n_estimators = 100, random_state = 0)
rf_classifier.fit(X_train, y_train)
y_pred_rf = rf_classifier.predict(X_test)
```

```
important_features = pd.Series(rf_classifier.feature_importances_,
                               index = X_df.columns, name = "Important Features")
sort_values(ascending=False)
```

XG BOOST MODELLING

XGB. CODE :

```
ordinalencoder = OrdinalEncoder()
X_train[:,0:7] = ordinalencoder.fit_transform(X_train[:,0:7])
X_test[:,0:7] = ordinalencoder.transform(X_test[:,0:7])
sc = StandardScaler()
X_train[:,7:] = sc.fit_transform(X_train[:,7:])
X_test[:,7:] = sc.transform(X_test[:,7:])
le = LabelEncoder()
y_train = le.fit_transform(y_train)
y_test = le.transform(y_test)
```

```
xgbmodel = XGBClassifier(eval_metric='error')
xgbmodel.fit(X_train, y_train)
y_pred = xgbmodel.predict(X_test)
```

MODEL OUTPUT

OUTPUT ACCURACY =

~90%

OUTPUT CODE:

```
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix : \n", cm)
print("Accuracy Score : {:.3f}%".
      format(accuracy_score(y_test, y_pred)*100))
```

```
Confusion Matrix :
 [[100  80]
 [ 12 786]]
Accuracy Score : 90.593%
```

```
accuracies = cross_val_score(estimator = xgbmodel,
                             X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))
```

```
Accuracy: 89.82 %
Standard Deviation: 1.10 %
```

IMPROVEMENTS

If the data had provided information about the self-health care or dietary habits of customers during tours, the model would have been stronger

Neural Networks could have been useful in the analysis of the categorical data

Information about the USP of the company, competitors in the market, could have assisted in SWOT analysis for generating better insights about customers

Post-Covid survey data could have helped to understand the current scenario of willingness to travel

THANK YOU!

ANY QUESTIONS?