Project704_0506_07

ReadMe Documentation

Course: BUDT704
Data Processing and Analysis using Python

Professor: Adam Lee

We are the BlueWin Business Consultancy Services providing our services to the University Of Maryland Alumni Association (as a client).

This is the overview of our project, beginning with the code and then the analysis, findings, suggestions and recommendations.

Libraries used in our Project:

numpy, pandas, matplotlib, seaborn, calendar, sklearn

Our Target Variables:

1. First Time Attendees
2. Major Prospects

Dataset:

This Dataset was provided as an assignment downloadable on this link:

https://docs.google.com/spreadsheets/d/1m__yXchUwkI0SYvaOHVMHL30_TTX8Rk8/edit?usp=sharing&ouid=114449525189224539715&rtpof=true&sd=true

Reading this dataset:

Using the **concat** function in the Pandas library and the **read_Excel** function to read all The sheets in the datasets.

Data Processing:

Checking for Null Values or missing data: df.isnull() function

Checking for duplicate values: df.duplicated()

```
print( df.isnull().sum())
```

```
Event Name                        0
Activity Code                     0
Activity Description              0
Location Code                     0
Location Description              0
Group Code                        0
Group Description                 0
Event Date                        0
Participated                      0
Average Age                       0
First Time Attendees              0
Percentage First Time Attendees   0
Major Prospects                   0
Percentage Major Prospect         0
dtype: int64
```

```
df.duplicated(subset=None,keep='first')
```

```
2019-20  0     False
         1     False
         2     False
         3     False
         4     False
               ...
2013-14  9     False
         10    False
         11    False
         12    False
         13    False
Length: 622, dtype: bool
```

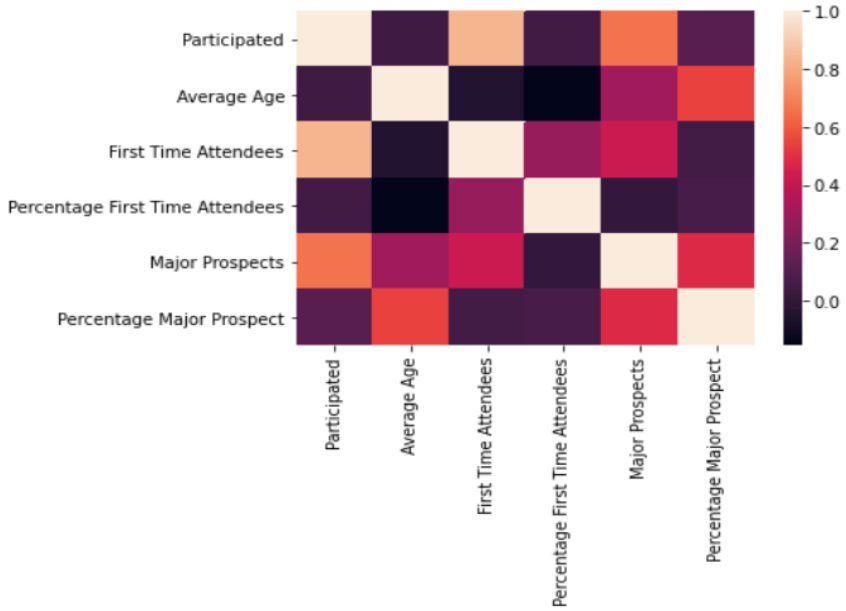
Statistical summary of the dataset: **df.describe()**

	Participated	Average Age	First Time Attendees	Percentage First Time Attendees	Major Prospects	Percentage Major Prospect
count	622.000000	622.000000	622.000000	622.000000	622.000000	622.000000
mean	44.803859	40.117363	13.456592	0.276282	5.966238	0.102214
std	93.165049	9.741459	41.103936	0.242273	14.123466	0.131444
min	1.000000	19.000000	0.000000	0.000000	0.000000	0.000000
25%	10.000000	33.000000	1.000000	0.068523	0.000000	0.000000
50%	20.000000	40.000000	4.000000	0.237327	1.000000	0.058824
75%	44.750000	46.000000	11.000000	0.444444	5.000000	0.166667
max	1657.000000	75.000000	702.000000	1.000000	131.000000	0.818182

For better understanding of the variables amongst themselves and to draw meaningful Relationships the correlation of the variables in the dataset is calculated using the: df.corr()

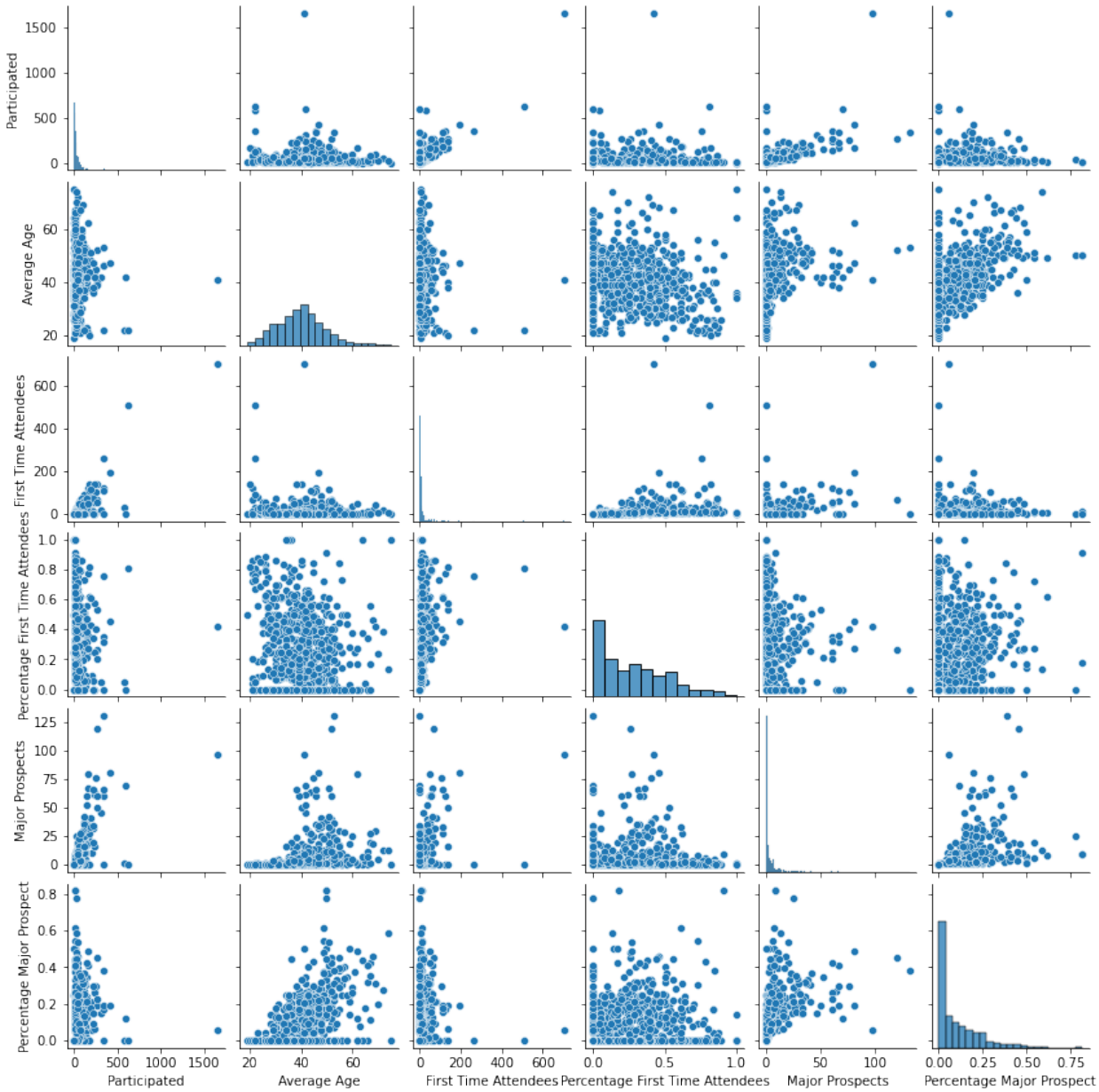
	Participated	Average Age	First Time Attendees	Percentage First Time Attendees	Major Prospects	Percentage Major Prospect
Participated	1.000000	0.037616	0.835996	0.047840	0.658973	0.113415
Average Age	0.037616	1.000000	-0.048204	-0.152633	0.308342	0.549320
First Time Attendees	0.835996	-0.048204	1.000000	0.281961	0.420884	0.051069
Percentage First Time Attendees	0.047840	-0.152633	0.281961	1.000000	0.000751	0.067701
Major Prospects	0.658973	0.308342	0.420884	0.000751	1.000000	0.481370
Percentage Major Prospect	0.113415	0.549320	0.051069	0.067701	0.481370	1.000000

The correlation matrix is visualized using the **seaborn library's function heatmap()**.

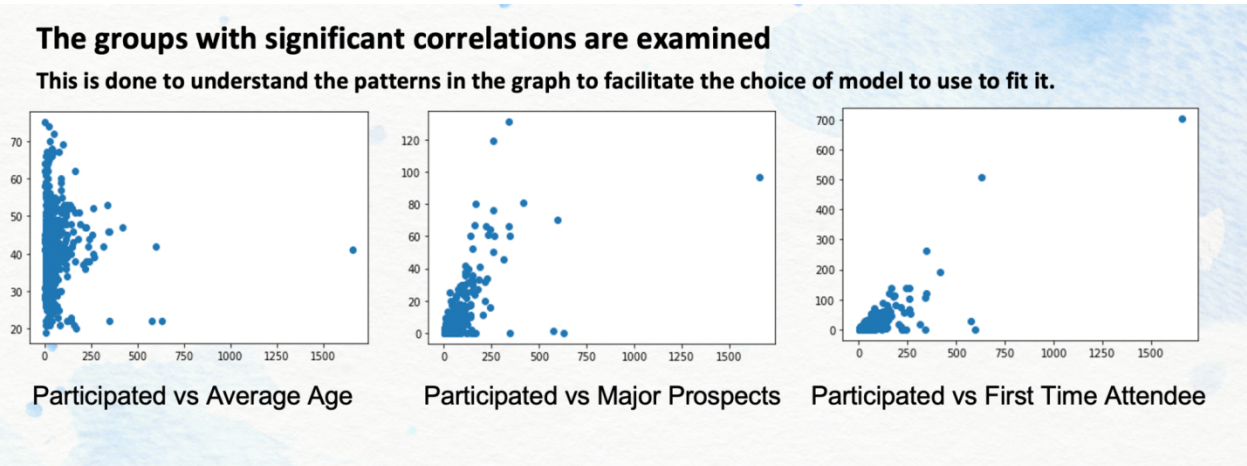


Exploratory Data Analysis:

1. Each variable in the dataset is paired and visualized as graphs/plots using the **Seaborn** library's function: **pairplot()**



2. We use the **scatter() function** from the **matplotlib.pyplot** library to visualize three scatter plots to show the correlation between participation and other variables.



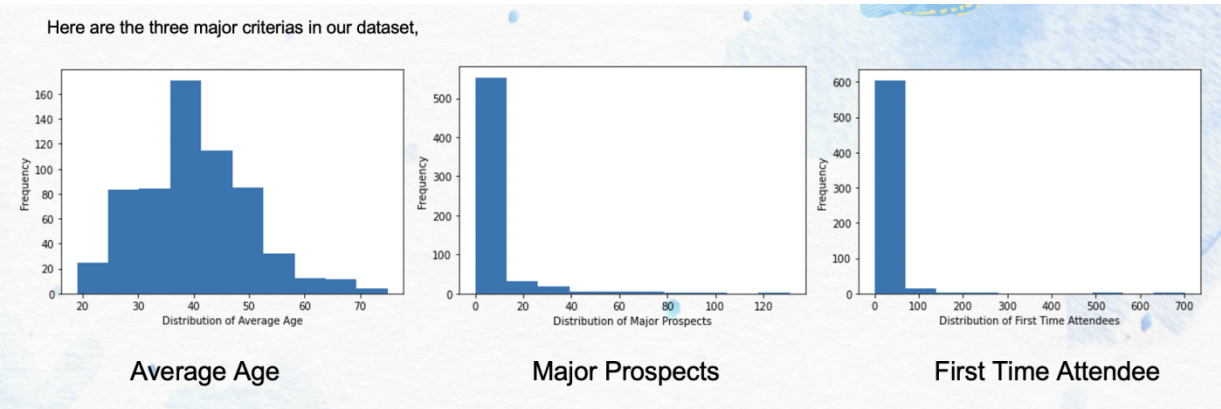
Introducing two new variables to target our visualization and models better:

1. **Regular attendees**: it will give us a no. of people who are committed to the system so that we have a clear picture of the target audience for the first time attendees.
   df['Regular Attendees']= df['Participated']-df['First Time Attendees']
2. **Year:** To have an idea of the yearly trend of the events.
   Extracted from the event date.

Project Testing/Analysis:

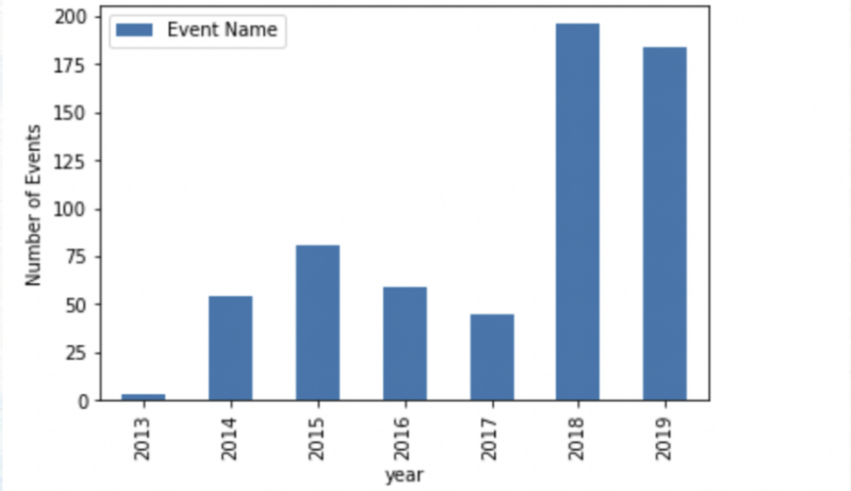
3. Moving forward observing the correlations and the pairwise plots, our target variables are plotted using different functions to get insights into the data.

We analyze the frequency using the histogram first.



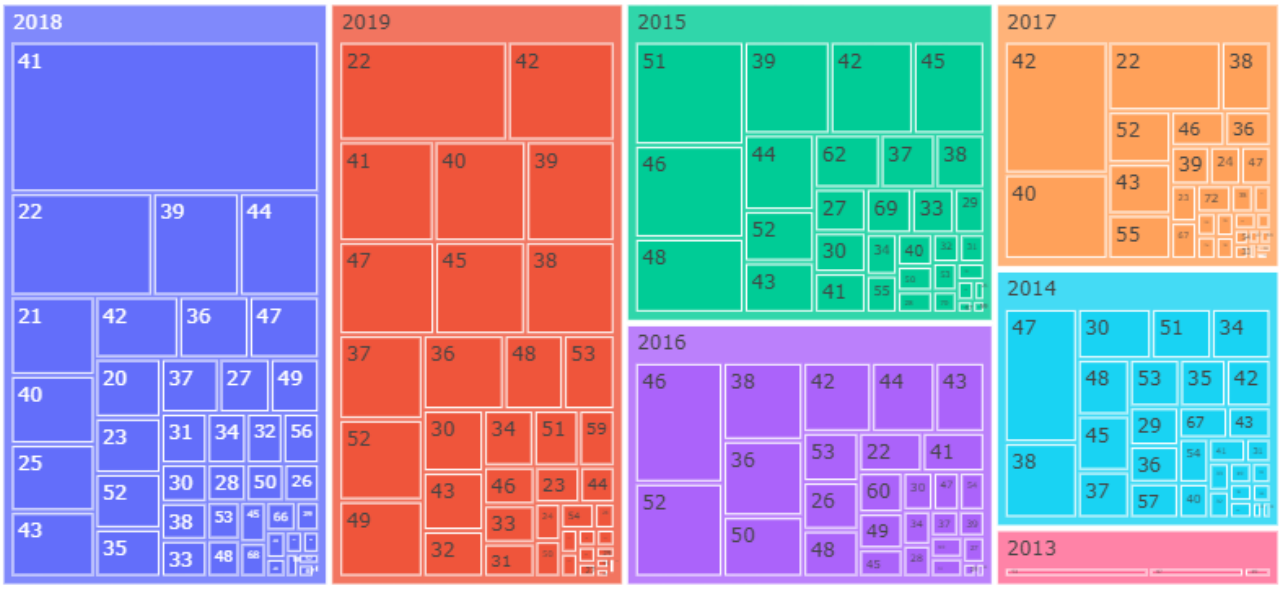
4. We use group by function to count the number of events for each year.

```
df['year']= pd.DatetimeIndex(df['Event Date']).year
event = df.groupby('year').count()
event.reset_index(inplace = True)
event.plot(x = 'year', y = 'Event Name',kind = "bar")
plt.ylabel("Number of Events")
plt.show()
```

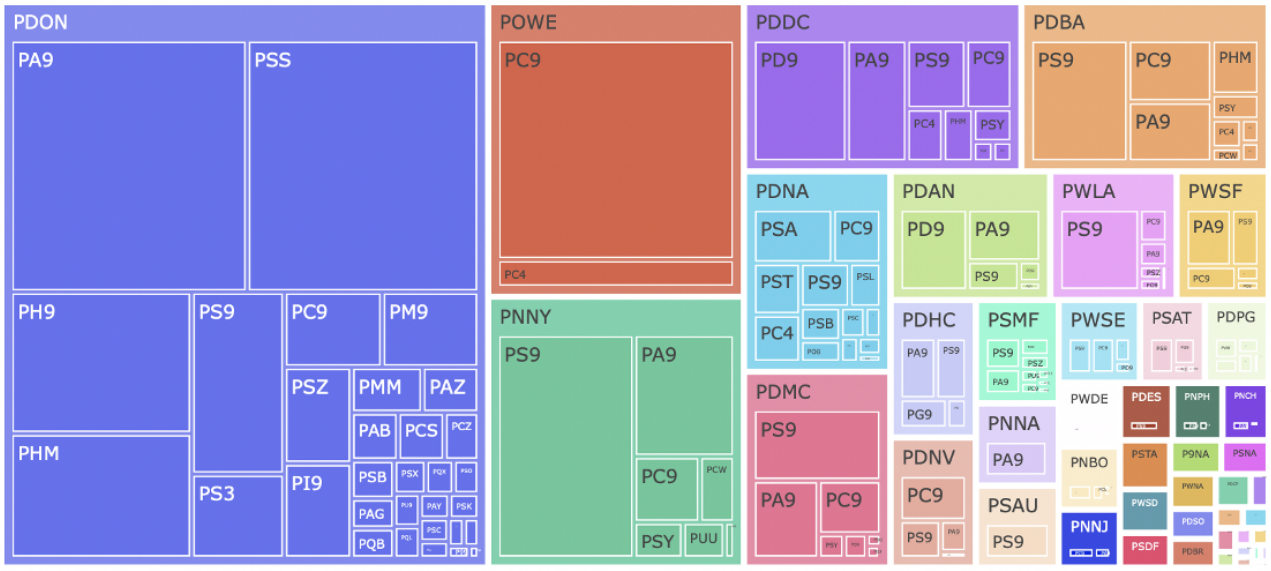


5. To know the number of events conducted each year, to get a timeline of what the progress of the events have been we create a tree map to report numbers of participated each year and their respective average ages using: px.treemap from the plotly.express as px library

.

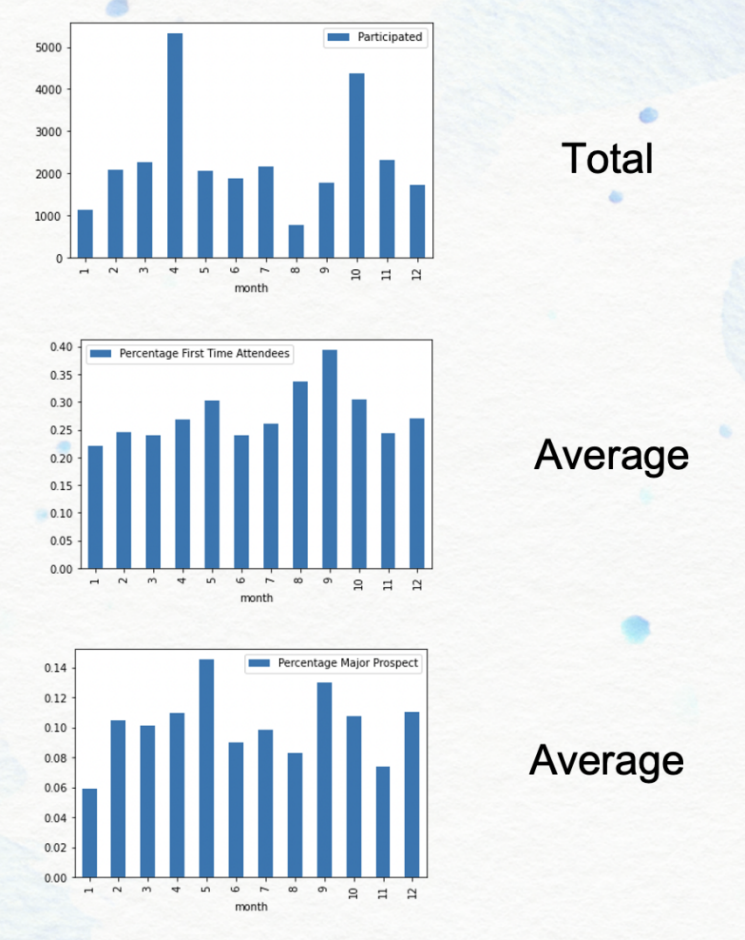


6. As there are a lot of location codes with different events, we use another tree map to show which location has the most attendees and which kind of people would like to attend the association most.

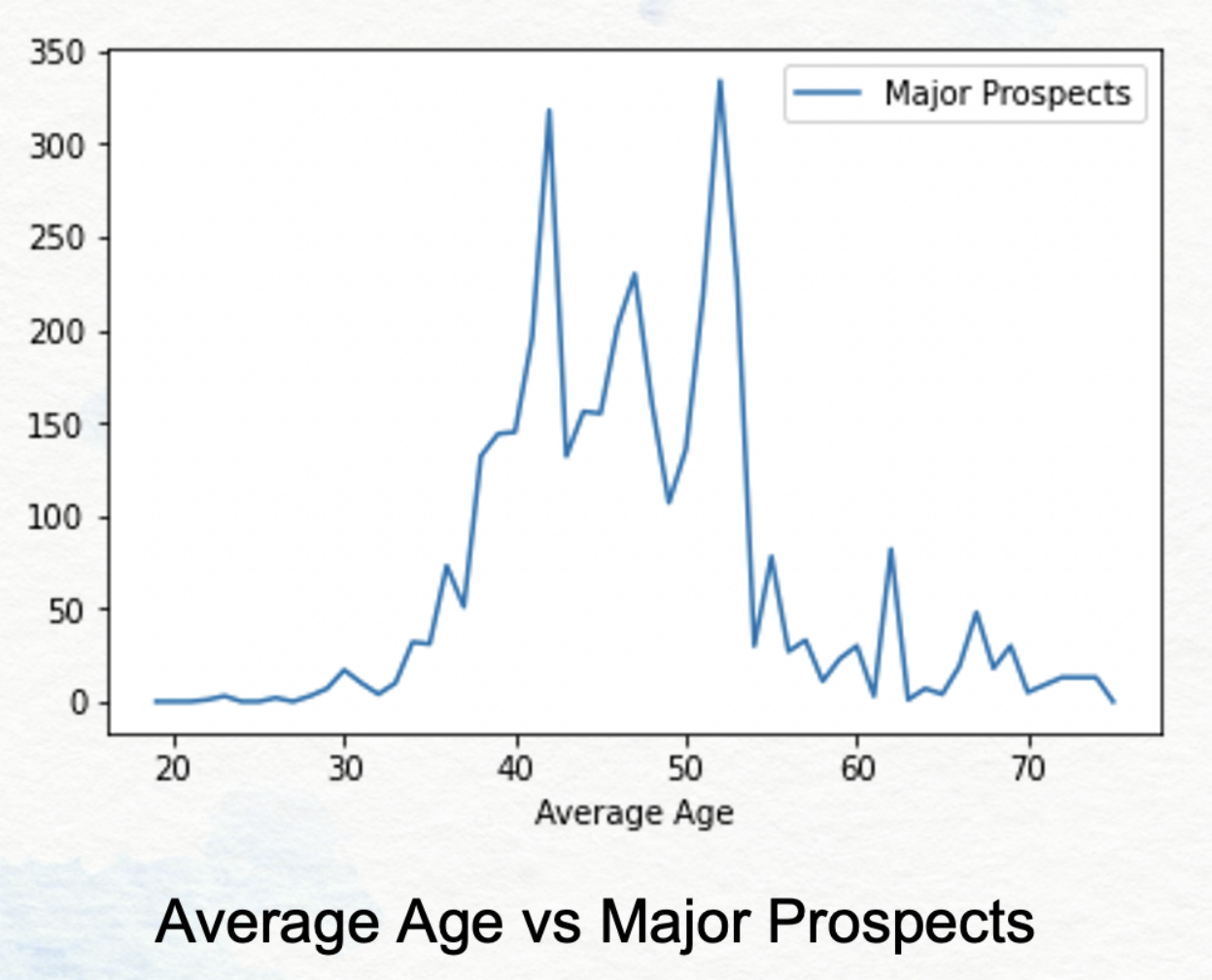


7. Next we visualize the Average number of attendees through the months using the group by function. This is done to target which time of the year should we target to host more events which in-turn will increase our first

time attendees and major prospects. In order to find a best month to hold the event, From total participated and average percentage of first time attendee and percentage of major prospects, percentage of first time attendee and major prospects are the most. We find that October is the best month.



8. Then we analyze the number of prospects using the group by function to know the attendees between 40 to 52 years old would like to donate, and attendees less than 30 years old have less ability to donate.




```
age = df.groupby('Average Age').sum()
age.plot(y = 'Major Prospects')
plt.show()
```

Prediction Models:

Regression:

We ran Six linear regression models using:

- Concatenating independent variables using pandas library concat() fxn
- Using library statsmodel.api for adding constants
- Running the OLS model function from the statsmodel library

- Then fitting the model for the dependent variable Major Prospects for combination of independent variables listed below
- Same for is done for Y= First time attendees
- Reading the summary of these models

REGRESSION MODELS						
Model	**Dependent Variable**	**Independent variable**	**Method**	**R2**	**AIC**	**BIC**
1	Major Prospects	Average age,Participated,year	OLS	0.562	4552	4570
2	Major Prospects	Average age, Participated, year, First time attendees	OLS	0.611	4480	4502
3	First Time Attendees	Age, Participated	OLS	0.705	5633	5646
4	First Time Attendees	Age, Participated,Group code	OLS	0.738	5656	5882
5	First Time Attendees	Age, Regular Attendees	OLS	0.354	6123	6141
6	Major Prospects	Age, Participated, Regular Attendees	OLS	0.554	4564	4582

Model Efficiency:

- Comparing the **R2 values** (Quality of the model)
- Compaing the **AIC** (estimator of prediction error and thereby relative quality of statistical models for a given set of data)
- Comparing the **BIC** values(criterion for model selection among a finite set of models; models with lower BIC are generally preferred)

The categorical variables like group code and location code have too many unique values and don't have much business significance so they were not considered for our analysis.
Furthermore, we ran two more prediction models to analyse the data, Decision Trees and Random Forest.

Decision Trees:

We have used four decision trees to analyze our dependent variables Major Prospects and First Time attendees.

Using the libraries:

sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
sklearn.model_selection import train_test_split #train_test_split function
sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

Data: 70:30:: Train:Test split

Method:

- Concatenating independent variables using pandas library concat() fxn
- Using function:pd.get_dummies(df['Location Code']) for categorical var
- Test and train split using train_test_split
- Using function DecisionTreeClassifier()
- Fitting independent and dependent variables: fit(X_train,y_train)
- Predicting test values: predict(X_test)
- Finding the accuracy of the model: metrics.accuracy_score(y_test, y_pred)

Model Accuracy:

- All the model's accuracy is printed out as shown in the figure below.
- Model with higher accuracy is considered but fewer the variables in the model, better the model.

DECISION TREES			
Model	**Dependent Variable**	**Independent variable**	**Accuracy**
1	Major Prospects	Average age, Regular attendees, Participated	0.411
2	Major Prospects	Average age, Regular attendees, Participated,year	0.411
3	First Time Attendees	Average age, Regular attendees, Major Prospects	0.1764
4	First Time Attendees	Average age, Regular attendees	0.16577

Random Forest Model:

We have used four random forest models to analyse our dependent variables Major Prospects and First Time attendees.

Using the libraries:

from sklearn.ensemble import RandomForestClassifier # Random forest
sklearn.model_selection import train_test_split #train_test_split function
sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

Data: 70:30:: Train:Test split

Method:

- Concatenating independent variables using pandas library concat() fxn
- Using function:pd.get_dummies(df['Location Code']) for categorical var
- Test and train split using train_test_split
- Using function RandomForestClassifier()
- Fitting independent and dependent variables: fit(X_train,y_train)
- Predicting test values: predict(X_test)
- Finding the accuracy of the model: metrics.accuracy_score(y_test, y_pred)

Model Accuracy:

- All the model's accuracy is printed out as shown in the figure below.
- Model with higher accuracy is considered but fewer the variables in the model, better the model.

RANDOM FOREST			
Model	Dependent Variable	Independent variable	Accuracy
1	Major Prospects	Average Age, Regular Attendee	0.417
2	Major Prospects	Average Age, Regular Attendees, Participated, year	0.417
3	First Time Attendees	Average age, Regular attendees, Major Prospects	0.139
4	First Time Attendees	Average age, Regular attendees	0.139

Findings:

Insights from EDA:

- For the distribution of average age, the most attendees are **35-45 years** old totally.
- There are mostly **0-10** major prospects for each events.
- Most of the participants are first time attendees.
- It is obvious that there are more events in **2018 and 2019**.
- The majority of attendees are over **40 years old** in each year except **2019**.
- There are most attendees that participate in events that are held in **PDON, which is on campus**.
- There are more attendees in **group PA9**, which is general athletics.

Insights from Prediction Models:

a) Regression Findings:

- First time attendees are closely related to age and participation.
- Major prospects are related to age, participation, first time attendees, year.

b) Decision Tree Findings:

- Major prospects can be predicted through age and regular attendance.
- Year and participation does not have a major impact in prediction of Major prospects.
- The existing variables are not a good predictor of first time attendees.
- We need more data touch points in order to improve the accuracy of predicting first time attendees.

c) Random Forest Findings:

- Similar to decision trees, Major prospects can be predicted through age and regular attendance.
- Year and participation does not have a major impact in prediction of Major prospects.
- The existing variables are not a good predictor of first time attendees.

In order to improve the overall prediction and accuracy of the models:

1) We can introduce more data touch points like income, industry, family size.
2) We can also do feature engineering on existing dataset like text analysis on existing columns to find unique patterns and then use them for modeling.

Accomplishments:

- Successfully generated insights from the data set using exploratory data analysis.
- Construct data pipelines to simplify data cleansing and model training processes.
- Successfully figured out the factors affecting the turn-up of First-time attendees and Major prospects.
- The key factors that influence First Time Attendees according to the correlation matrix:

  Average Age, year, month, location, group, activity

- The key factors that influence Major Prospects according to the correlation index:

  Year, month, weekday, location, activity

- From a marketing perspective the model seems to work even if the location, group, or activity codes as the events and location for the future events might vary as they are of less significance.
- From EDA using 19 graphs we successfully identified ways to generate more insights and improve the models.
- Discover ways to improve First Time Attendees:

  Events should be held in May, August, September, and October.

  Preferable locations are on campus.

  Preferable types are athletics.

- Discover ways to improve Major Prospects:

Events should be held in May, September and October.

Try to attract alumni of higher age.

- Overall, we can hold more events in September and October.
- Built three predictive models and compared the performances to find merged conclusions
- Able to predict Major prospects and First-Time attendees almost accurately in the future.
- Test varied prediction models to find the ones which modelled the data better and gave higher prediction accuracy.

Suggestion improvements:

- More data in the future could help increase our prediction accuracy.
- The location codes have too many categorical variables, so converting them into smaller regions will be helpful for modelling the data.
- The group code categorical variables are difficult to interpret.
- More event specific data with the names and ages of each customer can help us filter out the Regular attendees and the first-time attendees better.

Thankyou for your time, we hope you like our work.

Team members:

Saswati Mohanty

Yijie Fan

Yunfei Xia

Anurag Sharma

—----------

THE END