



# Ascend Tech Workshop Final Presentation

Team B



# Project Introduction

---

- Learned Python basics throughout Tech Workshop
- Using Python to create a model to analyze data and create a prediction
- Using charts and graphs to describe data
- Determining how likely it is for Ascend members to get Monkey Pox



# Agenda

01. **Project Introduction**

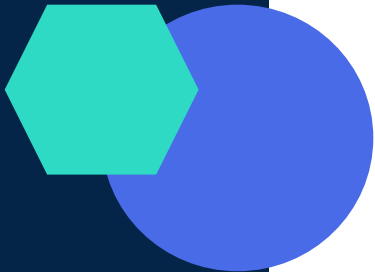
02. **Variable**

03. **Data Visualization**

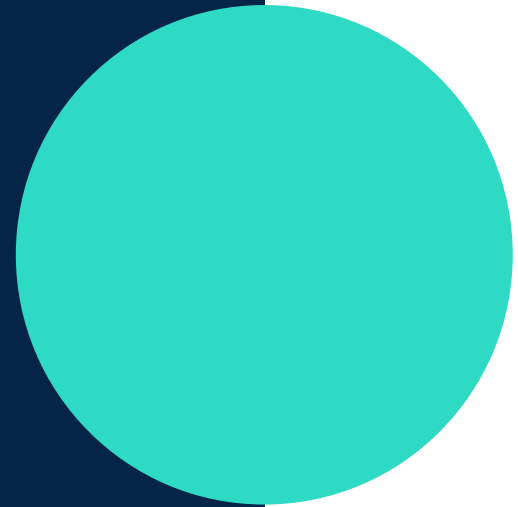
04. **Data Cleansing**

05. **Correlation Plot**

06. **Logistic Regression**



# 02. Variable





# Variable List

## Categorical Variables

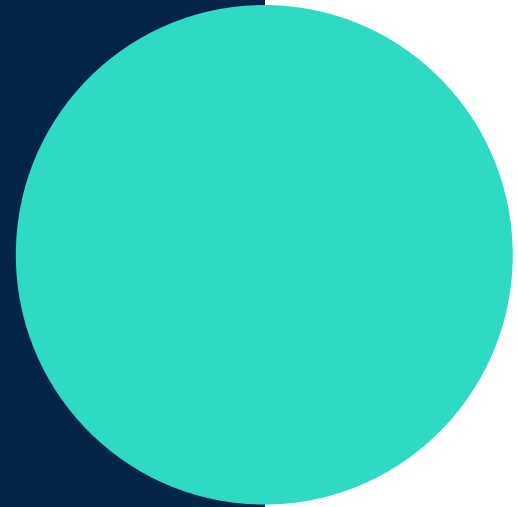
- 1 Rectal Pain
- 2 Sore Throat
- 3 Penile Oedema
- 4 Oral Lesions
- 5 Solitary Lesion
- 6 Swollen Tonsils
- 7 HIV infection
- 8 Sexually Transmitted Disease

## Unused Variables

Systemic illness

4 Values contained

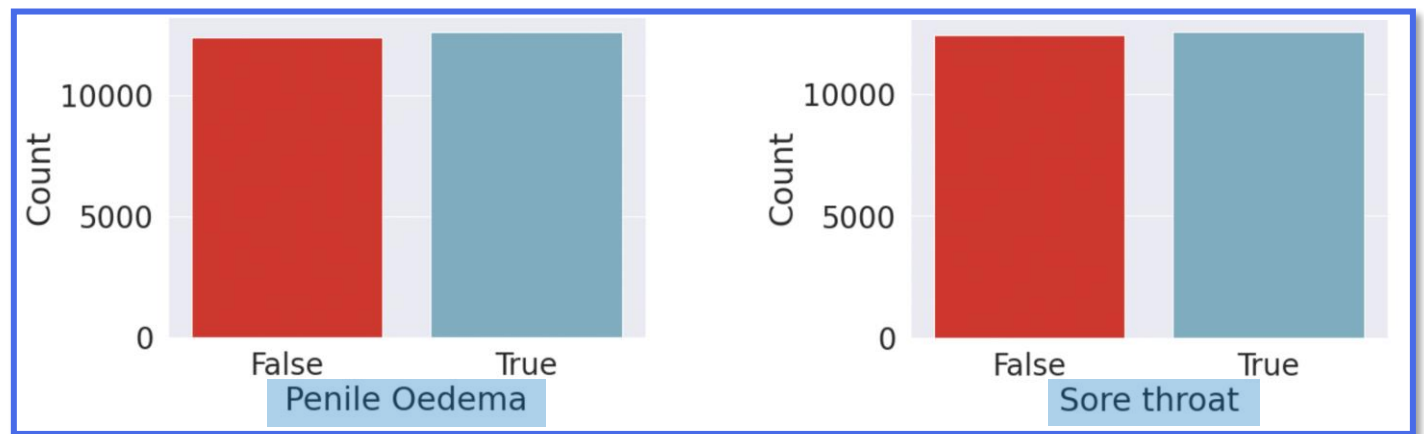
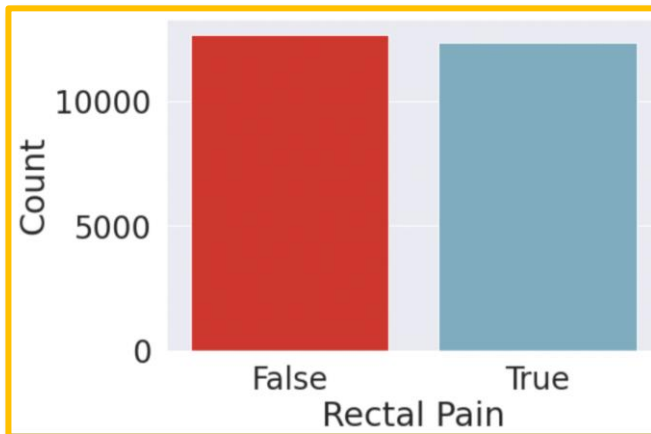
# 03. Data Visualization





# Vertical Bar Plot

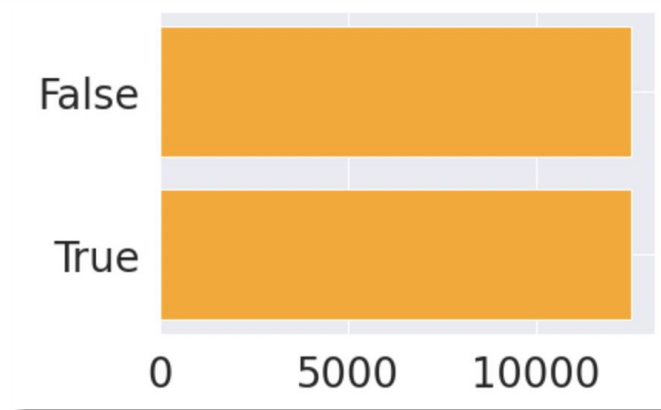
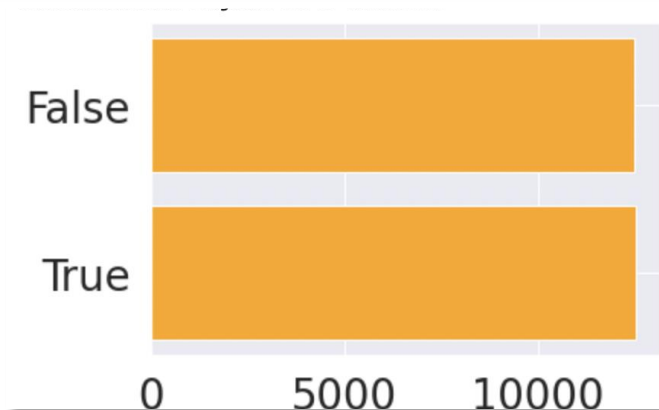
```
## Rectal Pain
RectalPain_unique = pd.unique(data['Rectal Pain'])
RectalPain_unique_count=pd.value_counts(data['Rectal Pain'])
RectalPain_unique_count
RectalPain_dataset = pd.DataFrame(RectalPain_unique_count)
RectalPain_dataset = RectalPain_unique_count.reset_index()
RectalPain_dataset.columns = ["Rectal Pain", "Count"]
RectalPain_dataset
sns.barplot(data=RectalPain_dataset, x="Rectal Pain", y="Count", palette = ['red', 'c'])
```





# Horizontal Bar Plot

```
## Solitary Lesion
Solitary_Lesion_unique = pd.unique(data['Solitary Lesion'])
Solitary_Lesion_unique_count=pd.value_counts(data['Solitary Lesion'])
Solitary_Lesion_unique_count
Solitary_Lesion_dataset = pd.DataFrame(Solitary_Lesion_unique_count)
Solitary_Lesion_dataset = Solitary_Lesion_unique_count.reset_index()
Solitary_Lesion_dataset.columns = ['Solitary Lesion', 'Count']
Solitary_Lesion_dataset
df_ycolumn = ['True', 'False']
plt.figure(figsize=(6,4))
plt.barh(df_ycolumn, Solitary_Lesion_dataset['Count'], color = 'orange')
```







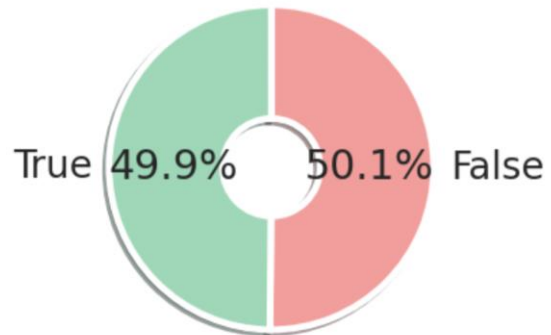
# Pie Chart

```
#Swollen Tonsils
from matplotlib.patches import Shadow
y = pd.value_counts(data['Swollen Tonsils'])
y
labels = ['False', 'True']
colors = ['#ff9999', '#8fd9b6']
wedgeprops={'width': 0.7, 'edgecolor': 'w', 'linewidth': 5}
plt.pie(y, labels = labels, autopct= "%1.1f%%", colors = colors, wedgeprops = wedgeprops, shadow = True, startangle = 90, counterclock = False)

plt.title("Swollen Tonsils")
plt.show
```

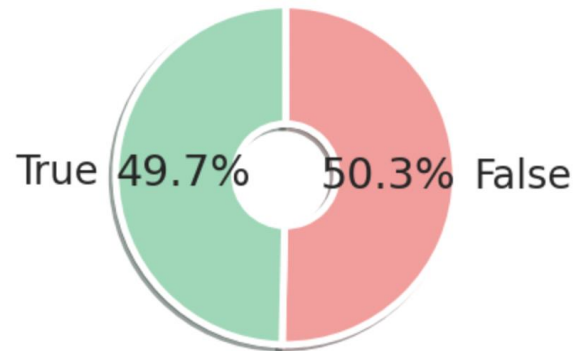
<function matplotlib.pyplot.show(\*args, \*\*kw)>

Swollen Tonsils



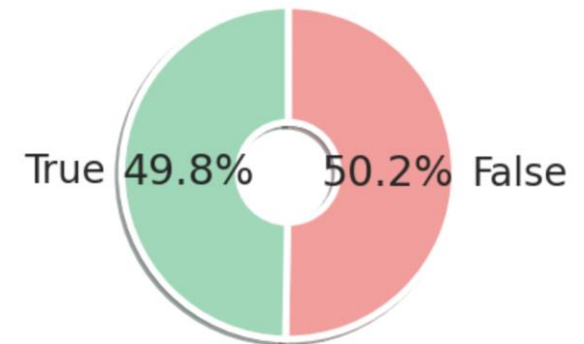
<function matplotlib.pyplot.show(\*args, \*\*kw)>

HIV Infection

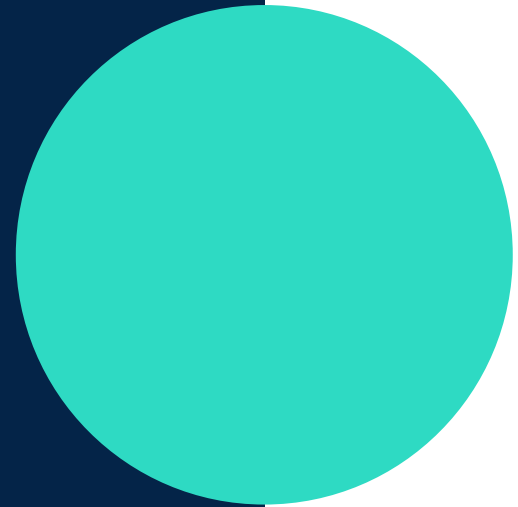


<function matplotlib.pyplot.show(\*args, \*\*kw)>

Sexually Transmitted Infection



# 04. Data Cleansing





# Information of Train Set

Diseases

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 11 columns):
 #   Column                                     Non-Null Count  Dtype
---  -
 0   Patient_ID                               25000 non-null  object
 1   Systemic Illness                         25000 non-null  object
 2   Rectal Pain                             25000 non-null  bool
 3   Sore Throat                             25000 non-null  bool
 4   Penile Oedema                           25000 non-null  bool
 5   Oral Lesions                            25000 non-null  bool
 6   Solitary Lesion                         25000 non-null  bool
 7   Swollen Tonsils                         25000 non-null  bool
 8   HIV Infection                           25000 non-null  bool
 9   Sexually Transmitted Infection          25000 non-null  bool
10  MonkeyPox                               25000 non-null  object
dtypes: bool(8), object(3)
memory usage: 781.4+ KB
```



# Information of Train Set

Train Set

25000

df\_train.head(10)



Test Set

16

df\_test.head(10)

df\_train.head(10)

	Patient_ID	Systemic Illness	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection	MonkeyPox
0	P0	None	False	True	True	True	False	True	False	False	Negative
1	P1	Fever	True	False	True	True	False	False	True	False	Positive
2	P2	Fever	False	True	True	False	False	False	True	False	Positive
3	P3	None	True	False	False	False	True	True	True	False	Positive
4	P4	Swollen Lymph Nodes	True	True	True	False	False	True	True	False	Positive
5	P5	Swollen Lymph Nodes	False	True	False	False	False	False	False	False	Negative
6	P6	Fever	False	True	False	False	False	False	True	False	Positive
7	P7	Fever	True	True	False	True	True	True	False	False	Positive
8	P8	Muscle Aches and Pain	False	True	True	True	False	False	False	False	Positive
9	P9	Fever	False	False	True	True	True	False	True	False	Negative

df\_test.head(10)

	Patient_ID	Systemic Illness	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection	MonkeyPox
0	Juhui Yun	False	True	False	True	True	True	False	False	False	False
1	Bhavya	False	True	True	False	True	False	True	False	False	False
2	Kiyun Oh	True	True	True	True	True	True	True	True	False	False
3	Daniel Nguyen	False	True	False	True	False	True	False	True	False	False
4	Chris Guo	True	True	True	True	True	True	True	True	False	False
5	Clara	True	False	True	False	False	True	True	False	False	False
6	Drake	False	True	False	False	True	False	True	False	False	False
7	Kevin Wu	False	False	True	True	False	True	False	False	False	False
8	Aileen	False	False	False	False	False	True	True	True	False	False
9	Jason	False	True	False	True	False	True	False	True	False	False



# Data Cleansing> Column

Delete

Move to first column

	Patient_ID	Systemic Illness	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection	MonkeyPox
0	P0	None	False	True	True	True	False	True	False	False	Negative
1	P1	Fever	True	False	True	True	False	False	True	False	Positive
2	P2	Fever	False	True	True	False	False	False	True	False	Positive
3	P3	None	True	False	False	False	True	True	True	False	Positive
4	P4	Swollen Lymph Nodes	True	True	True	False	False	True	True	False	Positive
5	P5	Swollen Lymph Nodes	False	True	False	False	False	False	False	False	Negative
6	P6	Fever	False	True	False	False	False	False	True	False	Positive
7	P7	Fever	True	True	False	True	True	True	False	False	Positive
8	P8	Muscle Aches and Pain	False	True	True	True	False	False	False	False	Positive
9	P9	Fever	False	False	True	True	True	False	True	False	Negative

```
df_train['Systemic Illness'].value_counts()
```

```
Fever          6382
Swollen Lymph Nodes  6252
None           6216
Muscle Aches and Pain  6150
Name: Systemic Illness, dtype: int64
```

```
df_train['MonkeyPox'].value_counts()
```

```
Positive    15909
Negative     9091
Name: MonkeyPox, dtype: int64
```



# Data Cleansing > Column

Column sequence conversion

```
df_train = df_train[['MonkeyPox', 'Patient_ID', 'Systemic Illness', 'Rectal Pain', 'Sore Throat',  
                    'Penile Oedema', 'Oral Lesions', 'Solitary Lesion', 'Swollen Tonsils',  
                    'HIV Infection', 'Sexually Transmitted Infection']]  
df_test = df_test[['MonkeyPox', 'Patient_ID', 'Systemic Illness', 'Rectal Pain', 'Sore Throat',  
                  'Penile Oedema', 'Oral Lesions', 'Solitary Lesion', 'Swollen Tonsils',  
                  'HIV Infection', 'Sexually Transmitted Infection']]  
train = df_train.drop(['Patient_ID', 'Systemic Illness'], axis = 1)  
test = df_test.drop(['Patient_ID', 'Systemic Illness'], axis = 1)
```

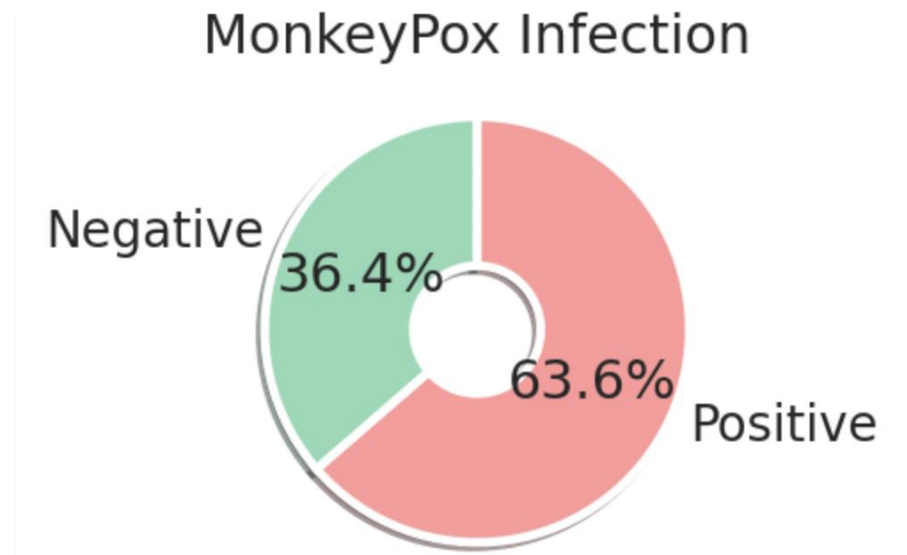
Column Delete



# Train Set > Monkey Pox

```
from matplotlib.patches import Shadow
y = pd.value_counts(df_train['MonkeyPox'])
y
labels = ['Positive', 'Negative']
colors = ['#ff9999', '#8fd9b6']
wedgeprops={'width': 0.7, 'edgecolor': 'w', 'linewidth': 5}
plt.pie(y, labels = labels, autopct= "%1.1f%%", colors = colors, wedgeprops = wedgeprops, shadow = True, startangle = 90, counterclock = False)

plt.title("MonkeyPox Infection")
plt.show
```







# Mapping

## Categorical Variable

```
Truth_mapping = {False: 0, True: 1}
train['Rectal Pain'] = train['Rectal Pain'].map(Truth_mapping)
train['Sore Throat'] = train['Sore Throat'].map(Truth_mapping)
train['Penile Oedema'] = train['Penile Oedema'].map(Truth_mapping)
train['Oral Lesions'] = train['Oral Lesions'].map(Truth_mapping)
train['Solitary Lesion'] = train['Solitary Lesion'].map(Truth_mapping)
train['Swollen Tonsils'] = train['Swollen Tonsils'].map(Truth_mapping)
train['HIV Infection'] = train['HIV Infection'].map(Truth_mapping)
train['Sexually Transmitted Infection'] = train['Sexually Transmitted Infection'].map(Truth_mapping)
test['Rectal Pain'] = test['Rectal Pain'].map(Truth_mapping)
test['Sore Throat'] = test['Sore Throat'].map(Truth_mapping)
test['Penile Oedema'] = test['Penile Oedema'].map(Truth_mapping)
test['Oral Lesions'] = test['Oral Lesions'].map(Truth_mapping)
test['Solitary Lesion'] = test['Solitary Lesion'].map(Truth_mapping)
test['Swollen Tonsils'] = test['Swollen Tonsils'].map(Truth_mapping)
test['HIV Infection'] = test['HIV Infection'].map(Truth_mapping)
test['Sexually Transmitted Infection'] = test['Sexually Transmitted Infection'].map(Truth_mapping)
```

*{False: 0, True : 1}*

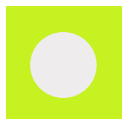
## Monkey Pox

```
Negative_mapping = {"Negative": 0, "Positive": 1}
train['MonkeyPox'] = train['MonkeyPox'].map(Negative_mapping)
test["MonkeyPox"] = test["MonkeyPox"].map(Truth_mapping)

train
test
```

*{"Negative":0, "Postive":1}*





# Mapping Result

Train Set

25000

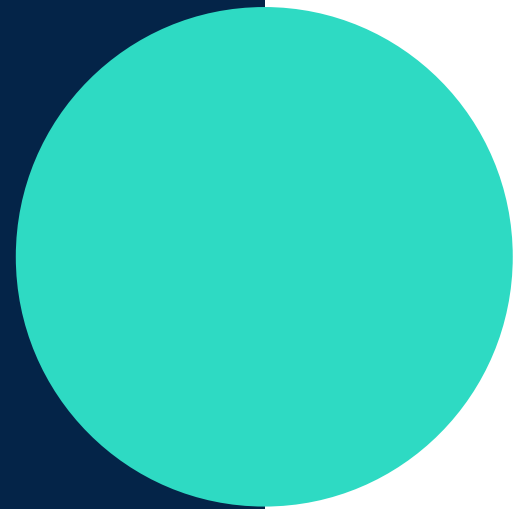
Test Set

16

	MonkeyPox	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection
0	0	0	1	1	1	0	1	0	0
1	1	1	0	1	1	0	0	1	0
2	1	0	1	1	0	0	0	1	0
3	1	1	0	0	0	1	1	1	0
4	1	1	1	1	0	0	1	1	0
...	...	...	...	...	...	...	...	...	...
24995	1	1	1	0	1	1	0	0	1
24996	1	0	1	1	0	1	1	1	1
24997	1	1	1	0	0	1	1	0	0
24998	0	0	1	0	1	1	1	0	0
24999	1	0	0	1	0	0	1	1	0

	MonkeyPox	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection
0	0	1	0	1	1	1	0	0	0
1	0	1	1	0	1	0	1	0	0
2	0	1	1	1	1	1	1	1	0
3	0	1	0	1	0	1	0	1	0
4	0	1	1	1	1	1	1	1	0
5	0	0	1	0	0	1	1	0	0
6	0	1	0	0	1	0	1	0	0
7	0	0	1	1	0	1	0	0	0
8	0	0	0	0	0	1	1	1	0
9	0	1	0	1	0	1	0	1	0
10	0	0	1	0	0	0	0	0	0
11	0	0	0	1	0	0	1	0	0
12	0	0	1	0	0	0	1	0	0
13	0	0	1	0	0	0	0	0	0
14	0	0	1	0	0	0	1	0	0
15	0	0	1	0	0	0	0	0	0

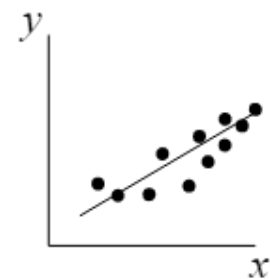
# 05. Correlation Plot



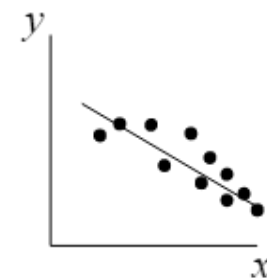


# Correlation Table

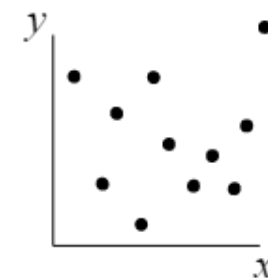
```
cormat= train.corr()  
round(cormat,2)
```



Positive



Negative



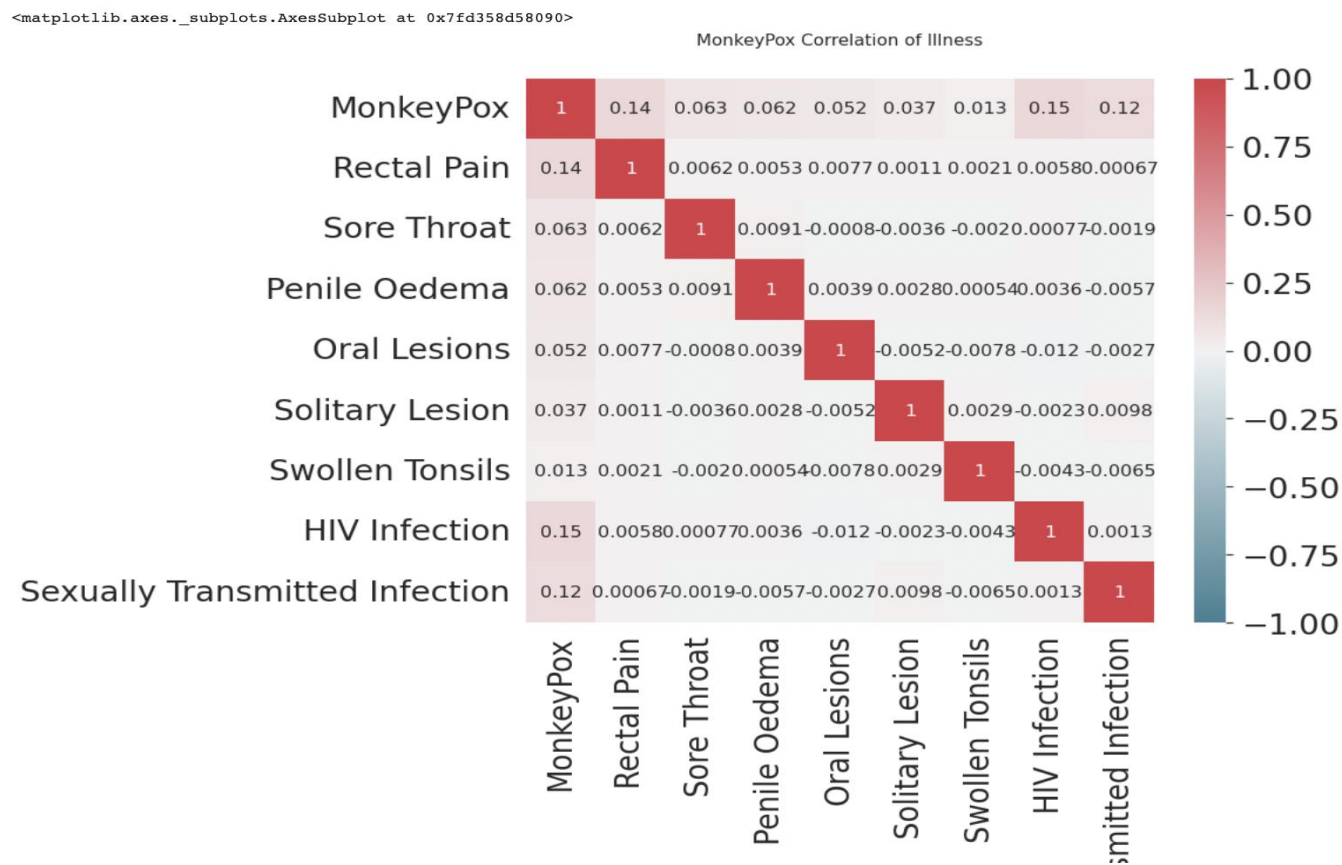
No correlation

	MonkeyPox	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection
MonkeyPox	1.00	0.14	0.06	0.06	0.05	0.04	0.01	0.15	0.12
Rectal Pain	0.14	1.00	0.01	0.01	0.01	0.00	0.00	0.01	0.00
Sore Throat	0.06	0.01	1.00	0.01	-0.00	-0.00	-0.00	0.00	-0.00
Penile Oedema	0.06	0.01	0.01	1.00	0.00	0.00	0.00	0.00	-0.01
Oral Lesions	0.05	0.01	-0.00	0.00	1.00	-0.01	-0.01	-0.01	-0.00
Solitary Lesion	0.04	0.00	-0.00	0.00	-0.01	1.00	0.00	-0.00	0.01
Swollen Tonsils	0.01	0.00	-0.00	0.00	-0.01	0.00	1.00	-0.00	-0.01
HIV Infection	0.15	0.01	0.00	0.00	-0.01	-0.00	-0.00	1.00	0.00
Sexually Transmitted Infection	0.12	0.00	-0.00	-0.01	-0.00	0.01	-0.01	0.00	1.00

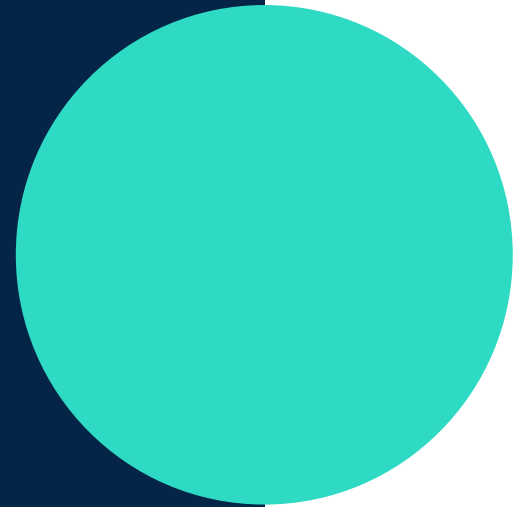


# Correlation Heat Map

```
plt.figure(figsize = (12, 10))
plt.title("MonkeyPox Correlation of Illness", y=1.05, size = 15)
sns.heatmap(cormat, annot=True, annot_kws={"size":16}, cmap=sns.diverging_palette(220, 10, as_cmap=True), vmin=-1, vmax=1)
```

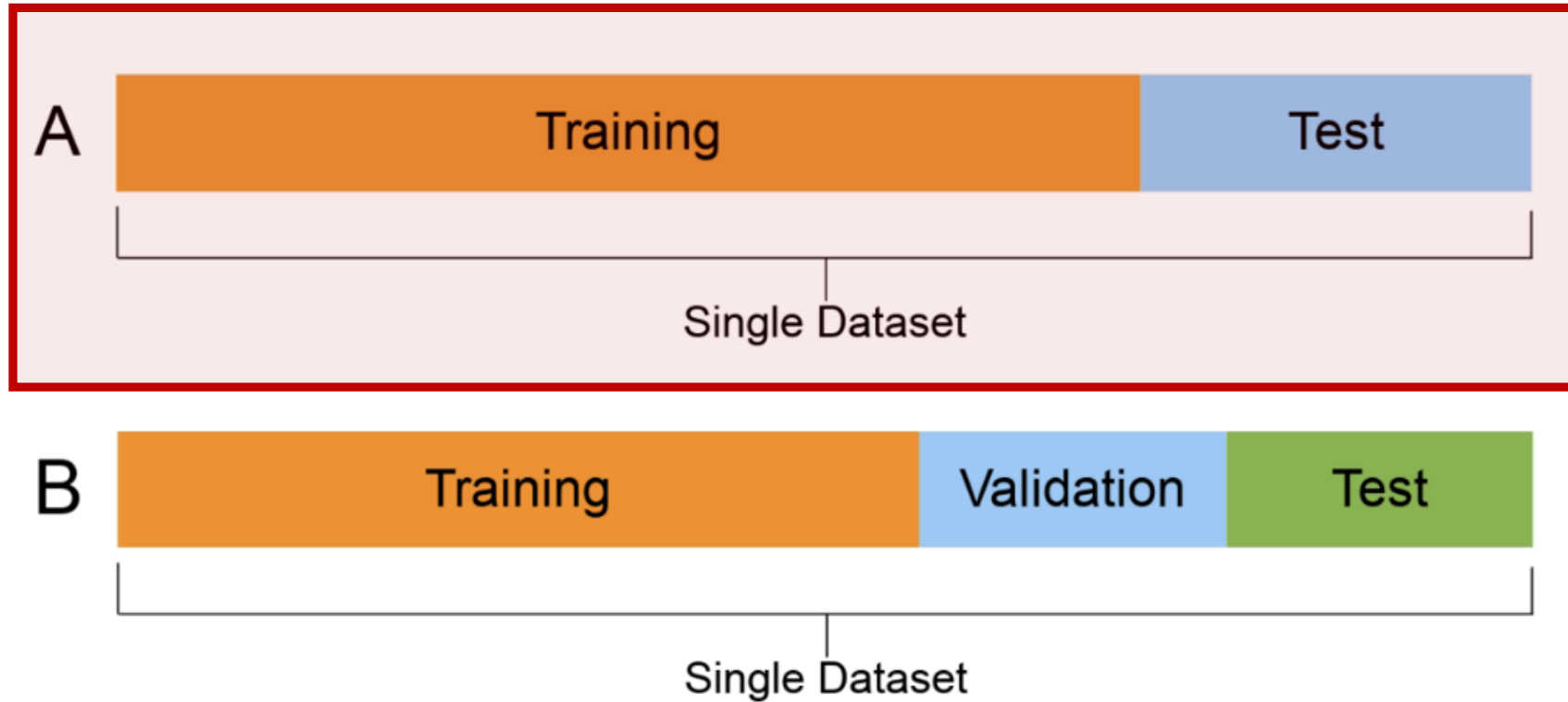


# 06. Logistic Regression





# Splitting the Data





# Creating Regression Model

```
data= pd.concat([train, test])
```

```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
from sklearn.model_selection import train_test_split
```

```
X_train = data.drop('MonkeyPox', axis=1).values
target_label = data['MonkeyPox'].values
X_test = train
```

```
X_train.shape, X_test.shape
```

```
((25016, 8), (25000, 9))
```

```
X_tr, X_vld, y_tr, y_vld = train_test_split(X_train, target_label, test_size=0.0006, shuffle = False) #Adjust test size
y_tr.shape, y_vld.shape
```

```
((25000,), (16,))
```

```
prediction
```

```
array([1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0])
```

```
print('Number of people: {} \naccuracy: {:.2f}% '.format(y_vld.shape[0], 100 * accuracy_score(y_vld,prediction)))
```

```
Number of people: 16
accuracy: 50.00%
```

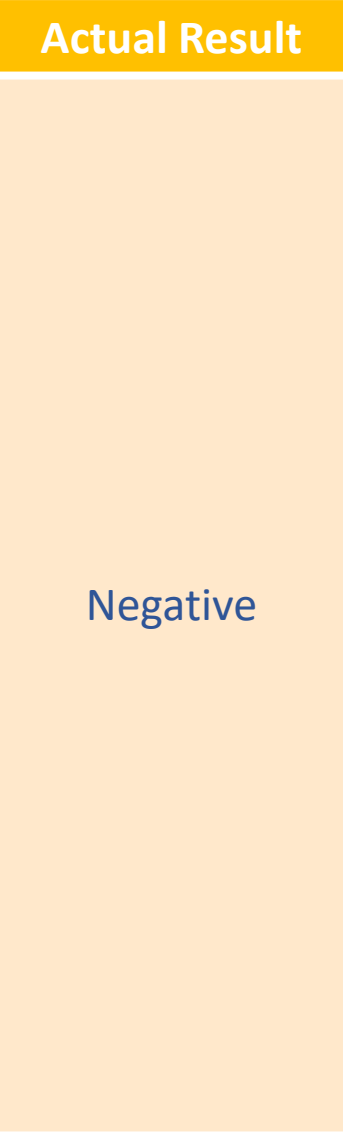


**Splitting  
Formula**

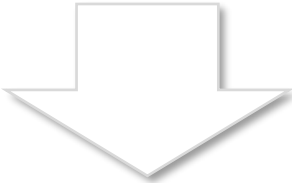


# Regression Model Result

	Name	Prediction Result
1	Juhui Yun	Positive
2	Bhavya	Positive
3	Kiyun Oh	Positive
4	Daniel Nguyen	Positive
5	Chris Guo	Positive
6	Clara	Negative
7	Drake	Positive
8	Kevin Wu	Negative
9	Alleen	Positive
10	Jason	Positive
11	Jasmine	Negative
12	Bob	Negative
13	Johnny Lee	Negative
14	Nandini	Negative
15	Aaron	Negative
16	Seungeun Yi	Negative



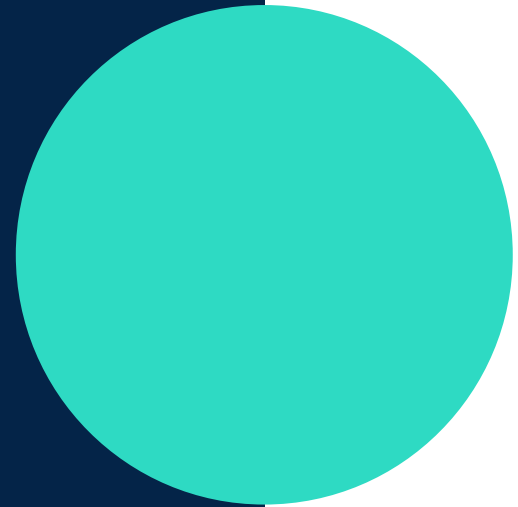
8 Negative  
8 Positive



50% Accuracy



# Wrap Up





# Process Overview

Data Visualization

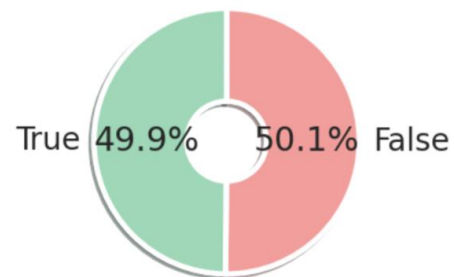
Data Cleansing

Correlation Plot

Logistic Regression

```
<function matplotlib.pyplot.show(*args, **kw)>
```

## Swollen Tonsils



Delete

Patient\_ID

Asymptomatic Illness

Rectal Pain

Sore Throat

Penile Oedema

Oral Lesions

Solitary Lesion

Swollen Tonsils

HIV Infection

Sexually Transmitted Infection

Monkeypox

0	P0	None	False	True	True	False	True	False	False	Negative	
1	P1	Fever	True	False	True	True	False	False	True	False	Positive
2	P2	Fever	False	True	True	False	False	False	True	False	Positive
3	P3	None	True	False	False	False	True	True	True	False	Positive
4	P4	Swollen Lymph Nodes	True	True	True	False	False	True	True	False	Positive
5	P5	Swollen Lymph Nodes	False	True	False	False	False	False	False	False	Negative
6	P6	Fever	False	True	False	False	False	False	True	False	Positive
7	P7	Fever	True	True	False	True	True	True	False	False	Positive
8	P8	Head Aches and Pain	False	True	True	True	False	False	False	False	Positive
9	P9	Fever	False	False	True	True	False	True	False	False	Negative

→ reason eliminating

df\_train['Asymptomatic Illness'].value\_counts()

Fever: 4362

Swollen Lymph Nodes: 6252

None: 4214

Head Aches and Pain: 4133

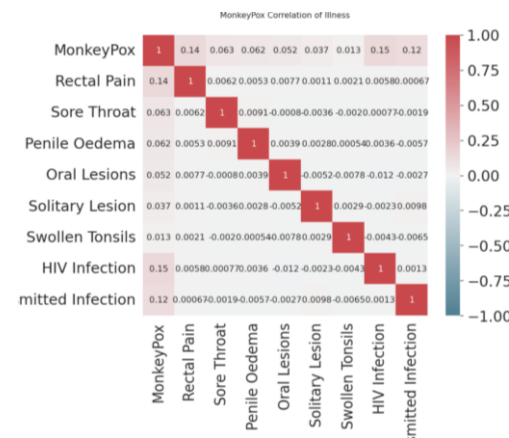
Name: Asymptomatic Illness, dtype: int64

df\_train['MonkeyPox'].value\_counts()

Positive: 15909

Negative: 9091

Name: MonkeyPox, dtype: int64



	Name	Prediction Result	Actual Result
1	Juhui Yun	Positive	Negative
2	Bhavya	Positive	
3	Kiyun Oh	Positive	
4	Daniel Nguyen	Positive	
5	Chris Guo	Positive	
6	Clara	Negative	
7	Drake	Positive	
8	Kevin Wu	Negative	
9	Alleen	Positive	
10	Jason	Positive	
11	Jasmine	Negative	
12	Bob	Negative	
13	Johnny Lee	Negative	
14	Nandini	Negative	
15	Aaron	Negative	
16	Seungeun Yi	Negative	