Website traffic analysis

Phase 4:

Topic: Use IBM Cognos to create interactive dashboards and reports that display insights such as popular pages, traffic sources, and user engagement metrics.

- Use Python libraries like Pandas and Matplotlib to perform more complex analyses on the data, such as time series analysis, user segmentation, or machine learning-based predictions.

Introduction:

- Website traffic refers to the volume of visitors or users who access a particular website during a given period. It is a fundamental metric in web analytics, providing valuable insights into a website's performance and the behavior of its audience. Understanding website traffic is essential for businesses, organizations, and individuals seeking to optimize their online presence and achieve various goals, such as increasing brand visibility, driving conversions, or improving user experience.

Program:

```
import numpy as np

import pandas as pd
import pandas_profiling
import warnings
warnings.filterwarnings('ignore')
import datetime
from datetime import date
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set_style("whitegrid")
# import chart_studio.plotly as py
import cufflinks as cf
import plotly.express as px
from plotly.offline import download_plotlyjs,
init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
cf.go_offline()
import pandas_profiling
import plotly.graph_objects as go
from sklearn.model_selection import train_test_split,
cross_val_score, GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.svm import SVR
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
```

```
import xgboost as xg

df=pd.read_csv('../input/daily-website-visitors/daily-website-visitors.csv')

df.rename(columns = {'Day.Of.Week':'day_of_week'

            ,'Page.Loads':'page_loads'

            ,'Unique.Visits':'unique_visits'

            ,'First.Time.Visits':'first_visits'

            ,'Returning.Visits':'returning_visits'}, inplace = True)

df=df.replace(',','',regex=True)

df['page_loads']=df['page_loads'].astype(int)

df['unique_visits']=df['unique_visits'].astype(int)

df['first_visits']=df['first_visits'].astype(int)

df['returning_visits']=df['returning_visits'].astype(int)

df
```

out 1:

```
Row   Day   day_of_week   Date   page_loads   unique_visit   first_visits

0     1     Sunday        1      9/14/2014    2146   1582   1430   152
1     2     Monday        2      9/15/2014    3621   2528   2297   231
2     3     Tuesday       3      9/16/2014    3698   2630   2352   278
3     4     Wednesday     4      9/17/2014    3667   2614   2327   287
4     5     Thursday      5      9/18/2014    3316   2366   2130   236
...   ...   ...   ...     ...    ...    ...   ...    ...
2162  2163  Saturday      7      8/15/2020    2221   1696   1373   323
2163  2164  Sunday        1      8/16/2020    2724   2037   1686   351
```

In 2:

```
df.isna().sum()
```

out 2:

```
Row                0
Day               0
day_of_week        0
Date              0
page_loads         0
unique_visits      0
first_visits       0
returning_visits   0
dtype: int64
```

In 3:

```
df.duplicated().sum()
```

out 3:

```
0
```

In 4:

```
df.info()
```

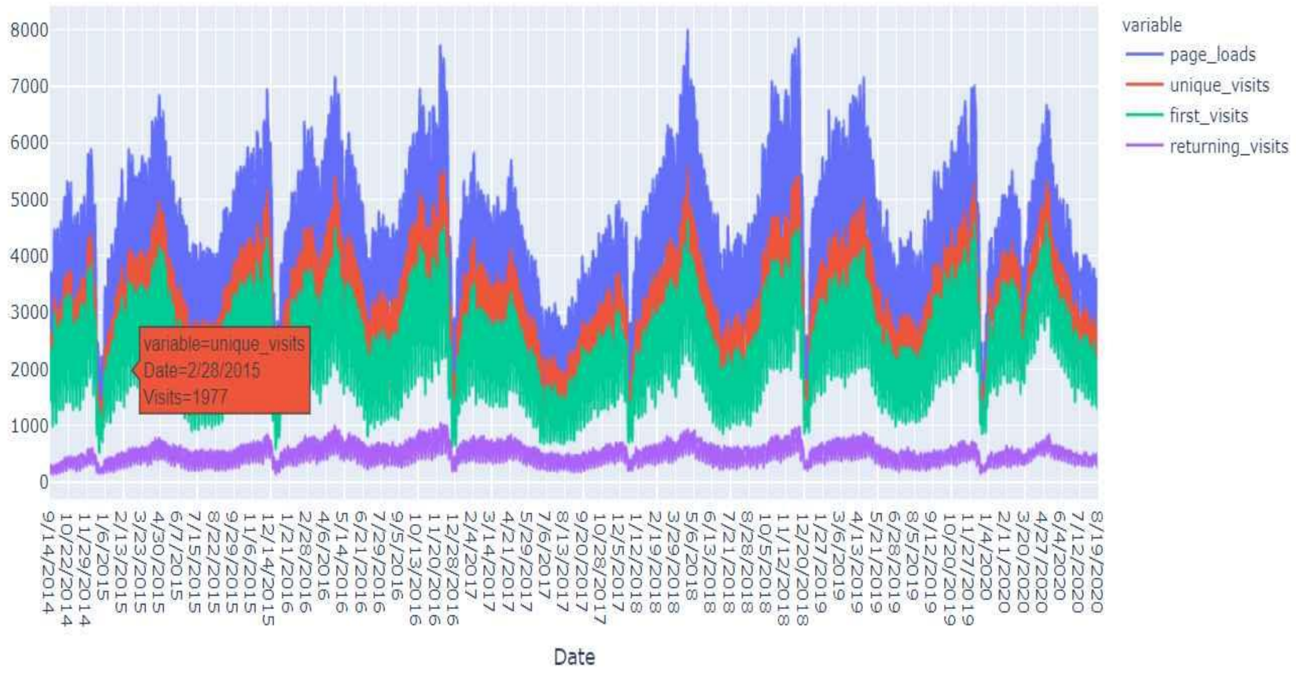
out 4:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2167 entries, 0 to 2166
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Row               2167 non-null   int64
 1   Day               2167 non-null   object
 2   day_of_week       2167 non-null   int64
 3   Date              2167 non-null   object
 4   page_loads        2167 non-null   int64
 5   unique_visits     2167 non-null   int64
 6   first_visits      2167 non-null   int64
 7   returning_visits  2167 non-null   int64
dtypes: int64(6), object(2)
memory usage: 135.6+ KB
```

In 5:

```
px.line(df,x='Date',y=['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits'],
        labels={'value':'Visits'}
        ,title='Page Loads & visitors over Time')
```

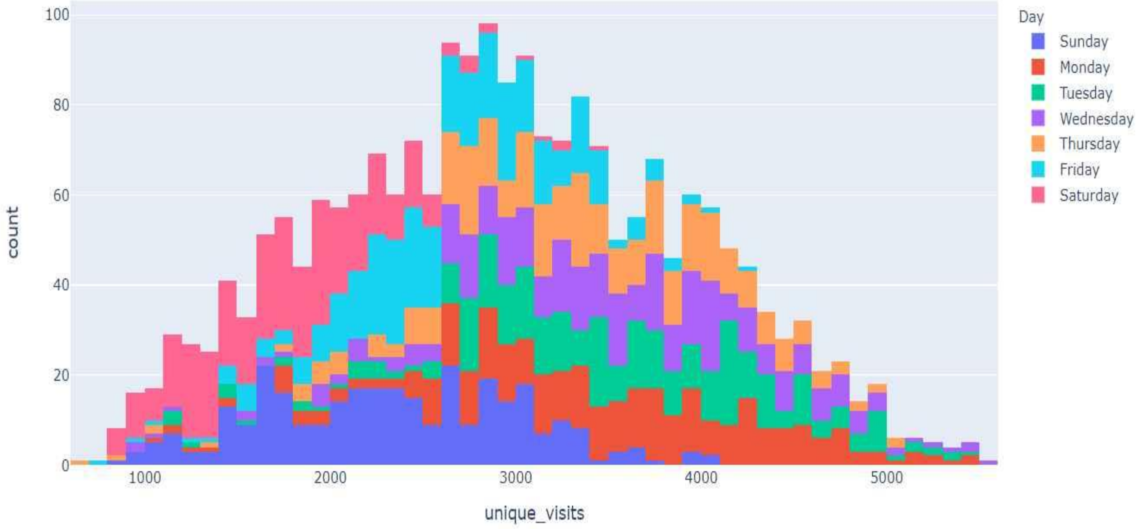
Out 5:



In 6:

```
px.histogram(df,x='unique_visits',color='Day',title='unique visits for each day')
```

out 6:

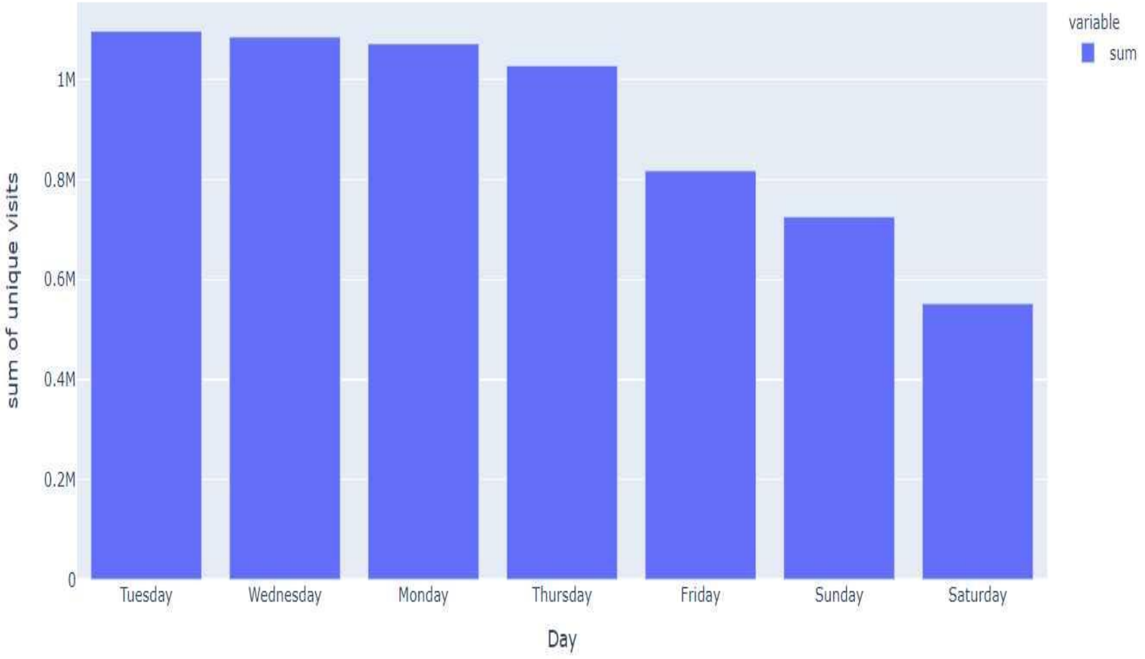


In 7:

```
day_imp=df.groupby(['Day'])['unique_visits'].agg(['sum']).sort_values(by='sum'
,ascending=False)
```

```
px.bar(day_imp,labels={'value':'sum of unique visits'},title='Sum of Unique
visits for each day')
```

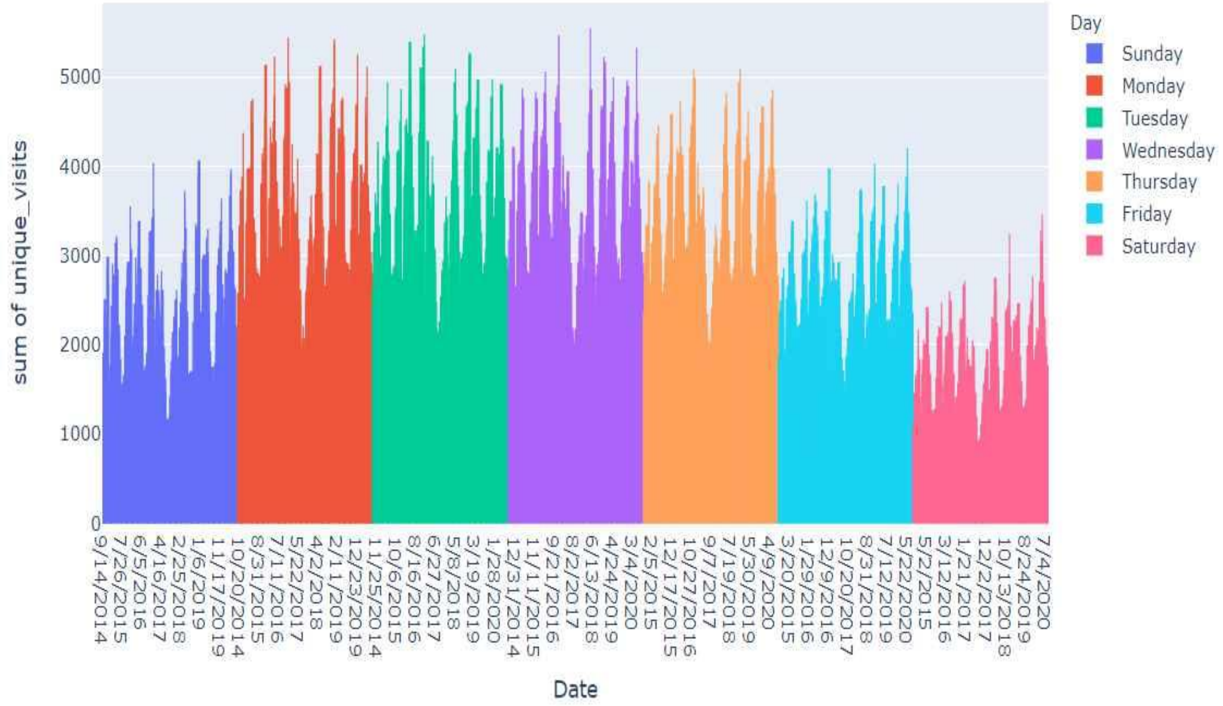
out 7:



In 8:

```
px.histogram(df,x='Date',y='unique_visits',color='Day',title='Sum of unique visits for e
ach day over Time')
```

out 8:



In 9:

```
sums=df.groupby(['Day'])[['page_loads' ,'unique_visits'
,'first_visits' ,'returning_visits']].sum().sort_values(
    by='unique_visits',ascending=False)
sums
```

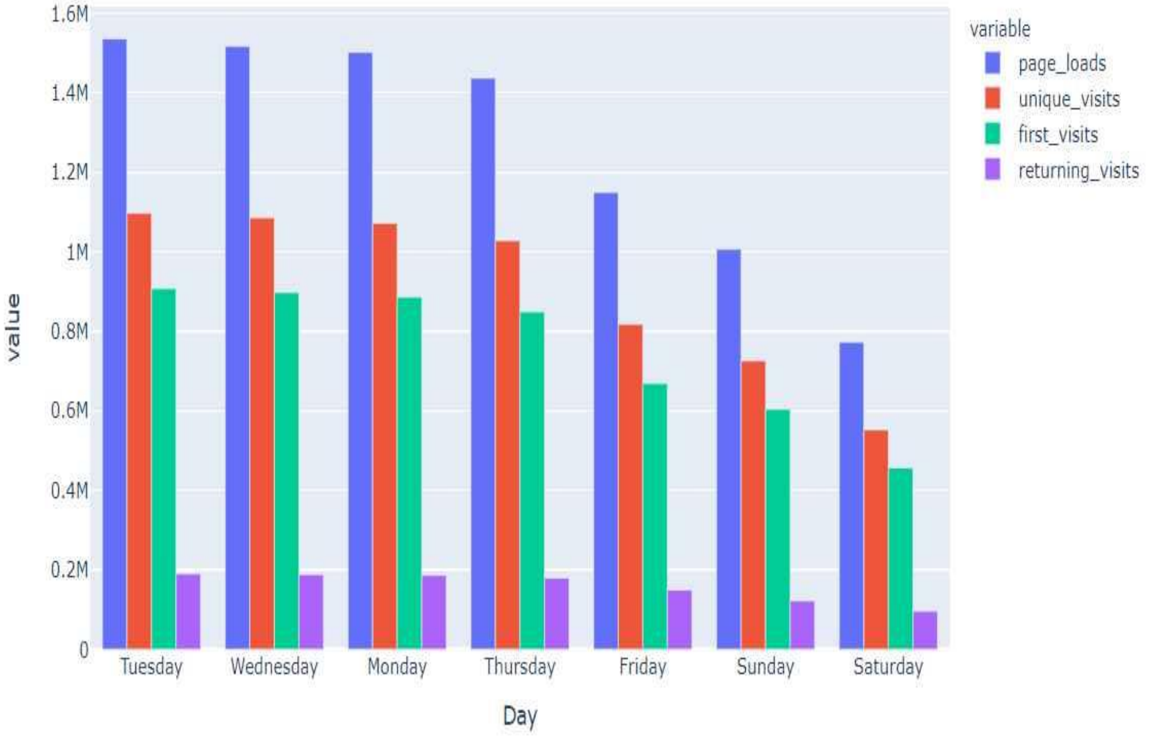
out 9:

page_loads	unique_visits	first_visits	returning_visits
Day			
Tuesday	1536154	1097181	907752
Wednesday	1517114	1085624	897602

In 10:

```
px.bar(sums,barmode='group',title='Sum of page loads and visits for each of their da
ys')
```

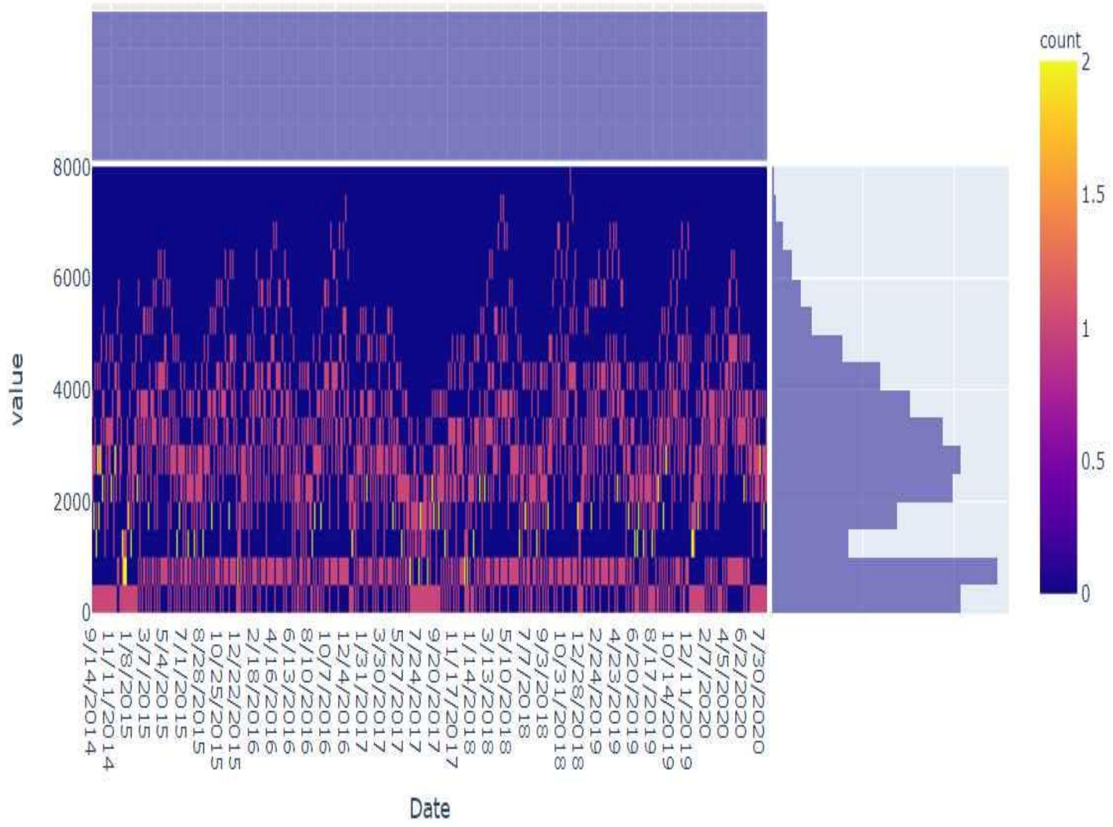
out 10:



In 11:

```
px.density_heatmap(df, x='Date',y=['page_loads' ,'unique_visits' ,'first_visits' ,'returni
ng_visits']
                  color_continuous_scale="Viridis"
                 ,marginal_x="histogram", marginal_y="histogram",title='Correlation for ea
ch data point')
```

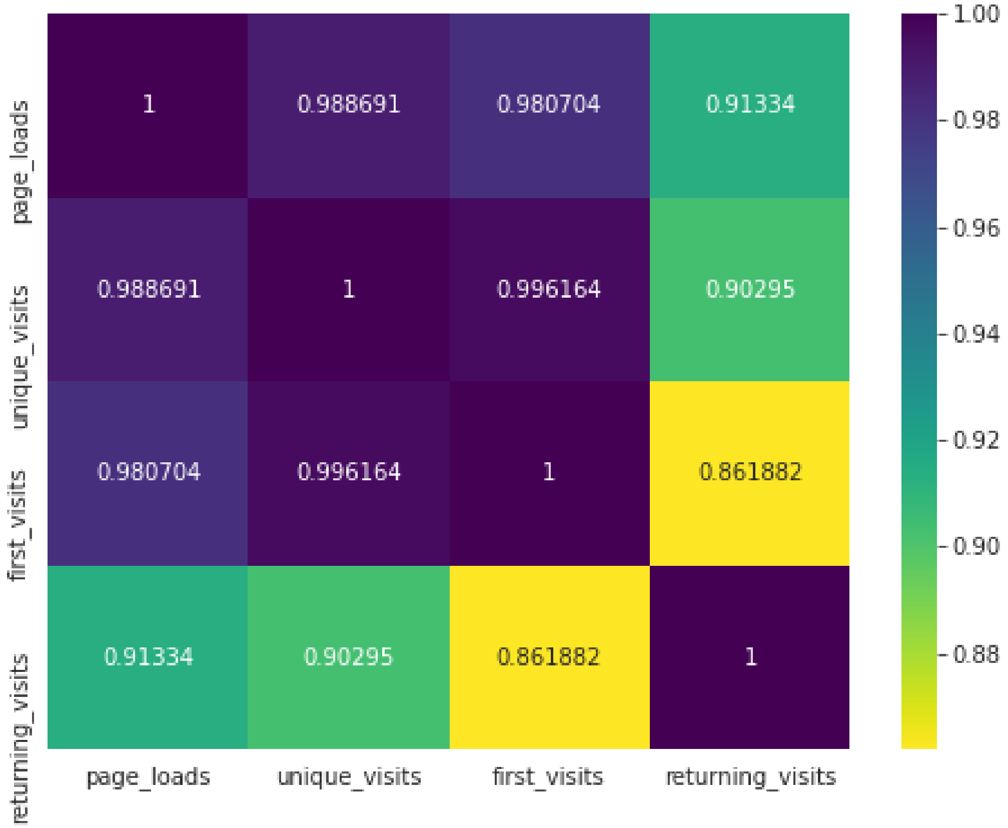
Out 11:



In 12:

```
fig, ax = plt.subplots()

fig.set_size_inches(8, 6)

sns.heatmap(df[['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits']].corr(),

            annot=True,

            cmap='viridis_r',

            fmt='g')
```

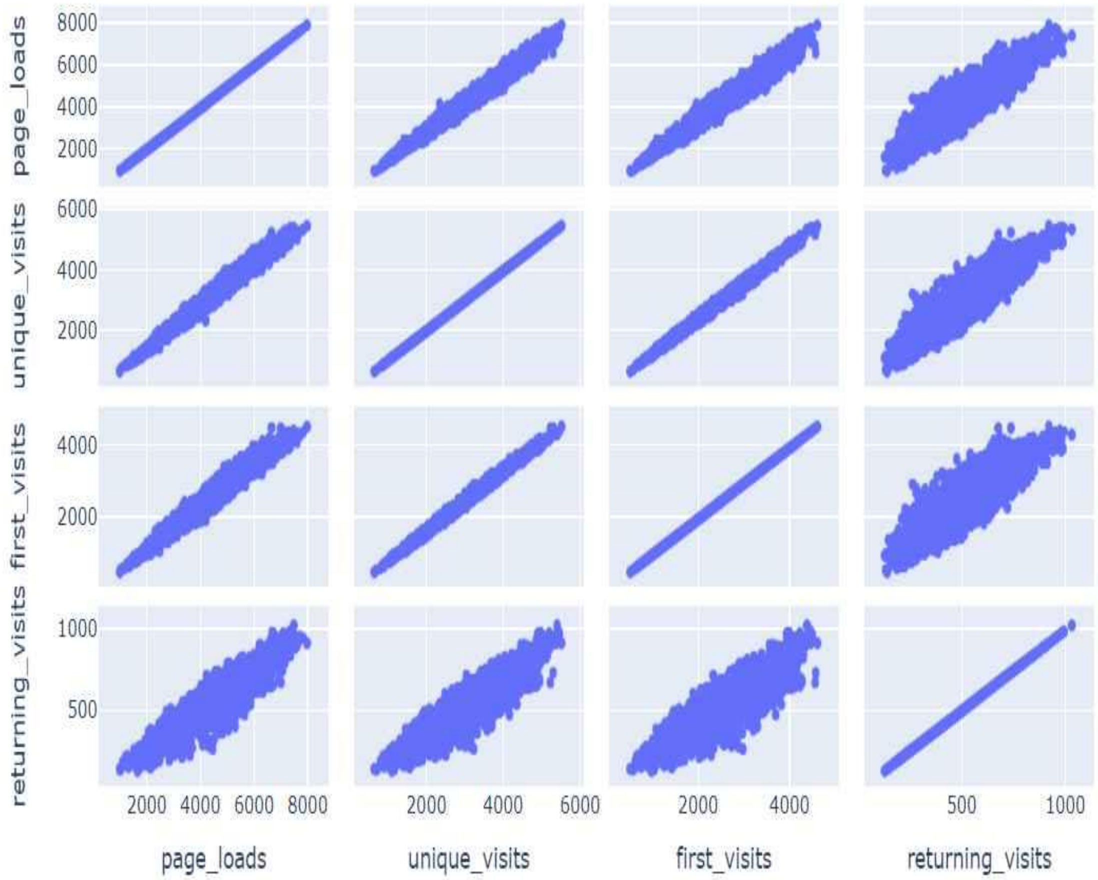
out 12:



In 13:

```
px.scatter_matrix(df[['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits']])
```

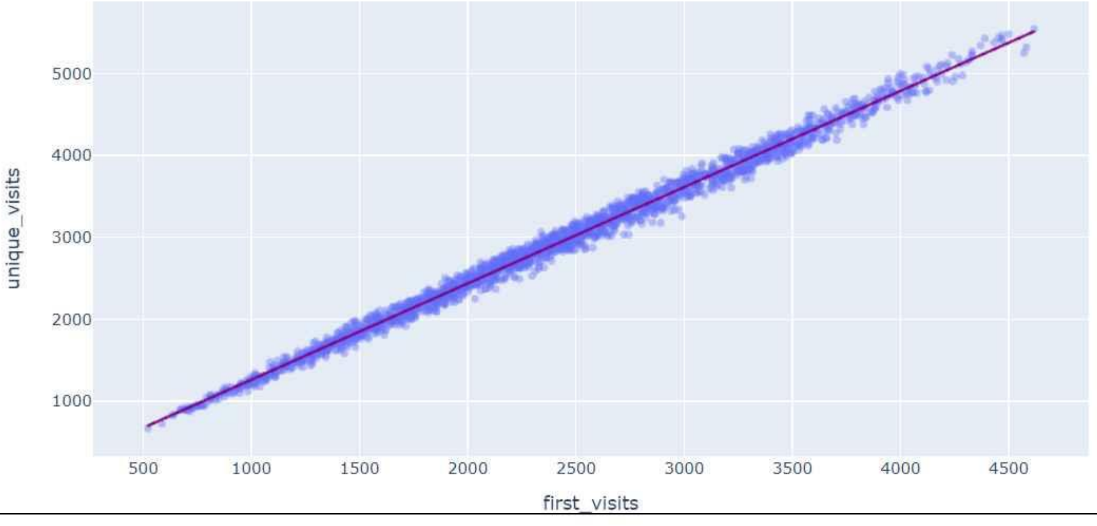
out 13:



In 14:

```
px.scatter(
    df, x='first_visits', y='unique_visits',opacity=0.4,
    trendline='ols',
trendline_color_override='purple',title="Regression line
for unique visits and first visits"
)
```

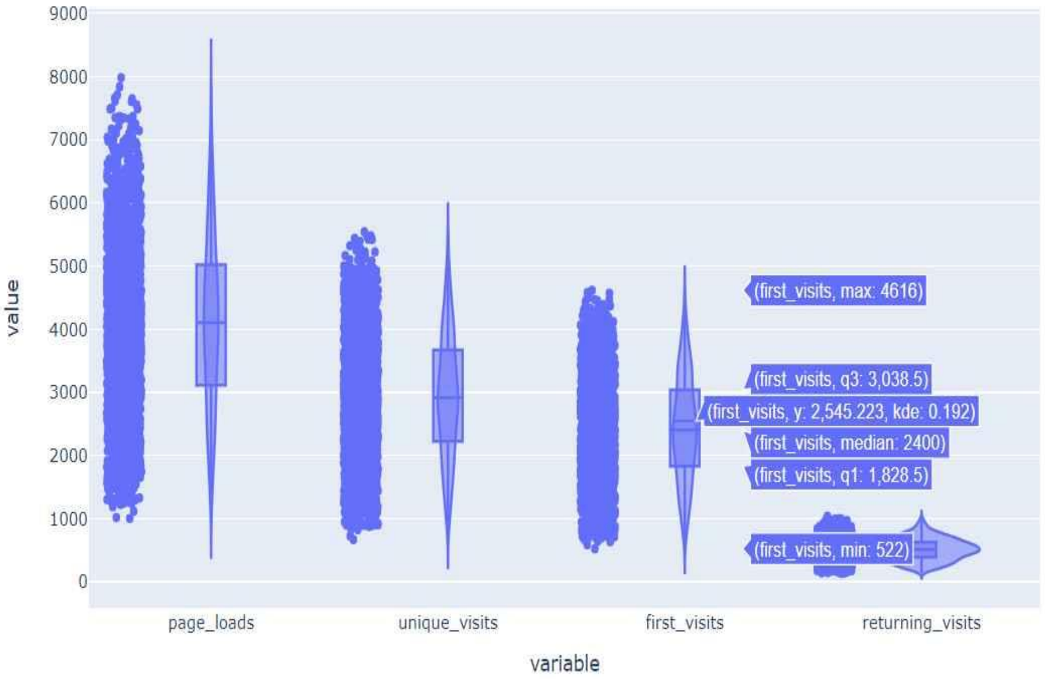
Out 14:



In 15:

```
px.violin(df,y=['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits'],box=True,points='all')
```

out 15:



In 16:

```
regressor2=LinearRegression(fit_intercept=False,normalize=True)
regressor2.fit(X_train, y_train)
```

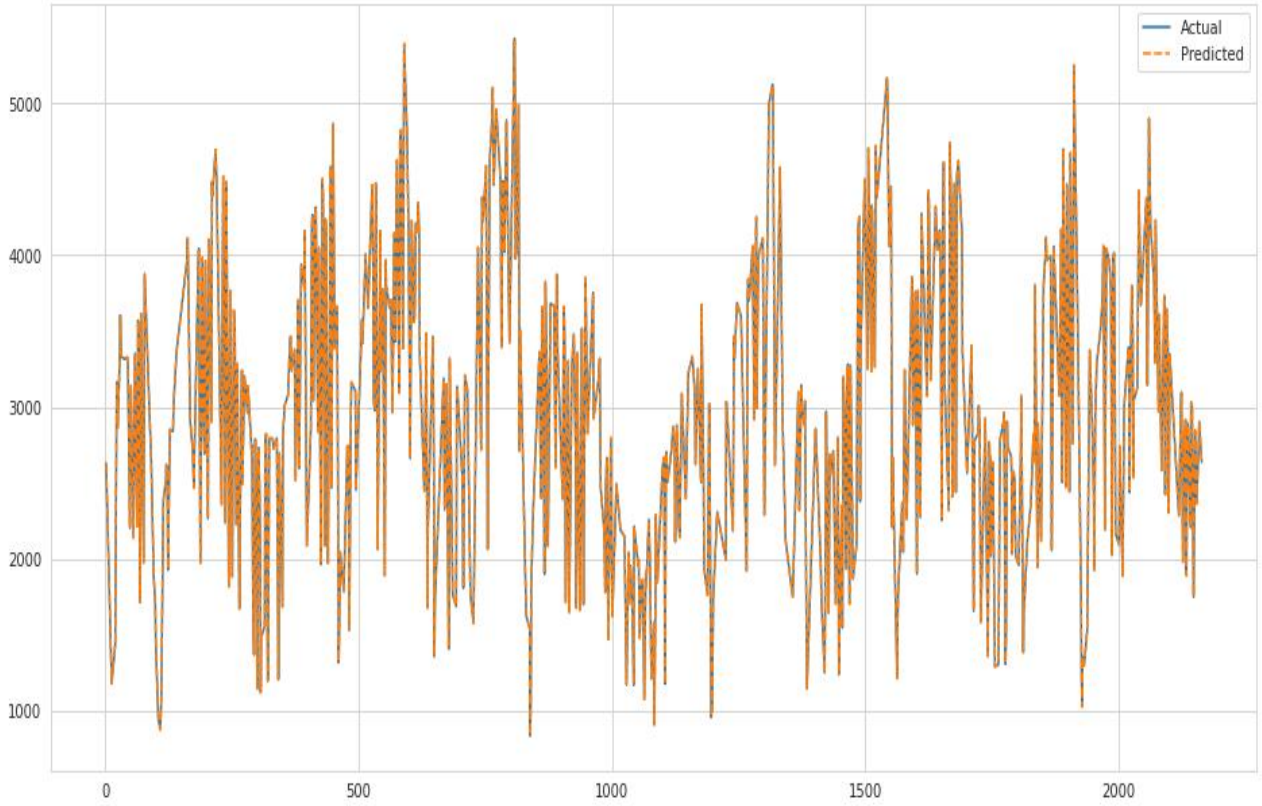
out 16:

```
LinearRegression(fit_intercept=False, normalize=True)
```

In 17:

```
plt.figure(figsize=(16,8))
sns.lineplot(data=lr2)
```

out 17:



In 18:

```
regressor2.score(X_test,y_test)*100
```

out 18:

```
100.0
```

In 19:

```
svr_rbf = SVR(kernel='rbf', C=1e3, gamma=0.00001)
svr_rbf.fit(X_train, y_train)
```

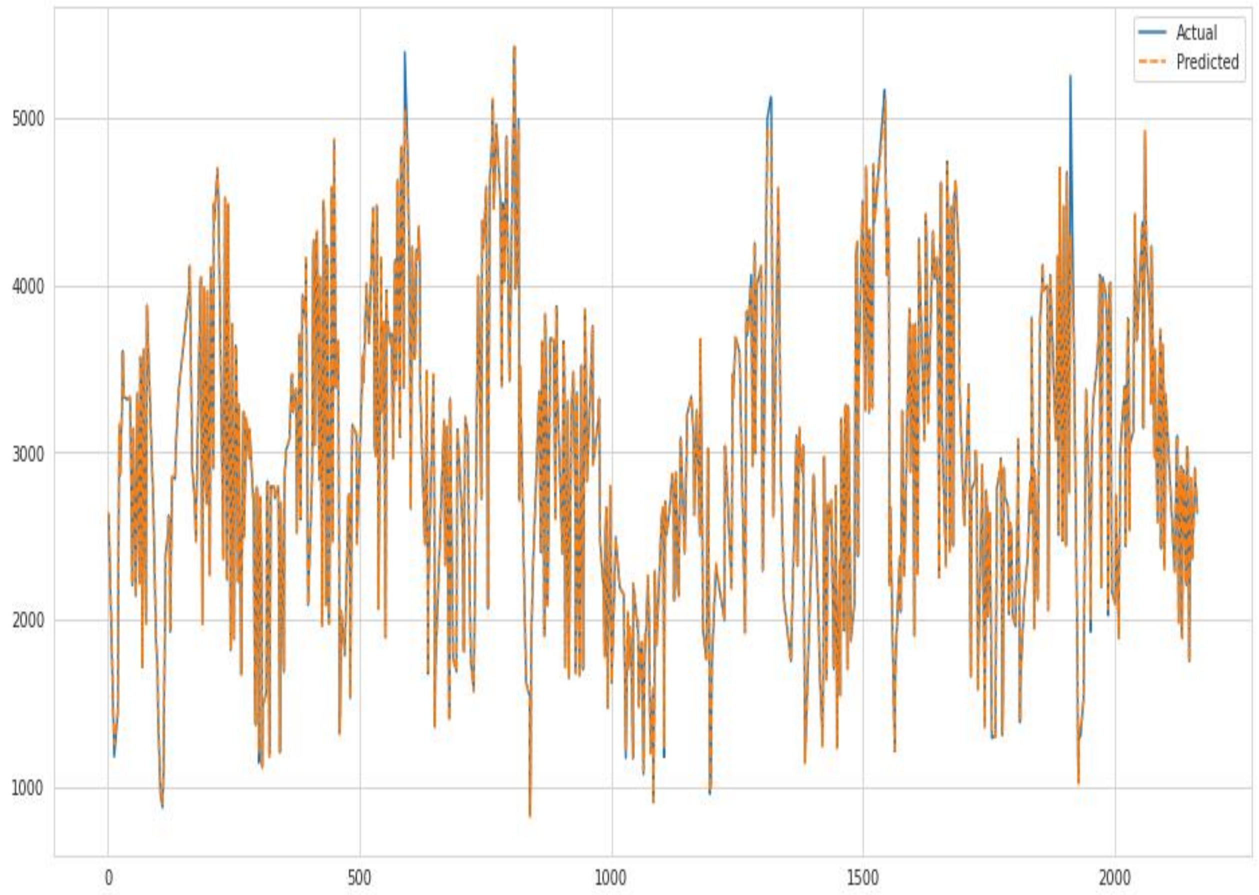
out 19:

```
SVR(C=1000.0, gamma=1e-05)
```

In 20:

```
plt.figure(figsize=(16,8))
sns.lineplot(data=svr)
```

out 20:



In 21:

```
svr_rbf.score(X_test,y_test)*100
```

out 21:

```
99.80054455767926
```

In 22:

```
xgb_r = xg.XGBRegressor(objective ='reg:squarederror',n_estimators = 10, seed = 123)

xgb_r.fit(X_train, y_train)
```

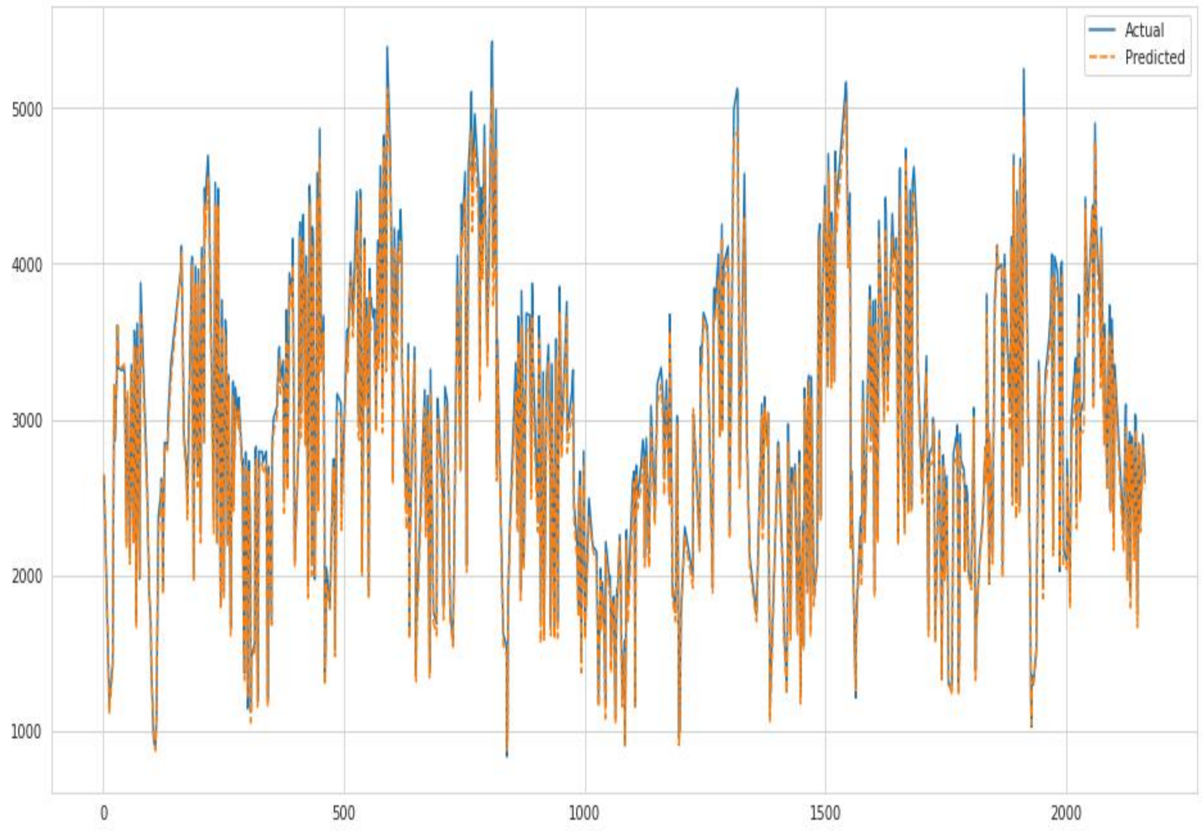
out 22:

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,

             colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,

             importance_type='gain', interaction_constraints='',
 learning_rate=0.300000012, max_delta_step=0,
max_depth=6,
             min_child_weight=1, missing=nan,
monotone_constraints='()',
             n_estimators=10, n_jobs=4, num_parallel_tree=1,
random_state=123,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1,
seed=123,
             subsample=1, tree_method='exact',
validate_parameters=1,
             verbosity=None)
```

In 23:

```
plt.figure(figsize=(16,8))
sns.lineplot(data=xgb_df)
```

out 23:



In 24:

```
xgb_r.score(X_test,y_test)*100
```

out 24:

```
98.7655882096893
```

Conclusion:

- In today's Web development, a good page design is essential. A bad design will lead to the loss of visitors and that can lead to a loss of business. In general, a good page layout has to satisfy the basic elements of a good page design. This includes color contrast, text organization, font selection, style of a page, page size, graphics used, and consistency. In order to create a well-designed page for a specific audience. The developer needs to organized and analyze the users' statistics and the background of the users. Although it can be hard to come up with a design that is well suited to all of the users, there will be a design that is appropriate for most of the audience. The better the page design, the more hits a page will get. That implies an increase in accessibility and a possible increase in business.