UNIVERSITY OF CAMERINO

SCHOOL OF SCIENCE AND TECHNOLOGY

MASTER'S DEGREE IN COMPUTER SCIENCE (CLASS LM-18)

DATA ANALYTICS

YOUNICAM FOR INTELLIGENT DATA ANALYSIS - YOUNICAM#AI

Students
Eric Coleman
Zhaoran Wang
Faiza Iqbal

Professor:
Massimo Callisto De Donato

Table of Contents

INSTALLATION

CLONE THE PROJECT

The project can be cloned from https://github.com/NiiColeman/DataAnalytics.git this URL of you already have git installed or you can download as a zipped file and extract the contents. Clone the project and navigate to the project folder (DataAnalytics)

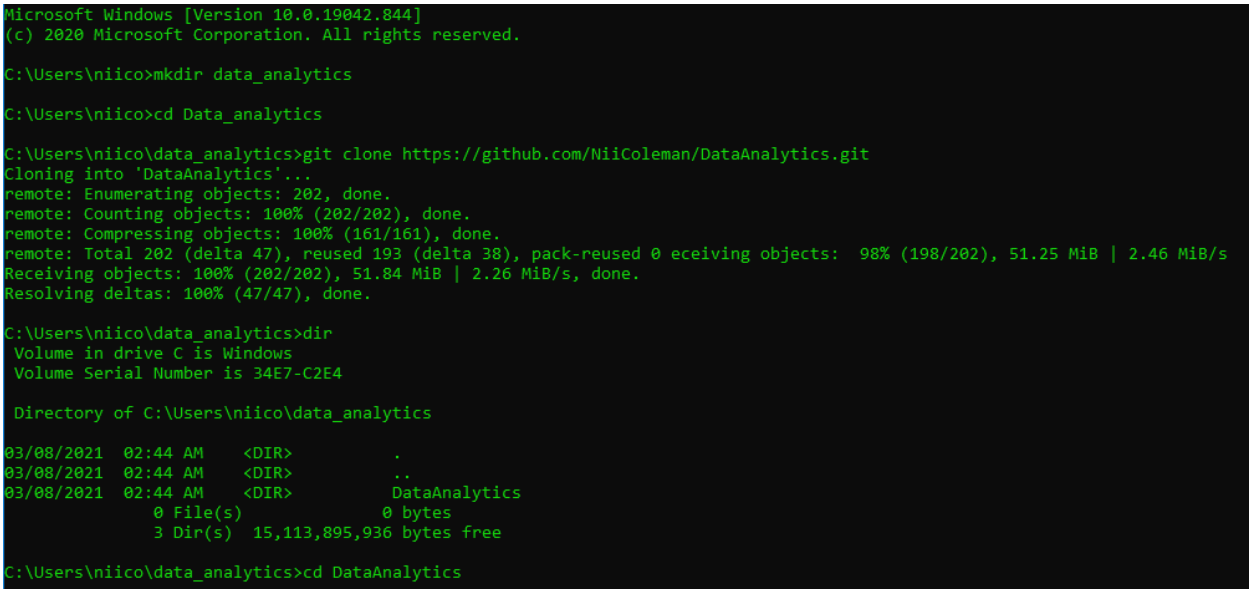


Figure 1

CREATE VIRTUAL ENVIRONMENT

A virtual environment is required to use the project, you can create the virtual environment using the Virtualenv Package or Anaconda. Activate the virtual environment after it has been created

INSTALLING REQUIRED PACKAGES FROM THE REQUIREMENTS.TXT FILE

To install the packages, run 'pip install -r requirements.txt' using the command line in the project directory.

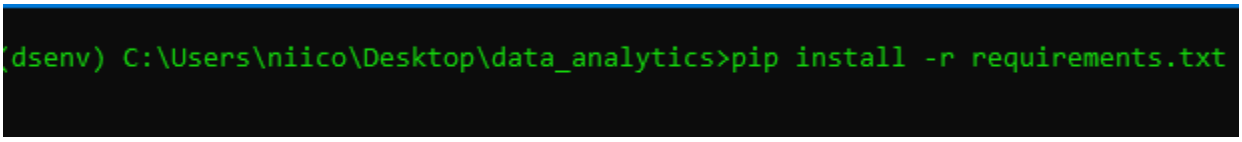


Figure 2

The required packages will be installed in the virtual environment

STARTING JUPYTER IN THE PROJECT DIRECTORY.

Use the 'jypter-notebook .' command to start a session in your current directory

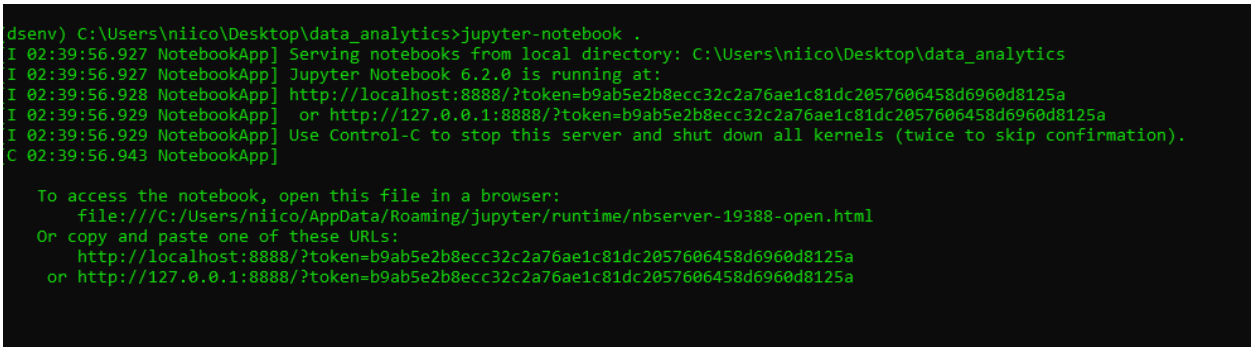


Figure 3

THE PROJECT STRUCTURE

Jupyter will display a list of notebooks as folders, all notebooks have been named according to their purpose, for example, the notebook responsible for data cleaning is called "Data Cleaning". Click on the desired notebook to view and run the code.

TECHNOLOGIES USED

APACHE SPARK

Spark is a fast, open-source data processing engine for Machine Learning and Artificial Intelligence applications. Spark provides a general-purpose cluster system and has a very large open-source community. It is part of the apache Big Data ecosystem and is used in a variety of industries. Pyspark was used in our project.

PYSPARK

Pyspark is the python distribution of apache spark and provides the same functionalities as apache spark. It provides a one time quick installation procedure via the command line and the 'pip' command or 'conda install' command. This makes it easier to maintain different versions of pyspark on the same computer.

PYSPARK IMPLEMENTATION.

Pyspark was used in our project to implement a Linear Regression Machine Learning model, using the spark.ml library. A spark session was created using Spark Session, which allows to programmatically create PySpark RDD, Data Frame. It's object spark is default available in pyspark-shell and it can be created programmatically. PySpark data frames were used to handle and manipulate the rows and columns in the data. The technical descriptions will be shown later in the documentation.

PANDAS

Pandas is a simple, flexible, open-source data analytics tool that is built on top of the python programming language. Pandas creates objects with rows and columns called Data Frames, from a variety of data sources and formats (json, excel, csv), and makes data wrangling as manipulation easy to handle.

PANDAS IMPLEMENTATION

Pandas was used to load the dataset from json format to create a data frame. Pandas methods such as describe(), corr(), info() were used to explore and understand the dataset and relationships between the individual columns. Pandas was also used to change the formats and types of some of the columns.

NUMPY

NumPy is a scientific computation library built on top python ,it lets users create complex arrays, similar to data frames and allows users to make several scientific and mathematical computations on data. NumPy works well with Pandas when handling complex data frames.

NUMPY IMPLEMENTATION

NumPy was used in our project to generate numerical values from a date column. The values of type datetime were convert to numbers using the 'np.to_numeric' function. This makes it easier to use dates as features since datetime objects cannot be considered as features for a linear regression model

SEABORN

Seaborn is a statistical Data visualization tool for python. It allows users to visualize data with its wide variety of graphs. It provides a high-level interface for drawing attractive and informative statistical graphics.

SEABORN IMPLEMENTATION

In our project we used seaborn to create a heatmap for a correlation matrix to visualize the correlation between columns in the dataset

JUPYTER NOTEBOOK

Jupyter is a free, open-source, interactive web tool known as a computational notebook, which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document. Just like a real notebook, Jupyter allows users to run and save outputs from code instantly, provides an interface for visualizations, and allows users to make notes using Markdown.

JUPYTER IMPLEMENTATION

Our project was built entirely in a Jupyter notebook installed in an Anaconda environment. The code was run and saved in several different notebooks appropriately named after what each code does. We used to Markdown tool to also make notes to make it easier to understand.

ANACONDA

Anaconda is an open-source, scientific package management tool built for R and Python. It is widely used in the Data Science community and comes pre-installed with several useful data science libraries and packages suck SkLearn, Pandas, NumPy etc.

ANACONDA IMPLEMENTATION

We used Anaconda to create a ‘conda environment’, which is a virtual environment that allows you to maintain different versions of different packages in an isolated manner. All our packages were installed in the virtual environment, the code was run within that activated virtual environment.

OBJECTIVE: PREDICT THE NUMBER OF PEOPLE IN THE LIBRARY AT A GIVEN TIME

METHODOLOGY

AGILE METHODOLGY

Agile Methodology is a people-focused, results-focused approach to software development that respects our rapidly changing world. It's centered around adaptive planning, self-organization, and short delivery times. It's flexible, fast, and aims for continuous improvements in quality, using tools like Scrum and eXtreme Programming.

Figure 4

APPROACH :MULTIVARIATE REGRESSION.

Multivariate regression is a linear regression approach that has more than one independent variable (the feature for prediction). The method is used to measure the degree at which more than one independent variable and more than one dependent variable, are linearly related. The method is widely used to predict the behavior of the response variables associated to changes in the predictor variables once a desired degree of relation has been established.

CHECKPOINTS FOR MULTIVARIATE REGRESSION

CORRELATION

Correlation helps to determine the strength of the relationship between the predicted value and the dependent variables, correlation ranges between 0-1 and the higher the value the stronger the

correlation hence the stronger the correlation. Correlation does not necessarily imply causality. The fact that there is a strong correlation does not always imply that it is a cause.

MULTICOLLINEARITY

The dependent variable (the predicted number of people) should always have a strong correlation with the independent variables (the features). There are times when the features themselves also have strong correlation, this is what is referred to as multicollinearity. In such cases it is always useful to pick only one among those features.

MEASUREMENT METRICS FOR MULTIVARIATE REGRESSION

R-SQUARED (R2)

The R-squared metric is the most popular practice of evaluating how well your model fits the data. R-squared value designates the total proportion of variance in the dependent variable explained by the independent variable. It is a value between 0 and 1; the value toward 1 indicates a better model fit.

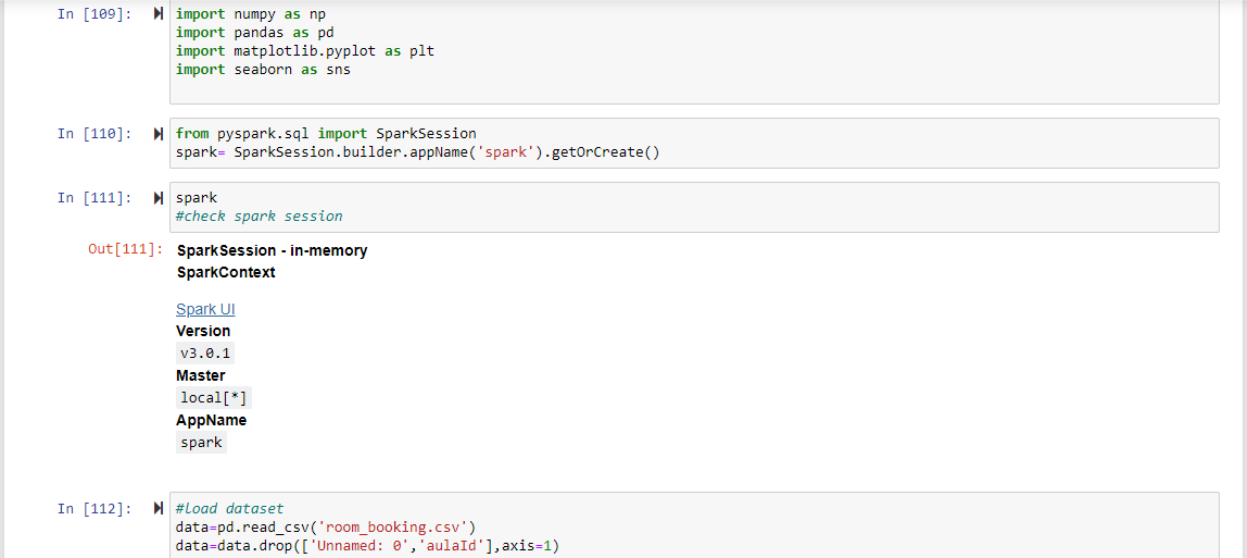
ROOT MEAN SQUARED ERROR (RMSE)

This is used to predict the difference between the predicted value from the model and the actual observed value. RMSE shows a numerical representation of how far off the predicted values are to the actual values; hence a lower RMSE value signifies that the model performance is good.

IMPLEMENTATION WITH PYSPARK

STARTING A SPARK SESSION IN JUPYTER

Spark and other related libraries are exported at the beginning of the project , a spark session is then required in order to use the features of pyspark.



```
In [109]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

In [110]:
from pyspark.sql import SparkSession
spark= SparkSession.builder.appName('spark').getOrCreate()

In [111]:
spark
#check spark session

Out[111]:
SparkSession - in-memory
SparkContext

Spark UI
Version
v3.0.1
Master
local[*]
AppName
spark

In [112]:
#Load dataset
data=pd.read_csv('room_booking.csv')
data=data.drop(['Unnamed: 0','aulaId'],axis=1)
```

Figure 5

The excel file is then loaded into a Pandas Data Frame after the session has been started.

CLEANING THE DATA

Missing data can lead to misinterpretation of the data and the weight of values , it is always necessary to handle missing data in the most fitting way. For our project , we used to rooms dataset and the roomSchedule dataset.

```
rooms
```

	_id	Sede	aulaId	aulaDes	capienza	poloDes	date	scheduleType	schedules
0	5f5b4fdd400b9fbfbe0b8e94	1	1.0	69	10	15	2020-10-02 19:17:16.414000+00:00	[]	[]
1	5f5b4fdd400b9fbfbe0b8e95	1	2.0	25	24	15	2020-12-15 15:38:27.918000+00:00	[]	[]
2	5f5b4fdd400b9fbfbe0b8e96	1	3.0	12	12	2	2020-10-29 10:10:23.875000+00:00	[]	[]
3	5f5b4fdd400b9fbfbe0b8e97	1	4.0	16	10	2	2020-12-15 10:17:12.238000+00:00	[]	[]
4	5f5b4fdd400b9fbfbe0b8e98	1	7.0	35	76	18	2020-12-16 10:04:16.903000+00:00	[]	[]
...	...	...	...	...	...	...	...	...	...
143	5f9ade072358534e7e6d469c	3	NaN	123	50	29	2020-10-29 15:21:43.206000+00:00	[]	[]
144	5fa05a957916daa867a158ae	1	NaN	31	14	1	2020-12-16 08:04:17.292000+00:00	[]	[]
145	5fa05aa77916daa867a158af	1	NaN	36	9	1	2020-12-16 09:46:29.789000+00:00	[]	[]
146	5fc95263eb6dbced74b94f68	2	NaN	124	6	4	2020-12-03 21:02:27.777000+00:00	[]	[]
147	5fca3ca8eb6dbced74b951b7	2	NaN	125	13	14	2020-12-04 13:42:00.827000+00:00	[]	[]

148 rows × 9 columns

Figure 6 rooms

```
rs.head(50)
#roomSchedules dataset
```

	_id	bookings	room	date
0	5fa8267c1bd2a03f4641a099	[{'_id': '5fa8267c1bd2a03f4641a09a', 'start': ...	5f5bd5fafc7a3450ea06e81c	2020-11-08 17:10:20.328000+00:00
1	5fa859611bd2a03f4641a0e0	[{'_id': '5fa859611bd2a03f4641a0e1', 'start': ...	5f5cc739fc7a3450ea06e838	2020-11-08 20:47:29.186000+00:00
2	5fa87aad1bd2a03f4641a130	[{'_id': '5fa87aad1bd2a03f4641a131', 'start': ...	5f5b4fdd400b9fbfbe0b8ec3	2020-11-08 23:09:33.878000+00:00
3	5fa8fb591bd2a03f4641a1e9	[{'_id': '5fa8fb591bd2a03f4641a1ea', 'start': ...	5f5bd5fafc7a3450ea06e81c	2020-11-09 08:18:33.344000+00:00
4	5fa8fc3d1bd2a03f4641a1f7	[{'_id': '5fa8fc3d1bd2a03f4641a1f8', 'start': ...	5f5cc50bfc7a3450ea06e834	2020-11-09 08:22:21.433000+00:00
5	5fa8fc9b1bd2a03f4641a203	[{'_id': '5fa8fc9b1bd2a03f4641a204', 'start': ...	5f5cc50bfc7a3450ea06e834	2020-11-09 08:23:55.389000+00:00
6	5fa8fcf21bd2a03f4641a20d	[{'_id': '5fa8fcf21bd2a03f4641a20e', 'start': ...	5f5cc50bfc7a3450ea06e834	2020-11-09 08:25:22.411000+00:00
7	5fa92a841bd2a03f4641a319	[{'_id': '5fa92a841bd2a03f4641a31a', 'start': ...	5f5cc781fc7a3450ea06e839	2020-11-09 11:39:48.395000+00:00
8	5fa933751bd2a03f4641a32e	[{'_id': '5fa933751bd2a03f4641a32f', 'start': ...	5f5bd4928db89350a13a80d8	2020-11-09 12:17:57.249000+00:00
9	5fa968cf1bd2a03f4641a438	[{'_id': '5fa968cf1bd2a03f4641a439', 'start': ...	5f5cc55bfc7a3450ea06e835	2020-11-09 16:05:35.221000+00:00
10	5fa971f71bd2a03f4641a443	[{'_id': '5fa971f71bd2a03f4641a444', 'start': ...	5f5cc739fc7a3450ea06e838	2020-11-09 16:44:39.238000+00:00
11	5fa9af961bd2a03f4641a48a	[{'_id': '5fa9af961bd2a03f4641a48b', 'start': ...	5f5b4fdd400b9fbfbe0b8ebd	2020-11-09 21:07:34.847000+00:00
12	5fa9bb261bd2a03f4641a4c0	[{'_id': '5fa9bb261bd2a03f4641a4c1', 'start': ...	5f5b4fdd400b9fbfbe0b8ee7	2020-11-09 21:56:54.603000+00:00
13	5faa478b1bd2a03f4641a527	[{'_id': '5faa478b1bd2a03f4641a528', 'start': ...	5f5cc781fc7a3450ea06e839	2020-11-10 07:55:55.149000+00:00
14	5faa8adf1bd2a03f4641a6a4	[{'_id': '5faa8adf1bd2a03f4641a6a5', 'start': ...	5f5bd4928db89350a13a80d8	2020-11-10 12:43:11.169000+00:00
15	5faa91731bd2a03f4641a71a	[{'_id': '5faa91731bd2a03f4641a71b', 'start': ...	5f5b4fdd400b9fbfbe0b8e9e	2020-11-10 13:11:15.957000+00:00

Figure 7 roomSchedules

Firstly we checked the number of rows that contain NaN (null values), 29 rows in the aulaId column had missing data. 29 rows out of 148 is quite a huge number and dropping these rows would not be a good idea. Given the context of the project and the dataset that we had, we had two options, the first was to fill the null values by checking the corresponding aulaDes column for

previous rows which do not have missing values or to drop the aulaId column entirely to and use aulaDes to Identify the Aula. We went with the latter since we just need a numerical representation of the aula(room) for identification and feature selection, also because of multicollinearity, we were bound to only use one of the 2 columns anyway since both have the same correlation.

We then proceeded to merge the 2 datasets (room and roomSchedules), while rooms contained information about a particular room in a given department, roomSchedules gave information about the booking of a particular room in a department, the day of the booking, the time and the number or people. Our goal is to predict the number of people in the library at a given time. We first fist concatenated both columns on a given axis to see if both data frames could be merged, then we applied a Pandas inner merge function based on a foreign key (room, _id). This is quite like an inner join in SQL on a particular foreign key column. After getting the data, we had to aggregate the number of persons per room on a given day, using the aggregate function. The final part of the data cleaning process was splitting the dates, into date, minutes, hours , and days of the week to get more features

FINDING CORRELATION

As explained earlier correlation helps to determine how independent variables relate to the dependent variable. We used the ‘df.corr()’ method to check the correlation and built a correlation heatmap with seaborn.

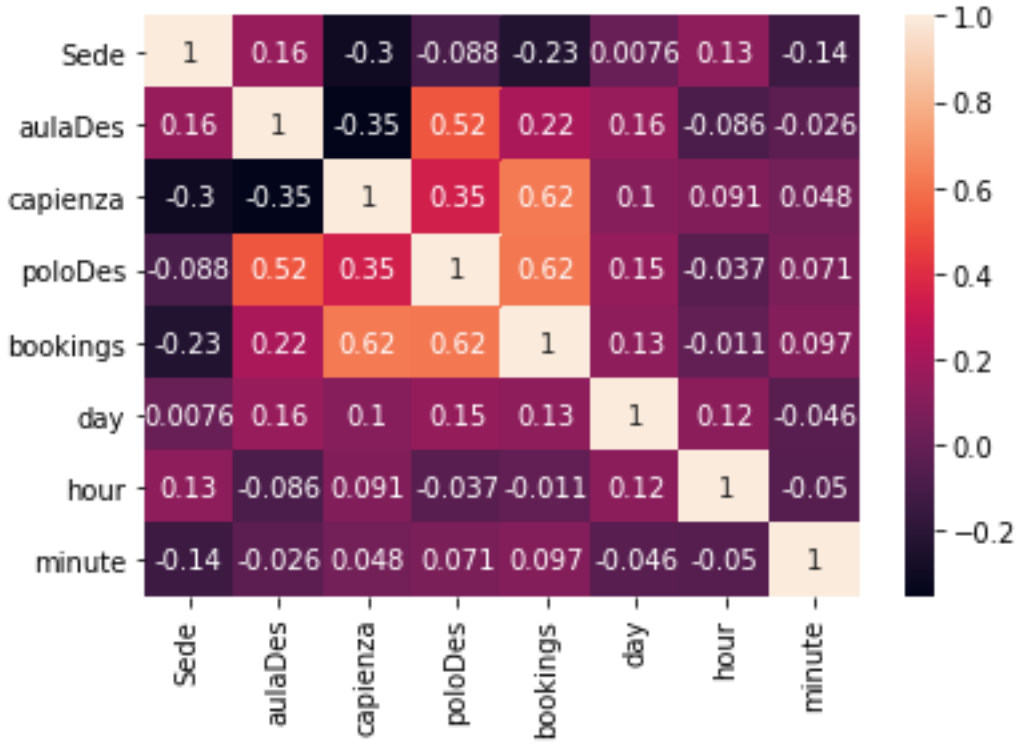


Figure 8 heatmap

FEATURE ENGINEERING

This step involved the selection of features to be used in the training of the machine learning model, in general machine learning algorithms provide the best results when fed with the best set of features. In order to predict the number of people in a given room, we first need to know the capacity of the room, room capacity is a fair way of determining how many people can fit in a room. We also need to know the number of people who booked a room in a given day or with a given timeframe. Then of course we have our aula, sede , and polo. Our initial feature were constructed with aulaDes ,poloDes, Capienza, bookings, day, hour, and time but these features provided really bad results. In order to achieve the optimum results, we proceeded to use the aulaDes, poloDes, capienza for the prediction. We could have tried to generate more features but we didn't want to over fit.

Pyspark VectorAssembler was used to assign the set of features and the dependent variable, booking. This generated a new column in our dataset called features with the list of features provided.

TRAINING AND TESTING.

This procedure helps to determine the accuracy of the model, the data is split into parts ,training and testing, usually in some kind of ratio. In this project it was split 70:30 using the random split method built on the PySpark Regression model after the model has been applied to features and dependent variable.

MEASUREMENT METRICS

PySpark ML library abstracts the complexities of machine learning from the user and makes otherwise complicated calculations readily available through simple methods. The model is trained with the training data and the accuracy in measured using RMSE and R Squared values

R SQUARED(R2).

We achieved an r2 score of 0.61 (about 61%)

RMSE

RMSE Score of 17.1

THE PREDICTED VALUES AND CONCLUSIONS

```
predictions=model.transform(test)
```

```
#display the predictions
predictions.toPandas()
```

	features	bookings	prediction
0	[26.0, 112.0, 11.0]	13	18.587484
1	[26.0, 112.0, 11.0]	14	18.587484
2	[26.0, 112.0, 11.0]	14	18.587484
3	[26.0, 112.0, 11.0]	17	18.587484
4	[26.0, 112.0, 11.0]	21	18.587484
5	[26.0, 112.0, 11.0]	33	18.587484
6	[10.0, 115.0, 10.0]	1	6.274557
7	[10.0, 115.0, 10.0]	2	6.274557
8	[10.0, 115.0, 10.0]	2	6.274557
9	[10.0, 115.0, 10.0]	8	6.274557
10	[26.0, 112.0, 11.0]	4	18.587484
11	[27.0, 113.0, 35.0]	69	62.286125
12	[27.0, 113.0, 35.0]	76	62.286125
13	[27.0, 113.0, 35.0]	55	62.286125

Figure 9 predictions

The predicted values were close in some cases, and far apart in some other. In this case the set of features used do not provide enough variety to train the model, the room, department and capacity of the mentioned room do not provide enough variety. Assuming row 0 represents the library, given the room, department and capacity of the room, and based on previous instances of the number of booking made in that room every single day, the model predicted 18 people as compared to the actual 13 that were

there on that given day. The same model predicted 14 for the same instances but for another day. There are factors that affect the usage of the room beyond the given parameters for example weather conditions on that day, it could have rained or snowed hence the small number of people. as mentioned before correlation does not necessarily imply causality.

Although the goal was to predict the number in each time frame, the given features were not enough to guarantee a trusted result, so improvisations had to be made in order to ensure some kind of reliable outcome.

OBJECTIVE 2: PREDICT THE NUMBER OF PEOPLE IN A GIVEN DEPARTMENT.

APPROACH : TIME SERIES

Our goal is that given an input(datetime) and return an output value. The time series model is very suitable for this problem.

There is some introduction and definition about time series.

Time series data is data that is collected at different points in time. This is opposed to cross-sectional data which observes individuals, companies, etc. at a single point in time.

Because data points in time series are collected at adjacent time periods there is potential for correlation between observations. This is one of the features that distinguishes time series data from cross-sectional data.

A time series is a sequence of numerical data points in successive order. In investing, a time series tracks the movement of the chosen data points, such as a security's price, over a specified period of time with data points recorded at regular intervals. There is no minimum or maximum amount of time that must be included, allowing the data to be gathered in a way that provides the information being sought by the investor or analyst examining the activity.

Important: Time series analysis can be useful to see how a given variable changes over time.

In my opinion. I think our project goal has some similarity with other regression problems such as given an input time return the price of stock, the price floating situation of an item over the time. So we can try to use time series.

PROBLEM

Our dataset is not continuous. Some days of them are null, which is not allowed and supported to time series. First choice is we can fill out by 0. All the null value will be replaced by 0. A better way to solve this problem is interpolation.

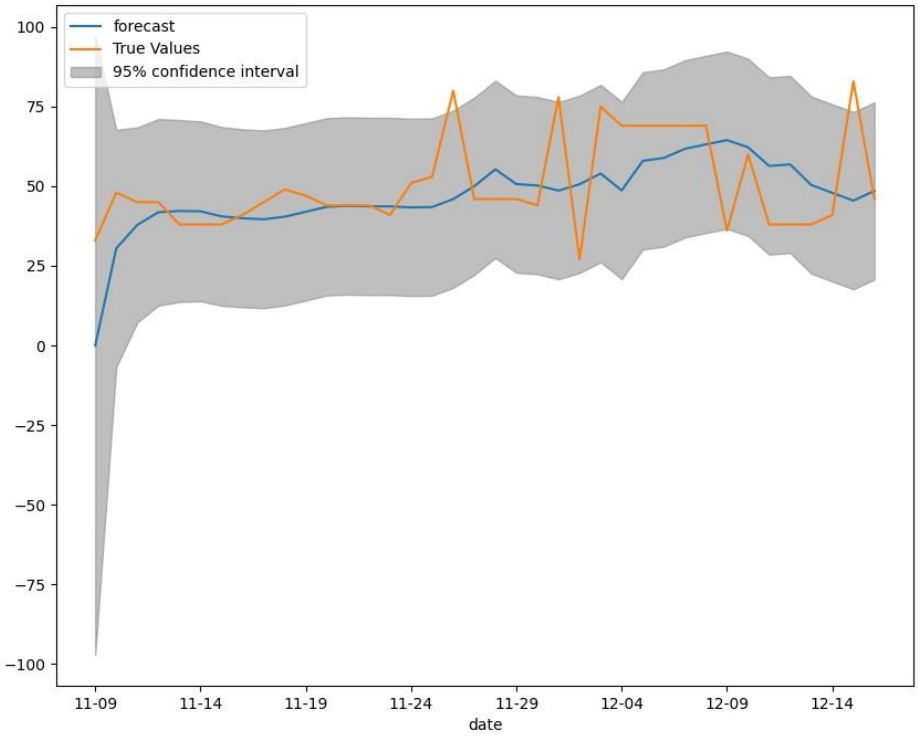
A simple solution is replace the null by the last one valid value before this day.

The following code will realize this task.

```
import pandas as pd

df = pd.read_csv('data_csv/presences2.csv')

df['date'] = pd.to_datetime(df['date'])

grouper = pd.Grouper(key='date', freq='1D')

res = df.groupby(grouper).first().ffill().reset_index()

res = res.fillna(0).drop(columns=['Unnamed: 0'])

res.to_csv('data_csv/presences3.csv')
```

Time series model training and evaluation code

```
dataset = pd.read_csv('data_csv/presences3.csv')

y1 = np.array(dataset['amount'])

dates = dataset['date']

for i in range(0, len(dates)):

    dates[i] = dates[i][5:]

y = pd.Series(y1, index=dates)

arma_mod = ARIMA(y, order=(2, 0, 2), trend='n')

arma_res = arma_mod.fit()
```


forecast
True Values
95% confidence interval
100
75
50
25
0
−25
−50
−75
−100
11-09
11-14
11-19
11-24
11-29
12-04
12-09
12-14
date