MLDB Project Final Report

Group 9

Managing ML data and models for Video Games

Project Members:

This is a one-person project. I'm Yunrui Shao from the applied data science program, and my USC-id is 8706491261. My undergraduate major is mathematics, and I'm really a beginner in machine learning. During the process of this program, I'm able to practice both skills of programming and algorithms, and I really learned a lot from this project.

Project Description:

This project is about video game sales. It's aimed to help determine which are the key factors determining game sales around the global market, and I also hope to provide a useful forecast to the level of popularity of the games. The targeted users could be not only sellers, but also customers and game designers. To think of the ultimate goals, the system is supposed to be also used for users to get inspirations and recommendations of new video games.

I got the raw dataset from Kaggle(https://www.kaggle.com), and it's in .csv format. The file contained 11 columns included the video game sales data from different parts of the world (North America, Europe, Japan, Other areas), and some basic characteristics, like name, genre, publisher, publishing year, and etc. of each video game.

[5]:
```
vgdata = pd.read_csv('vgsales.csv')
print(vgdata.shape)
vgdata.head()
```

```
(16598, 11)
```

[5]:

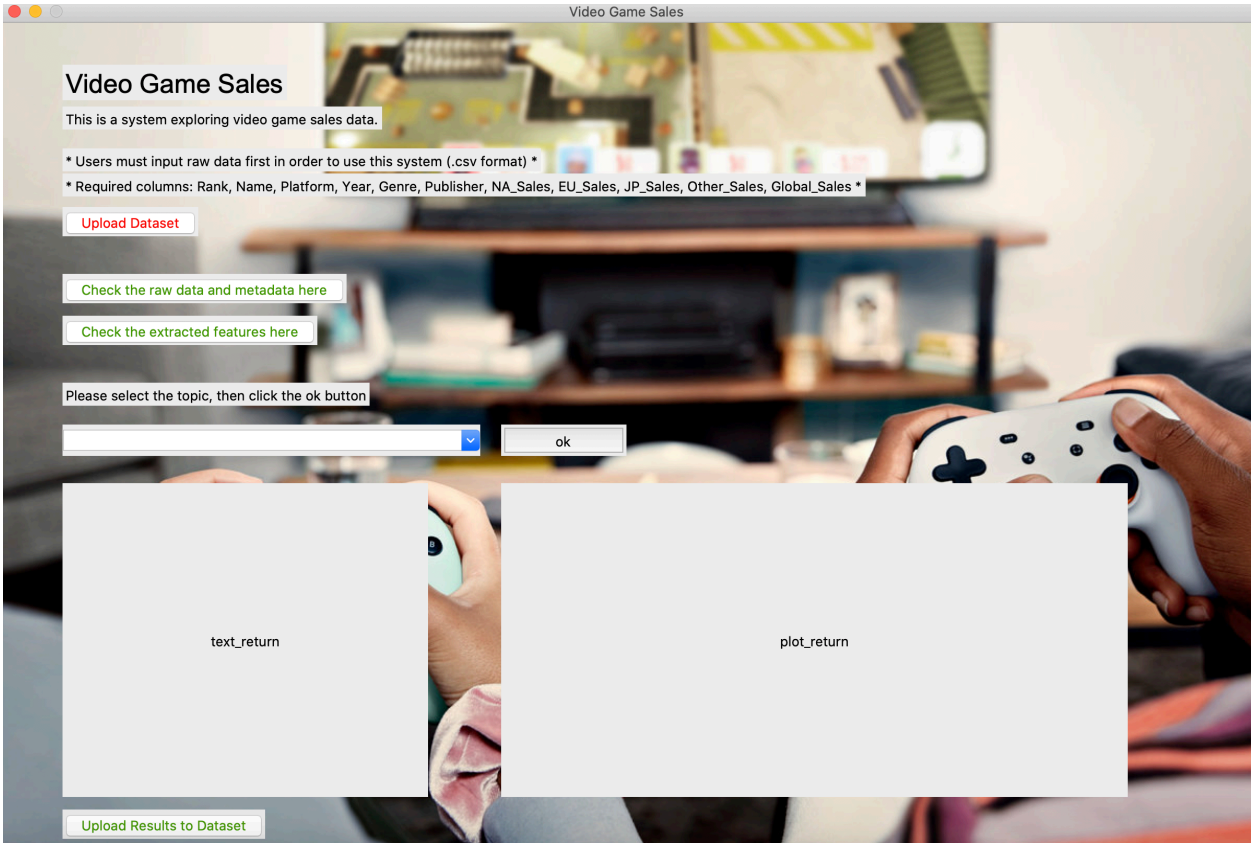
	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37

Graph 1: description of the original dataset

In this project, I used Firebase as the cloud database for uploading the required datasets. And there are buttons in my system to transfer the required data into the cloud database. For writing the main python codes, I used the Jupyterlab. Since this is a one-person project, I did not develop a web browser-based UI. Instead, I used the Tkinter in python to create the view of my final project.

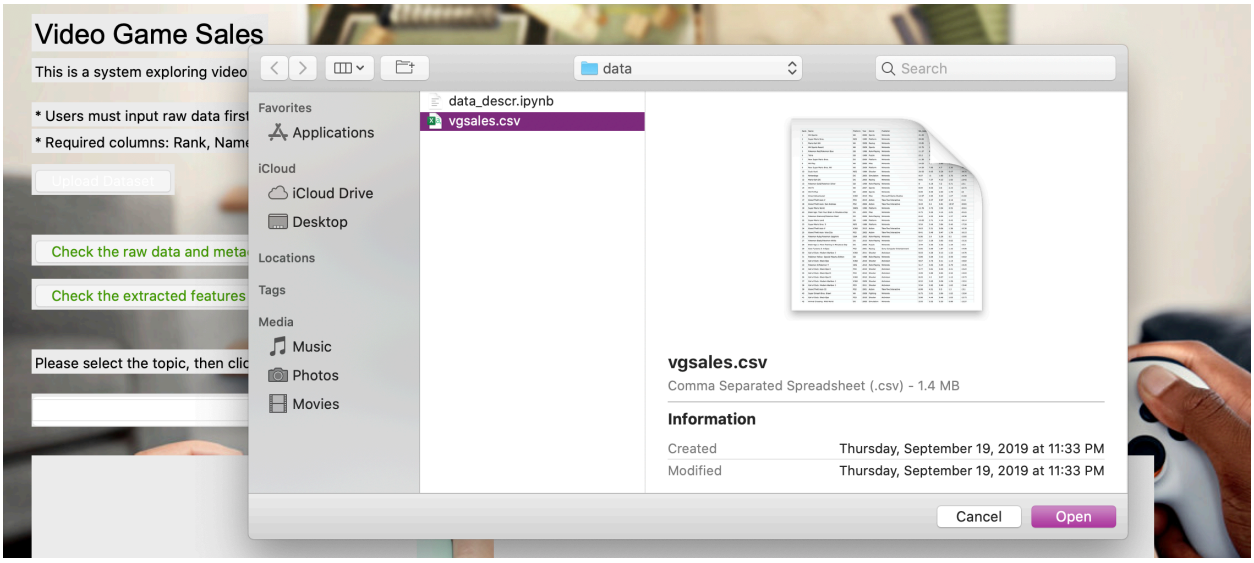
Project UI:

Here's a general view of my project's user interface:



Graph 2: general view of the Video Game Sales UI

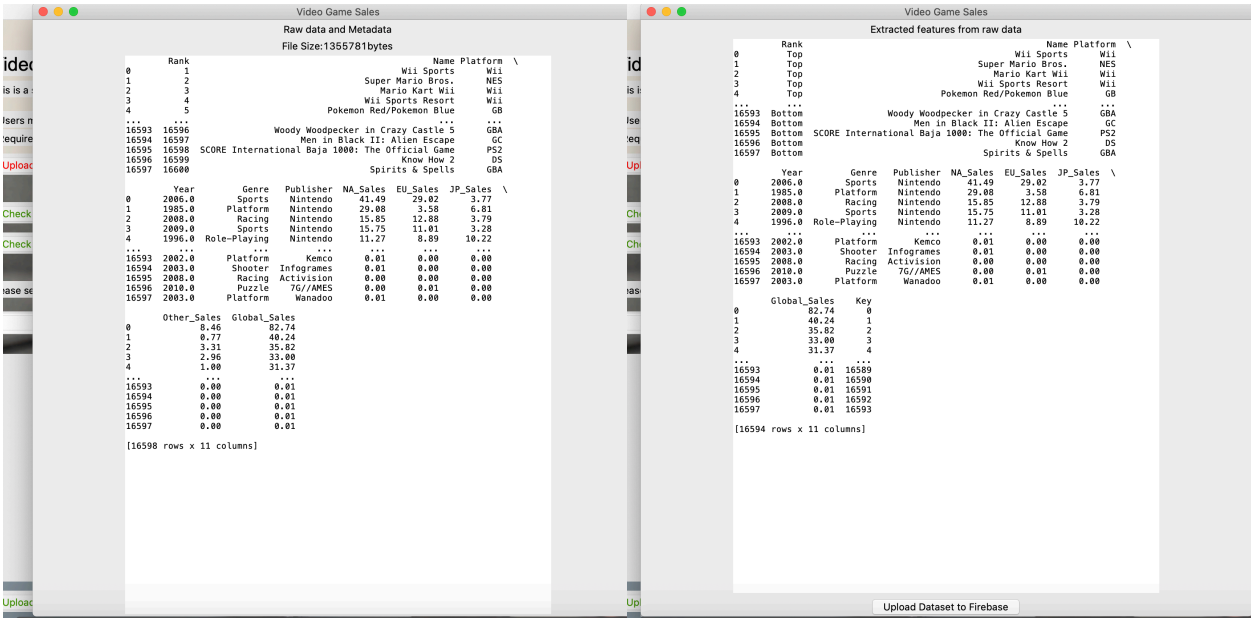
On top of the view, I put a title "Video Game Sales" with a simple introduction of the system. Under that, we can see an Upload Dataset button. In this system, the user must upload the raw data from their computer first in order to use all the functions inside. The required format of the uploaded file is also included as an important warning message on the top of the button. After clicking the "Upload Dataset", the user would get a pop-up window like this:



Graph 3: the pop-up window after clicking “Upload Dataset”

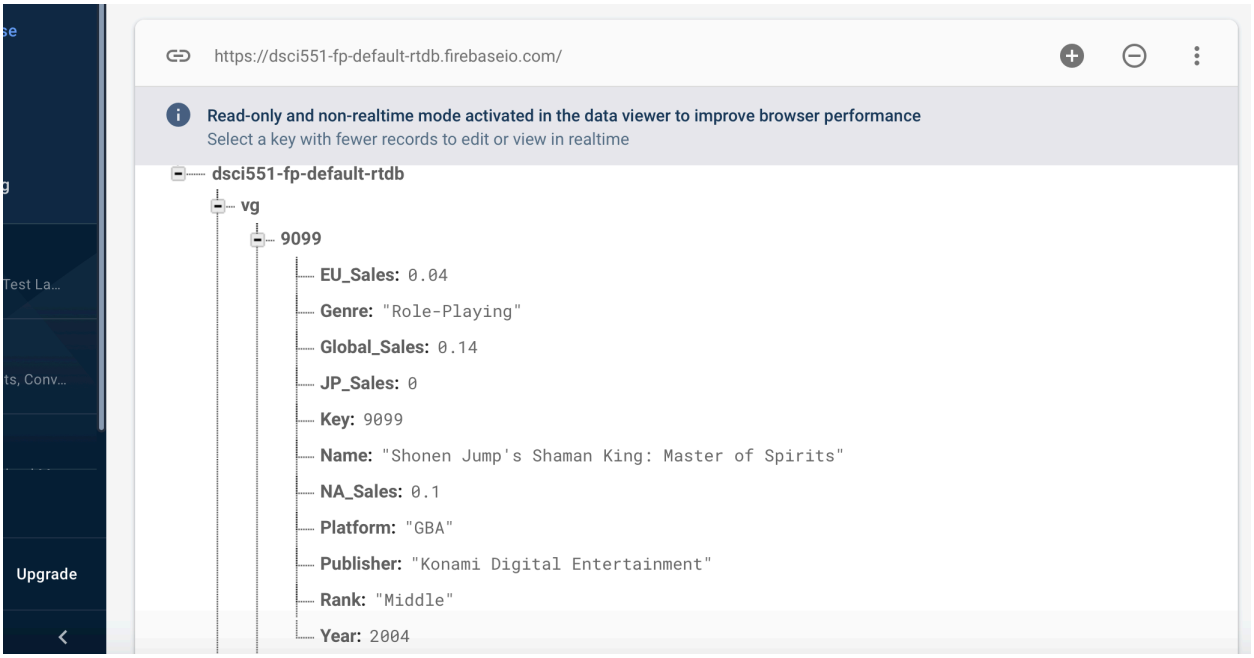
Any .csv file in the required format could be uploaded to explore in this system. In this case, I chose the “vgsales.csv” file to show as an example.

The file can be transferred to the system successfully after clicking “Open”. In this case, we are able to check the raw data, metadata, and extracted features by clicking the corresponding buttons in the system.



Graph 4: pop-up windows of showing informations extracted from the user’s dataset

The user could see their raw data printed after clicking “Check the raw data and metadata here” (on the left side in graph 4). Besides the original dataset, the user is also able to see the file size (in bytes), numbers of columns and rows as the metadata extracted from the dataset. On the other hand, the user could see the extracted features of the raw data after clicking “Check the extracted features here” (on the right side in graph 4). For the future usages of the dataset, I modified the raw data in several steps. First, I did the data cleaning, and remove all the NAs in the dataset. I dropped the “Other_Sales” column, while adding a new “Key” column, containing the row numbers of each game (this step is necessary for giving a key to the dataset in order to upload it into the firebase). After that, I replaced the original values inside the “Rank” column to 3 groups: 1 to 5000 is categorized as “Top”; 5000 to 10000 is categorized as “Middle”; rank after 10000 is categorized as “Bottom”. As showed in right side of the graph 4, we can see that the modified data frame has 16594 rows and 11 columns. There’s a “Upload Dataset to Firebase” button at the bottom of this pop-up window, and the user is able to upload the modified dataset into the firebase after clicking the button.

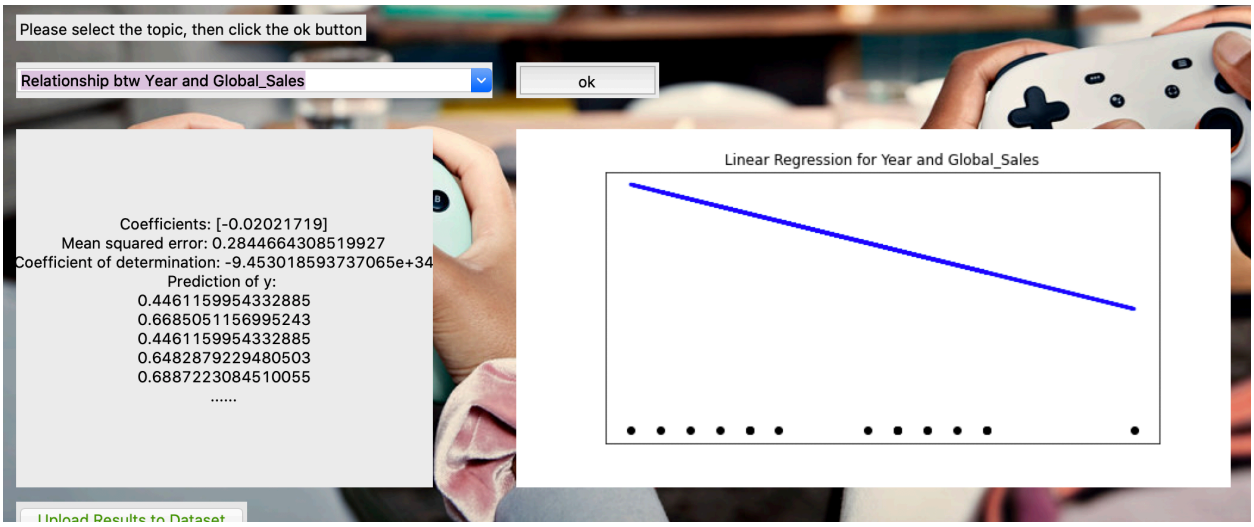


Graph 5: a screenshot of the firebase after clicking the first uploading button

After the functions described above, here comes to the most exciting part of this system. By using the dropbox, the user could select one of the two machine learning topics to explore.

Let me introduce the basic interface of this part first. After selecting the topic and clicking the "ok" button, the two grey areas below would show the results from the given dataset. The left area would show some text results, which are mainly mathematical expressions including mean squared errors, mean absolute errors, R2 coefficients, etc. These are the results prepared for users who know well about mathematics and ML algorithms. The right area would show a graph plotted for the topic. This would be a more visualized expression prepared for users who do not have much knowledge about ML algorithms.

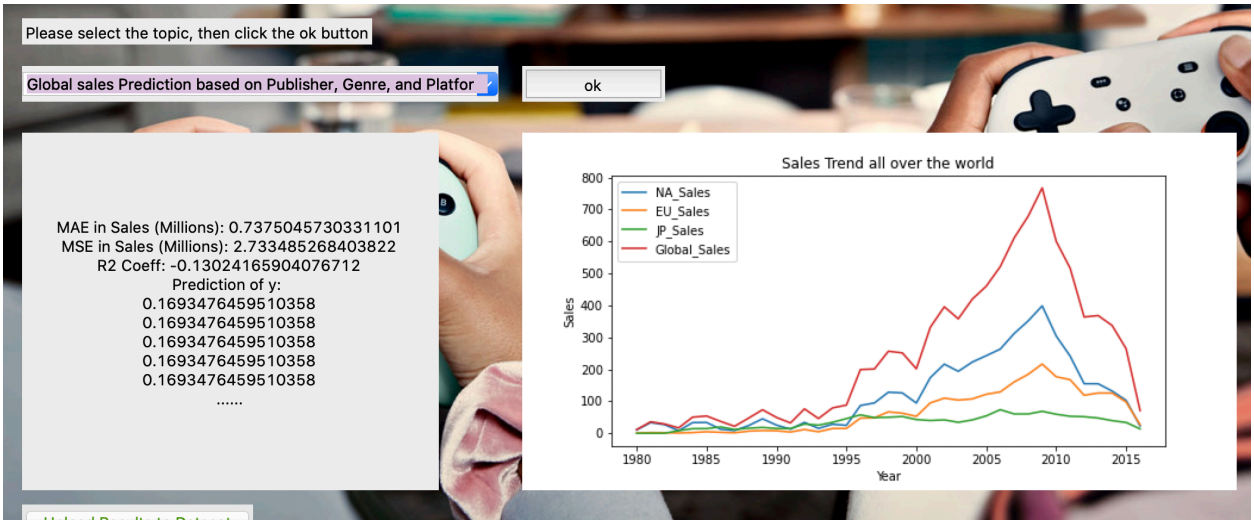
To be more specific, the first topic we could explore is Relationships between Year and Global_Sales. As you can see in the graph below, I prepared the coefficients, mean squared error, coefficient of determination, and the prediction of y on the left area. During this exploring, I used a simple linear regression. Combined with the plot on the right, we could see that "Year" might not be a significant factor affecting the global sales. At the same time, "Year" could not affect "Rank" either, since rank has a positive correlation with "Global_Sales". And in this case, I would not update and add the prediction of y into the cloud database.



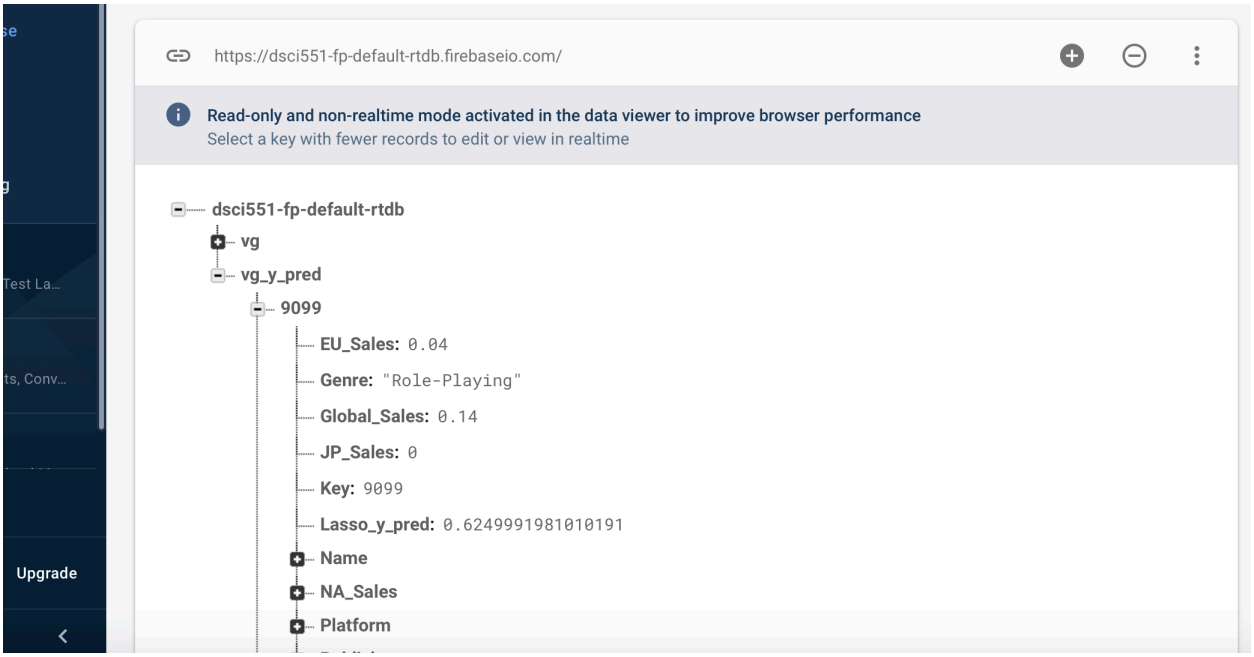
Graph 6: Relationship between Year and Global_Sales

The second topic users could explore is Global sales Prediction based on Publisher, Genre, and Platform. I explored several algorithms by myself including linear regression without cross validation, ridge regression with cross validation, and lasso regression with cross validation (detailed steps and conclusion included in the data_descr.ipynb file). After doing that, I chose the Lasso regression with cross validation to make the prediction, since this algorithm has the best

performances on values of MAE (Mean absolute error: the average of the absolute difference between the actual and predicted values in the dataset) and MSE (Mean squared error: the average of the squared difference between the original and predicted values in the dataset). On the right area, I explored the sales trend lines all over the world. In this case, we can see that the peak values and tendencies are similar, which means that this algorithm could be generalized to predict the sales in different parts of the world.



Graph 7: Global sales Prediction based on Publisher, Genre, and Platform



Graph 8: a screenshot of the firebase after clicking the final uploading button

After all these processes done, there's a button on the bottom of the whole interface to upload the resulted data with additional y prediction column from the second topic into the firebase.

Learning Experiences:

This project was really a challenge for me. I used to write java and r a lot, and this is actually my first semester writing a project using python language. Besides, this is also my first time using Tkinter to create a user interface. I would say that this process was both challenging and interesting for me, and I had a great sense of achievement at the end. Everything in 551 is new for me, I learned how to use aws, firebase, mongoDB, etc. Although I could not practice all the skills I learned from the class, I felt pretty excited. This project is real-life related, and this is exactly what I want to practice. As for the algorithms, since I was a mathematics undergrad student, this part was less challenge for me; however, I also found it interesting while testing different models to predict the results.

Files included in the zip:

. Data directory: Including the sample raw data file (vgsales.csv) and a description file (data_descr.ipynb)

. project_ui.ipynb: A jupyter notebook file including all the functions and UIs for this project, this is the actual file that I run in my computer

. project_ui.py: A .py file containing exactly same contents with project_ui.ipynb

. gaming.jpg: A jpg file, which is the background picture in my UI