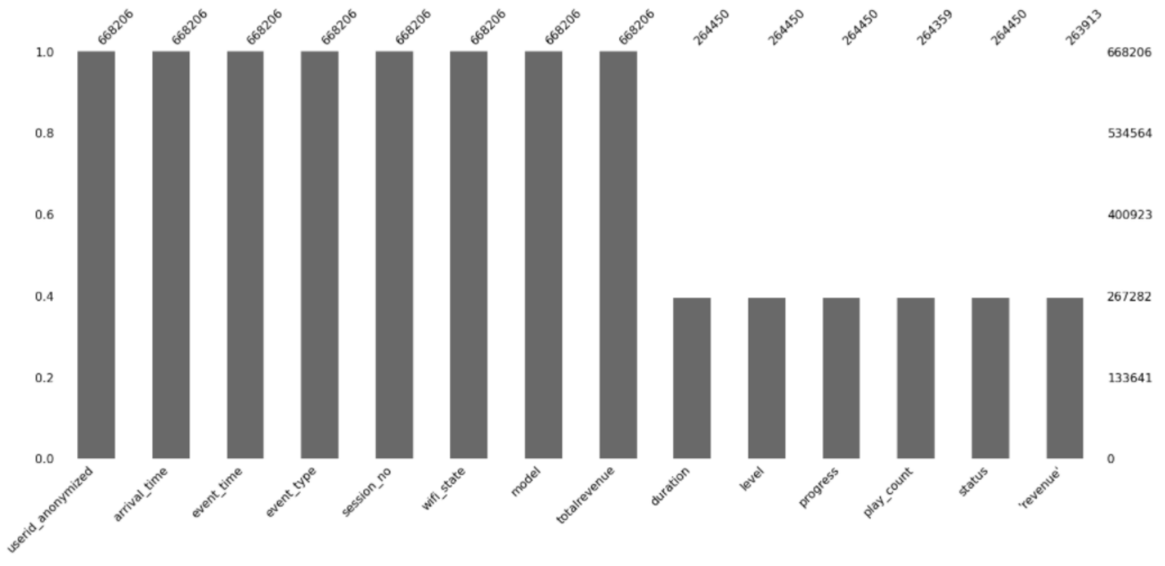
1- Introduction

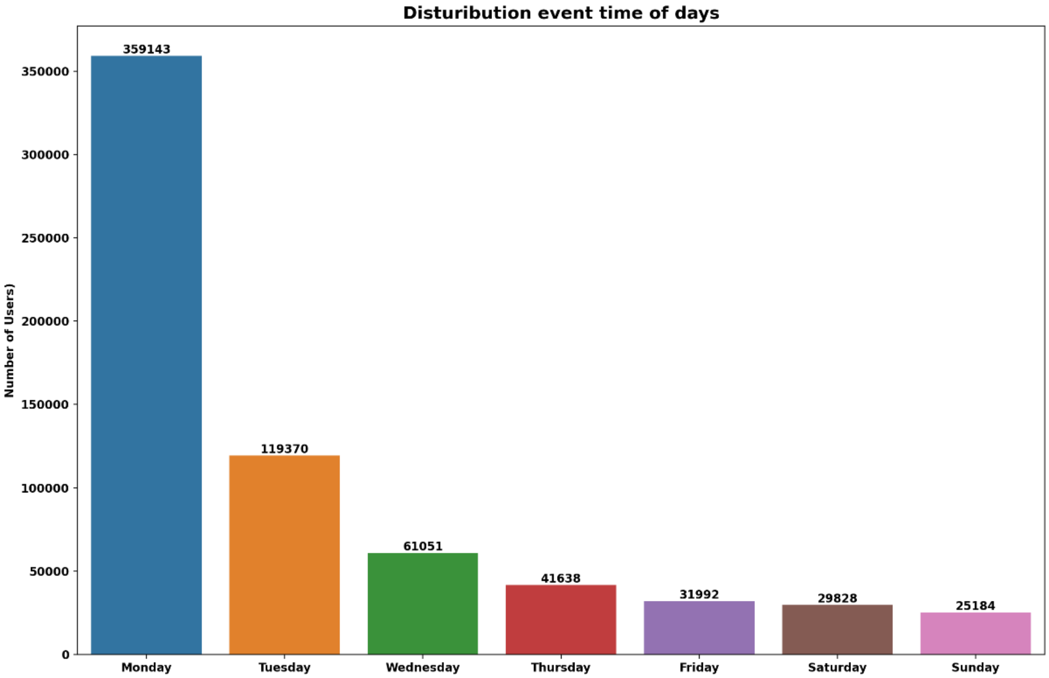
It was nice to dealing with LTV prediction. The most challenging part of me in this project were dealing with understanding the data. There were situations that I couldn't understand about the first three days. I could not fully understand the arrival time and event time explanations. This project helped me to understand that data is not entirely under our control since it is created by users’ decisions, thoughts and behaviors so we should find a way to deal with any condition. I used Linear Regression to predict LTV (Total Revenue) in my model ended up with 0.918 RSquare and 0.153 RMSE

2- Exploratory Data Analysis

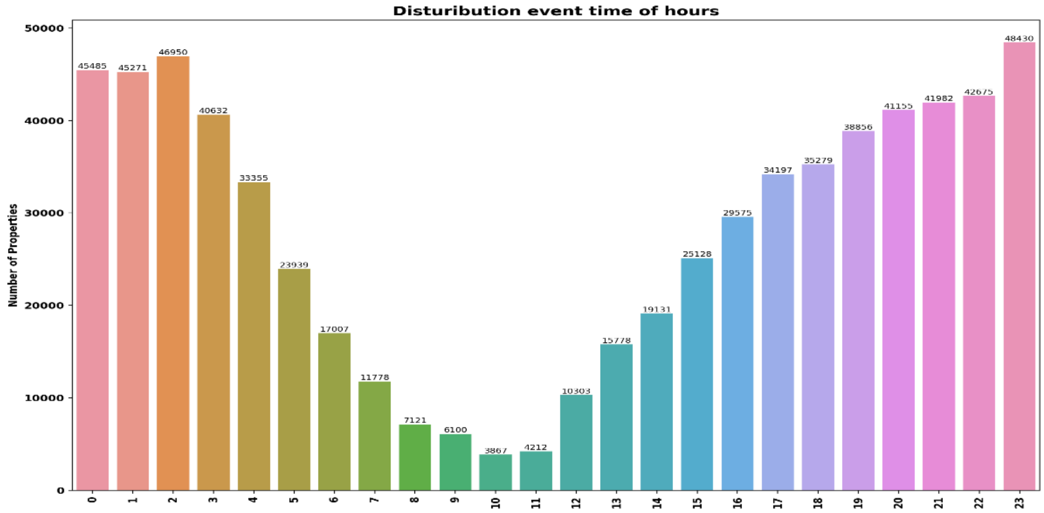
First, I looked at some basic things in the dataset. For example, column and row counts, missing values checking, data types checking etc. I also checked for duplicated rows in the dataset. I cleared these from the dataset. There were no missing values in the data set. I continue to explore the data. Under the event_data column, there was row-based data that would work. I changed the format in which these data are kept and made each of them a feature. I noticed that the new columns have missing value. Normally I wanted to use the KNN-Imputer method. But due to the problem caused by the processor of my computer, I filled it with the median because the computer turns itself off and on due to overheating.



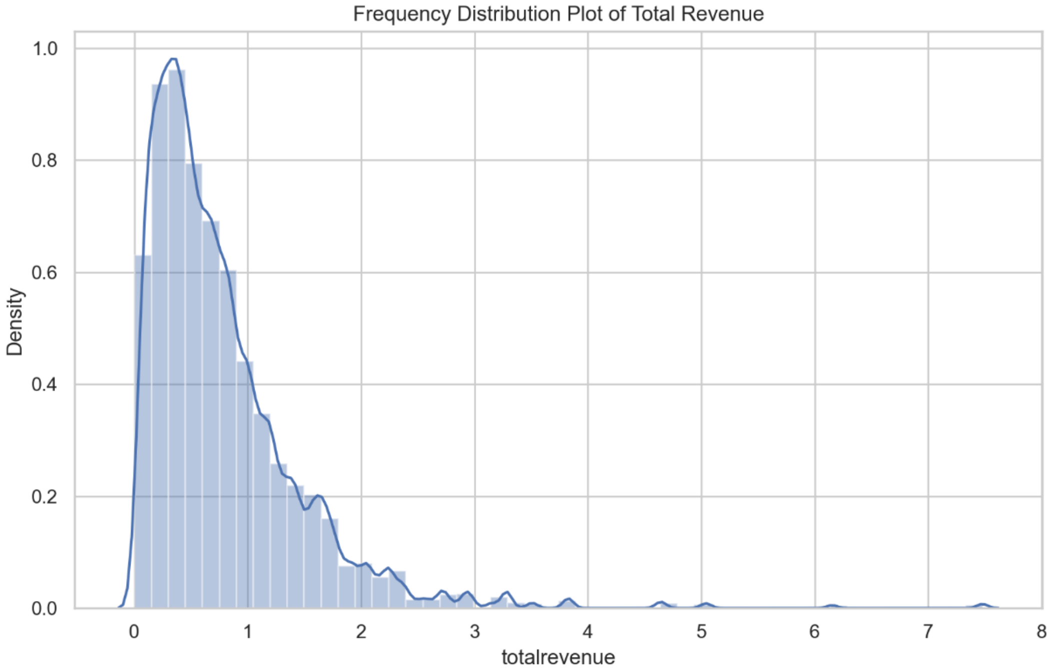
I was wondering about the daily distribution of users. I tried to create a column with the days of the week and analyze it. Monday is the day when users log in the most then comes Tuesday. Looking at the chart, it is decreasing towards the end of the week. Maybe we can generate more income by increasing the ads we will show to the players on Monday.



When we check the hours that users enter the game, we see a bipolar graphic. In the morning hours when the user input is quite low. It starts to increase over time. There is an ever-increasing momentum, especially after 7pm. We can think that people use phones or tabs more for entertainment in the evening.

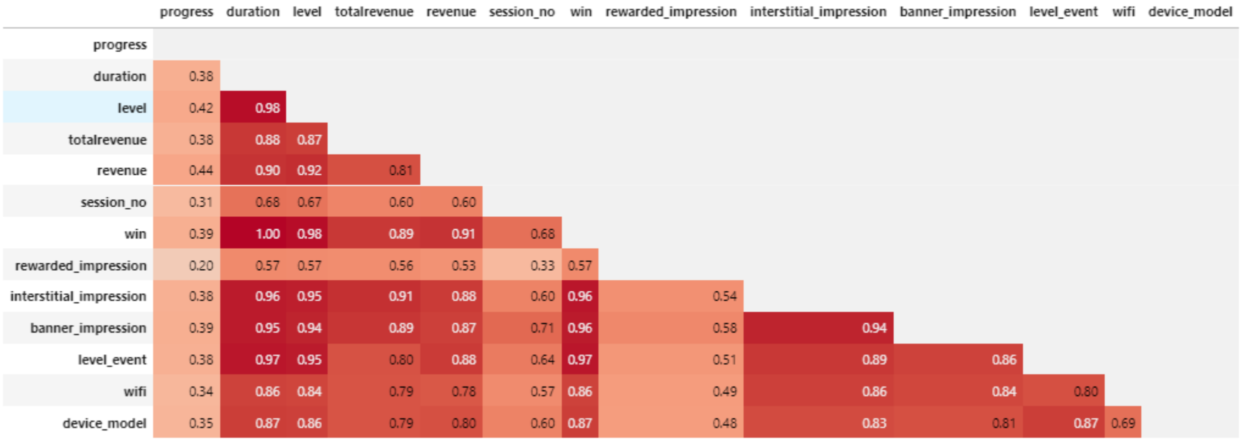


Below is our graph showing the frequency of our total revenue and how it is distributed. Our revenue is concentrated between 0 and 2.



3- Data Preparation and Feature Engineering

I made user ids singular using group by and aggregate function. I used various functions according to the property of each column, for example sum, mean, max, count etc. After feature engineering, I have created scatterplot and correlation heatmap. I cannot put scatterplot because its size is enormous. However, it is in my source code so you can uncomment it any time to investigate. Correlation between duration and win was really high. The more win, the more duration. Interstitial_impression, win and banner_impression features were most correlated variables with target variable (totalrevenue).

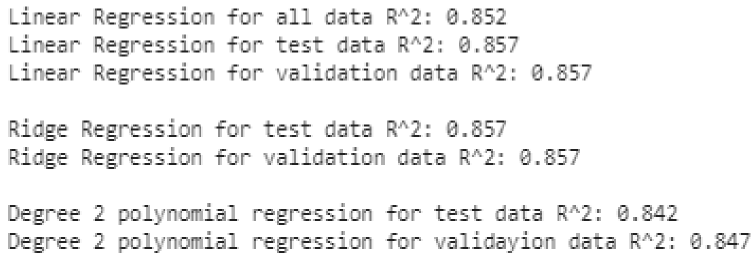


	progress	duration	level	totalrevenue	revenue	session_no	win	rewarded_impression	interstitial_impression	banner_impression	level_event	wifi	device_model
progress													
duration	0.38												
level	0.42	0.98											
totalrevenue	0.38	0.88	0.87										
revenue	0.44	0.90	0.92	0.81									
session_no	0.31	0.68	0.67	0.60	0.60								
win	0.39	1.00	0.98	0.89	0.91	0.68							
rewarded_impression	0.20	0.57	0.57	0.56	0.53	0.33	0.57						
interstitial_impression	0.38	0.96	0.95	0.91	0.88	0.60	0.96	0.54					
banner_impression	0.39	0.95	0.94	0.89	0.87	0.71	0.96	0.58	0.94				
level_event	0.38	0.97	0.95	0.80	0.88	0.64	0.97	0.51	0.89	0.86			
wifi	0.34	0.86	0.84	0.79	0.78	0.57	0.86	0.49	0.86	0.84	0.80		
device_model	0.35	0.87	0.86	0.79	0.80	0.60	0.87	0.48	0.83	0.81	0.87	0.69	

Before entering the model, there were features in the correlation matrix that were highly correlated with each other. Considering the negative impact of these on the model, I decided to apply PCA. PCA is generally used in horizontally large data sets, but it can also be used for features that are overly correlated with each other.

4.Modeling, Quantifying the Model and Result

Firstly, I decided to build a baseline model. It's a good way to both look at the result from the model and see the effect of the features next step. I decided to use the RMSE metric in my baseline model. The RMSE result was 0.198. My goal is to improve this error metric a little more and bring it to a point close to zero. I added new features to my new model and also wanted to look at R-2 using stats model. The results of the model are Root Mean Squared Error: 0.153 and R-2: 0.918. After splitting my dataset, I performed cross validation. I split my dataset into %60 -%20 -%20. %60 training data, %20 test data, %20 for validation. I also used Ridge and Polynomial regression. We can examine the results of the models below.

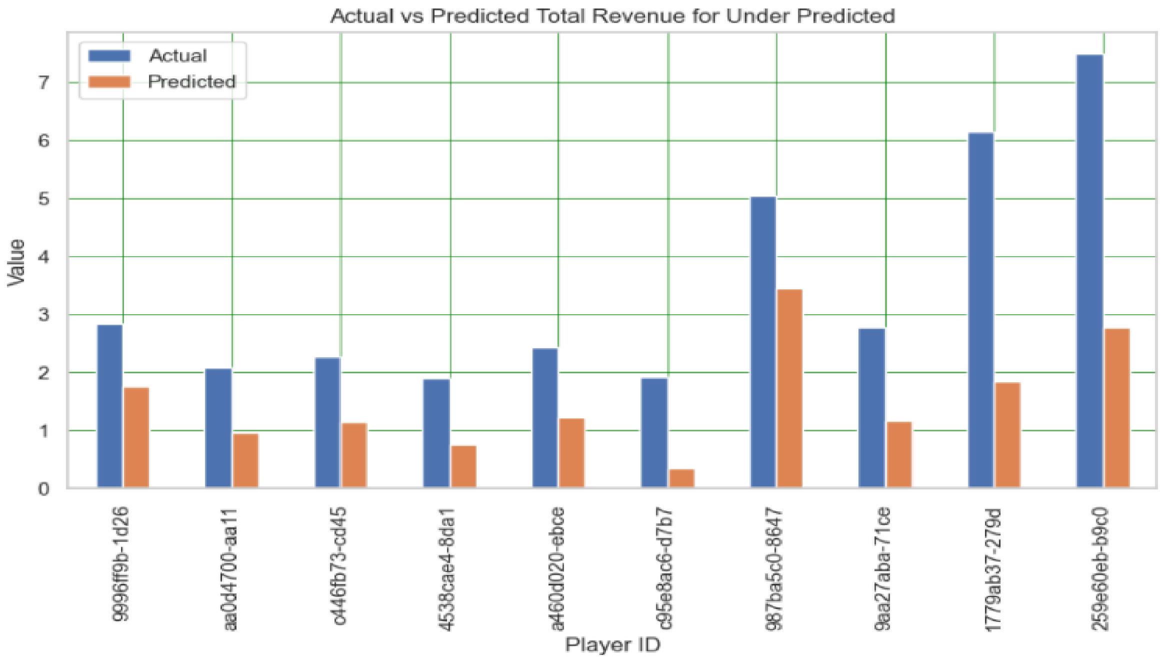


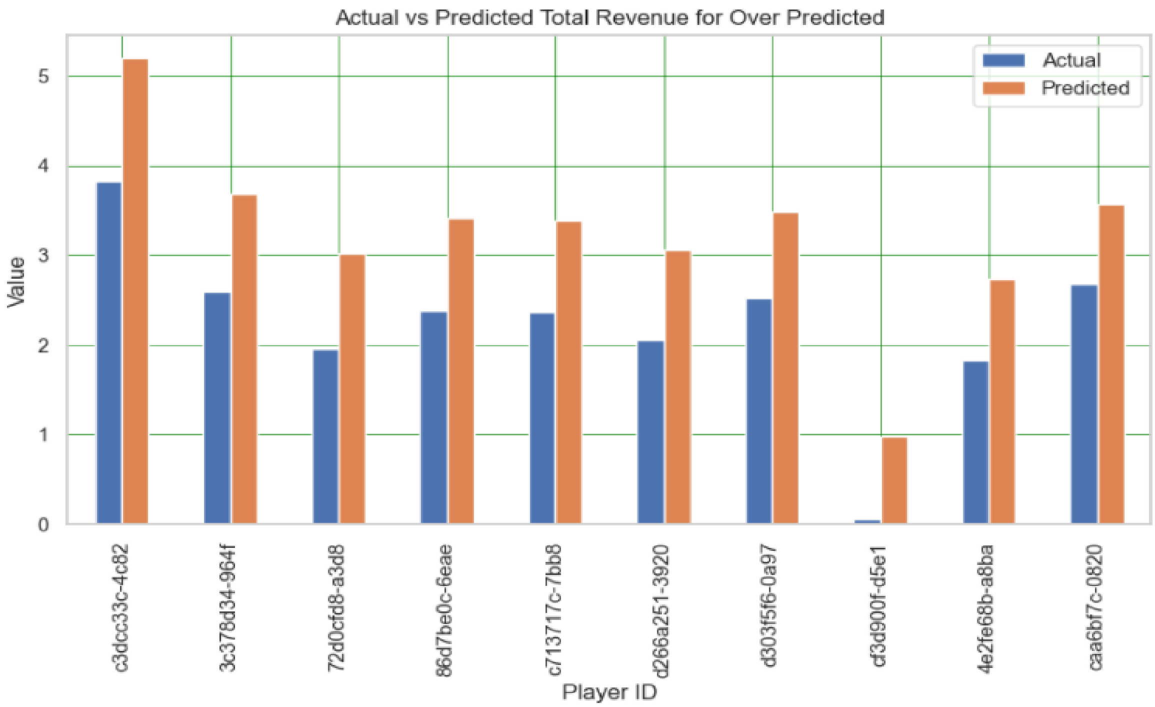
```
Linear Regression for all data R^2: 0.852
Linear Regression for test data R^2: 0.857
Linear Regression for validation data R^2: 0.857

Ridge Regression for test data R^2: 0.857
Ridge Regression for validation data R^2: 0.857

Degree 2 polynomial regression for test data R^2: 0.842
Degree 2 polynomial regression for validayion data R^2: 0.847
```

I wanted to show two graphs the values I estimated with the actual values. One of them is the ones I overestimate the true value and the other is the ones I overestimate the true value.





Before quantifying the model first, we need to understand how and why this model can help us. If we design such a successful LTV prediction system, we can improve our workforce management efficiency and decrease ads cost. I was stuck between the two options; mean absolute error and root mean squared error. These two metrics are used for quantifying linear regression models frequently. My last decision was using RMSE instead MAE. Because of RMSE takes the square of the errors, outliers will have a huge effect on the resulting error. If we make a huge mistake on our predictions, our metric would be much higher so evaluating the model with RMSE more logical to me.

Adnan Kılıç