Mayhem at DinoFun World: Visitor Movement

Amusement parks are awesome. Dinosaurs are awesome. DinoFun World is a fictional Dinosaur themed amusement park that is - you guessed it - awesome! DinoFun World is a modest sized amusement park known for its exciting rides and events. One event is honoring the hometown football super star, Scott Jones. This is a three-day event where Scott Jones would hold two shows a day and talk about his life and in another part of the park to display his sports memorabilia. However, everything did go according to plan as someone from Scott’s past vanished and stole his sports memorabilia. The assignment that was given to me is to characterize the attendance at the park, find any noticeable difference in guest activities over the three day time period, and find any anomalies or unusual patterns with an emphasis on anything that could be related to the crimes on Sunday. This dataset contains all the guest movements and check-in times at DinoFun World over the Scott Jones weekend.

The data comes in three files, one for each day of the Scott Jones weekend. Each file has five columns. The first is timestamp, which is the year, month, day, hour, minute, and seconds which has values from when the first guest enters the park to when the last guest leaves. ID is a unique numerical value to represent each guest in the park. Then there is type, which is either check-in or movement, which indicates whether the guest is moving through the park or checking into one of the attractions. The next two columns are X and Y that relate to the coordinates within the park, which conveniently is 100x100. The records were pretty clean for the number of rows that were in the files 26,021,961. However, in the Sunday file there was one record that had no usable information. Before dropping this record, I tried to see if there was a way to find who this record was associated with. The thought was that since this record was on Sunday, the day of the crime, it could be important. The weirdest part is that the timestamp was in the records much later than the records with the same timestamp. Unfortunately, without any other information, the record had to be dropped. Also, the last row of the Sunday file had the title columns in it, so that was also dropped. Now that the data was cleaned and all the values have been converted into their correct types, it was time to start exploring.

The first task I wanted to tackle was if there was a difference in the guests’ patterns over the course of three days. One reason was because I thought this would be the easiest task, and the second was that I wasn’t quite sure how I was going to complete the first task at this point. I decided that the best way to see this is to compare the movements and check-ins over the three days. To do this, the first thing I did was group all the times for each day into thirty-minute buckets and count the number of IDs for each type. Then, I dropped the date and just kept the time in the for each new timestamp and added the day of the week into a column. After that, I split the data into two data frames: one that had just the type for check-ins and the other for the movements. I then repeated this for each day. Now, the better way to do this would have been to write it as a function and just send each day to the function. If there were any more days that I was looking at I would have started with that, but with only three days in question, copying and pasting code didn’t seem to be a bad idea at first. However, this made the notebook very bloated. If I was starting this project from the beginning again, the functions would have been a much better way to do this. After getting a data frame for each day of either check-in or movement, I concatenated them together and graphed them in a line graph. From the graphs I would say that there is a difference between Sunday and the other two days. It has the greatest number of people in the park out of the three days. The morning starts off almost exactly the same as all the other days. However, once the crime was discovered, check-ins flattened out and became even with Saturday. When it comes to the people moving around the park the only big differences are a large spike in movement when the afternoon show was cancelled. After that it was pretty similar, just a much bigger spike as people leave for the day at 19:30 on Sunday, which brings the number of guests closer to Saturday’s totals. After I started using DBSCAN to look at the groups in the park, I checked that time frame on Sunday to see if that was just a very large amount of people leaving the park at once. Now it was time to start looking at the groups that are in the park.

The first thing that needed to be done was find all the locations that had check-ins. To do this I combined all three days into one data frame filtered by the type check-ins. From there I got all the X and Y coordinates and then removed the duplicates. This gave me forty-two check-in coordinates. I then plotted them in a scatter plot on the park map with their index next to them so that I could create a data frame by hand of the attractions, attraction ID, and type of attraction based on the park map that came with the data to line up properly. Once that was checked multiple times, I merged the data frames to get a table that had the attraction information next to the park coordinates. After completing that it was time to implement some machine learning clustering on the dataset.

I started with the classic K-Means. However, the big issue I kept running into was the timestamps and trying to keep groups together as time moved forward in the day. After trying to learn many new things and not getting the results I was looking for, I decided to bucket all the timestamps into five-minute intervals and run every 5 minutes and look at the difference. At first the K-Means seemed to work, but once the park filled up with guests, K-Means just grouped each section of the park together, which is pretty much how the park is already laid out. I then tried DBSCAN clustering, since it looks at the density of the groups. This got me closer to what I was hoping to achieve, and it also let me see some movement in the park. I began to use this model to find the groups within the park.

I named the first ground I found “the walkers”. These are the guests that have checked into the park but never went on any other attractions. I found this group by first searching to see if there were any guests in the park that never checked into anything, but it appears every guest at least checked into one of the entrances to the park. So, I increased the check-ins by one and found this group of people that only checked into the park. On Friday there were eight of these guests, on Saturday there were ten and on Sunday there were twenty-one. I believe these guests are probably there to assist other guests that have trouble getting around the park, such as chaperones for kids. If these guests are there to help other guests around the park, I would say the park should have a special day where they invite all of these guests back and let them enjoy the rides with shorter lines. The next group that was identified was another uncommon group.

The Scott Jones super fan group went to every Scott Jones show every day. This group consisted of nine people over the three days. They spent around nine to eleven hours in the park every day, and aside from seeing the Scott Jones show, they went on mainly thrill rides and rides for everyone riding Atmofear almost forty times between the nine of them. I made the assumption that they are super fans and not part of the plot to vandalize and steal the sports memorabilia and not gathering information to give to the next group.

The vandals are all named Randal. I put about an entire day’s worth of time into finding this group. The first thing I needed to do was find when the Creighton Pavilion closed before the Scott Jones morning show. I concluded that the crime had to have happened before his afternoon show and during the morning show, as the news article that came with the dataset said that they locked the Creighton Pavilion to allow extra security at the Grinosaurus Stage for the show. So, I began looking for any anomalies on Sunday. To narrow down the window I was looking in, I started to pull data from all the check-ins between 9:00 and 9:15 to see when people started to check into Grinosaurus Stage and stop checking into Creighton Pavilion. After running a few searches, I found that at 9:30, the check-ins at the Creighton Pavilion stop, since that’s the start time of the morning show. Next, I looked to see who was the first to check-in to the Creighton Pavilion after the show ended at 11:30, thinking that they might have been the ones to find the vandalism, but I didn’t really have a way to know that. I then ran a search to see if the last few people who checked-in and tried to see if I could find all the “check-out” by trying to match up the movement of the coordinate right outside of the check-in, but that was as helpful as that movement spot that also picked on everyone walking by at the time. Frustrated and on getting to the point where I thought might be, it was time to try something new. I looked up the four people that checked in at 11:30 and the last few people that checked into the Creighton Pavilion, and that’s when I saw it. The biggest anomaly that leads me to believe that the 3 vandals were Guest 1502920, 416790, and 461004 was their check-in times. They checked into the Creighton Pavilion around 9:00 and didn't have another entry in their movement or check-in until 11:30 right when the Pavilion was unlocked. Guest 1502920 checked in at the same time as guest 416790 and 461004, but then had another check-in at the Creighton Pavilion at 9:30, which I believe was him checking to make sure the Creighton Pavilion was locked while the other three waited for everyone else to leave so they could vandalize and steal the Scott Jones medals. At 11:30, guests 416790 and 461004 left from the entrance, while 1502920 left from the exit. One of the biggest improvements I believe DinoFun World could make would be to install check-out times at every exit like they have for check-in times at every entrance. While you could make a feature that calculates this info based on the first movement after someone checks-in, that will take time out strain on the database and make it so the timestamps are no longer in order. Plus, by adding this somewhat real time security, the park would have known that some guests never checked out the Creighton Pavilion after it was locked, and could have sent someone over there to check what the situation was. As fun as that was, I wanted to find a more pleasant group.

I started by going back to my DBSCAN plots and saw a lot of constant activity in the kiddie land part of the park, which led me to wonder if there is a group that just rides kiddies and nothing else. To do this, I typed out the longest filter to filter out forty-one of the forty-two attractions and then repeated that for each ride and merged the remaining data frames together. I ran each one at a time then added another ride. Once I got to four rides, I had twenty-five results, a reasonable amount to search through to see if they were similar. Turns out, in those twenty-five results, there were 5 families that went around the park and went on rides so that every member of the family could enjoy their day at the park. This is easiest to do with kiddie rides, but I found that you can do this with other ride combinations to find more families. This is probably the most common groups in the park and the target audience. After some time, I thought to myself, I wonder if there were any groups that only checked into two attractions.

Conclusively, there were two groups that only checked into two attractions. The first group only came to park to ride the Scholtz Express, but one of the five members never got off the ride. This was an anomaly in the data, and that is directly related to using apps on phones to track guests’ movements through the park. The most likely situation is that guest 657863’s phone died and no longer was sending the information to the park. I think the park would benefit greatly if it provided wearable rubber bracelets to send the information to the park. You could then use the app to create groups or keep track of your party, load money into the account, and even make transactions. This solves a lot of issues that having a cell in an amusement park can cause. Phones die, get dropped, put in lockers, or are not brought to the park to avoid getting them damaged or lost. DinoFun World provides a loaner device for anyone who needs one, but that is not cost effective. Also, newer iPhones allow you to not send data from certain apps. Having a system that the park has full control over would most likely prevent this kind of anomaly. In case you have another event with a celebrity, the bracelet could track them without putting their personal device on your system.

The only other group that only had two check-ins was the Scott Jones group. Now it is unclear if any of these was Scott Jones himself, his eight bodyguards, or anyone else that he brings with him wherever he goes. This group was highly organized; they arrived every day at the same time and took the same route to the show. They also left at the same time using the same route back, and they did this twice a day except on Sunday when the afternoon show was canceled. Just because it was Scott Jones weekend doesn’t mean everyone was there to see him.

The one-hundred sixty-nine guests in the “not a fan of Scott Jones” group are the people that went to park all three days and never did anything Scott Jones related. They were there to go on the thrill rides and other rides meant for all age groups. I would infer that this group is probably tourists, fans of a rival team, or Americans since Americans don’t watch football like the rest of the world. Even though this group rode almost all the rides over three days, there was one group that did nearly everything in one day.

There was a group of seven friends that showed up to the amusement park right at opening time and stayed until closing time. I have deemed this group the “all-dayers”. On one of the busiest days of the weekend, they managed to ride almost everything in the park. They did avoid the biggest roller coaster in the park, Wrightraptor Mountain, but I would still say they got their money's worth. By the end of the day, they would be sore, dehydrated, and probably sunburned. Hopefully they know the $12.00 refillable fountain drink trick. All soda fountains have water, and by using the refillable souvenir cup for fountain drinks, at $3.00 to $5.00 per water refill, it pays for itself quickly.

This dataset was really enjoyable to work on and has so many things to find within it. It seems simple being only five columns, but it holds many observations. As someone who enjoys amusement parks, working with this data amplifies my desire to attend one. I would love it if there was a weekend before these events and a weekend after to see just how different the Scott Jones weekend was. To complete this project, I spent about five percent of my time preparing the data, and five percent making models. The majority of my time spent was about eighty percent doing data analysis, with the last ten percent trying to learn new things and relearn something that I had not used in a while, or only used once before. One of the things I wanted to learn was how to make all my scatter plots and cluster plots animated, but unfortunately that remains on the list of things to learn. I plan on working with this Mayhem at DinoWolrd dataset a little more in my free time to practice and apply all the skills that I have learned in the program.