Soham | 250123060 | B.Tech. (MnC) 1st Year IITG

FINDINGS REPORT

Based on the EDA and the analysis I did, I had found the following theories from the singer's scribbles worth checking:

1. **Cursed Tuesdays:** The singer emphasized a lot on how Tuesdays are disasters but when I plotted graphs, it turns out that Tuesdays are actually better off than the other working days. What I did find useful was that weekends obviously had better energy levels than the working weekdays.
2. **Brother Thomas' wincing:** I figured out that brother Thomas, as the singer mentioned is a monk or a goth (which I'd just clarify), did not like the louder sounds as such so he winces and when he does, it's all over. So, the thing is he is described as not being able to tolerate the noise, whereas the goth people are quite the opposite, their culture is loud and stuff. Also, he has been given the title of "Brother" clearly a monastic tradition.
   So, I looked for the same in the Holy Grounds (the monastery), what I found was that the energy levels were almost same on volume level 11 as on level 1, it is a bit weird when compared to others so I decided to keep a penalty parameter.
3. **Goth people influenced by time:** The singer mentioned about this so I decided to look into it and indeed after the hour 16 itself the energy levels start rising and they peak at the last hours 22 and 23 hence actually validating his theory that gothic club (Vampire's den) i.e. V_beta people are more energetic in late hours of the day.
4. **Noise influence:** For other venues, namely The Snob Pit and The Mosh Pit the energy increased proportional to the volume levels whereas for the Vampire's Den, the energy remained almost the same regardless of the volume levels.
5. **The Full Moon theory:** The full moon theory is actually very interesting when I plotted the graph, it actually signaled very high energy on the dates when there was a full moon but, here's the catch, only in the daylight, the hours 9 to 13 experienced tremendous increases in the energy when it was a full moon day while at the night time, it had very insignificant effect on energy. So, I decided to keep a feature for it but only for daytime, is_Full_Moon_Daytime.
6. **Outfit's influence:** Singer speculated and thought that outfit matters but in reality, it doesn't when I actually plotted.
7. **Rain preferences:** Singer said rain is bad generally but in the mosh pit it is good but turns out he was just speculating again as there was nothing like that, among

all 4 weathers the distribution of crowd energies was fairly similar no matter the venue.

8. **Snob vs Mosh pricing preference:** The singer was absolutely correct for this part i.e. actually Snob people preferred higher prices and condemned the lower price tickets while the Mosh people hated the higher ticket prices. So, I introduced features tracking the same.
9. **Opener Ratings effect:** Again, the singer was correct this time, Snob people tended to be care more about the opener and their energy increased as the opener rating increased whereas the Mosh peoples' energy remained mostly the same, which is also the same for other 2 venues, people except Snobs don't really care about the opener ratings.
10. **Crowd vs Energy in Mosh:** The singer was not completely correct here as the graphs showed that the energy increases with some weird trends. It is not exponential or something we can actually map using an 'exponential' function as he said, because the graph is really messy, but there is still an overall trend which shows great correlation with a parabolic graph so I tried putting a crowd_size/1000 squared feature that would make it linear for the model.
11. **Weekends:** As I already said, weekends see more energy than normal working days so I made a feature, namely isWeekend to specifically make the model notice the correlation of weekends and energy.

Model Selection

The nature of our dataset is tabular and heterogeneous. It contains different kinds of data, from continuous like Crowd_Size and Ticket_Price to categorical or discrete like Venue_ID and Weather. On top of that, in our dataset, all of the things are context-dependent. The EDA shows it perfectly, for example, the effect of a factor like Ticket_Price or Crowd_Size isn't the same for all the cases, they are dependent on the Venue_ID, some of the trends vary drastically from venue to venue and some might completely reverse.

In such a case, if we used a simpler model like Linear regression, it won't work, because they assume the different features to be independent of each other. Hence, using such a linear model would be insufficient for our case. Also, Euclidian distance-based models like SVM and KNN don't prove to be better in functionality for our particular case as they rely on the 'Euclidian Distance' calculations, but you can't calculate the distance between Venue_ID and similarly Weather.

On the other hand, tree-based models like Random Forest and XGBoost skillfully handle such data as they go on partitioning the data recursively by themselves. A tree would automatically learn to split the data by Venue then for the resulting branches, it uses different set of rules to predict, which is perfect for our needs.

Now, in the tree-based models, we had two prominent choices, RandomForest and XGBoost. Finally, I selected XGBoost for our case because of some significant advantages over Random Forest specifically due to some trends revealed on performing the EDA of the dataset.

- Firstly, XGBoost is a boosting algorithm which trains the decision tree models in series. Each model in the series trains upon its predecessor's mistakes, trying to correct them. Our data does have some anomalies like, the 'Full Moon Daytime' effect which have low frequencies but are significant. If we use Random Forest here, it would average out all the independent decision trees which results into the loss of such effects whereas, XGBoost due to its boosting framework, would specifically target such errors and would learn from them the way we want.
- Further, if you saw we have also used One-Hot encoding which results into multiple zeroes over different columns of the dataset. XGBoost has a 'Sparsity-aware Split Finding' [1] which automatically ignores such null values and hence it has better training speeds and accuracy for our encoded dataset.

References

[1] Chen, T., & Guestrin, C. (2016). **XGBoost: A Scalable Tree Boosting System.** In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785