Predicting apps used by user and change in battery level END TO END project

Data was given by the Uni Bonn CS dept

1. **Data -**

   a) Each day was divided into 48 half an hour slots and for each slot we were given a binary vector which says which apps were in use and were not in use during the time slot. Give an example.
   b) We were also given the change in battery level after the slot, data related to time and whether the phone was charging.
   c) We were also given a column called battery_status which we didn't know what it meant, but figured out later on that it was related to phone charging status

2. **Data Assumptions -**
   a) We assumed that a particular app was running in the full duration of the time slot(as our data was binary and not a fraction)

3. **Cleaning and Feature Engineering (40% of project time) -**
   a) We removed all the rows with null values (<10)
   b) We removed app data for apps which were used >95 % of the time or <5% of the time. This was actually done later on and this improved our model performance. All the apps removed were android based system apps which usually run in the background when the phone is switched on.
   c) We removed erroneous data. (Like sometimes the phone wasn't charging and yet the change in battery level was negative)
   d) Removed outliers. (1970)
   e) We removed rows in which phone was charging (less than 8 % of the data)
   f) Noticed that the time slots were not contiguous and there were periods of time in which data from many days were missing. This was done by introducing a new feature called time which combined all the time related data like date, year, etc

4. **Data Analysis -**
   a) Histogram of which apps were running most frequently.
   b) Relation between battery status and phone charging.
   c) Feature engineered a column for time and graphed.

5. Model for predicting User Profile -

The main model which we used for predicting user profile was the Markov Chain model. Just to show that our Markov Chain model is doing good we also used a baseline model and showed that our complex model performed better than this.
Assumptions of the Markov Chain model-

1) The Probability that we will go from state a to state b is dependent on only state a and independent of the past. (For example the probability that we will use spotify a time step from now given that we are using spotify rn is independent of us using spotify a time slot back)

Math behind Markov Chain model -
P(a|b) = no of jumps from a to b / No of visits to b
Pro of using this model -:

1) Simple model with high interpretability.
2) Works with non contiguous time slots

5. Regression Model for predicting change in battery level given which apps were in use in that time slot -
We used a linear regression model so as to have a higher degree of interpretability, for example we wanted to know which apps drain the most battery etc

We performed leave one out validation
Checked for overfitting or underfitting. Not surprisingly, the linear model had a high bias and low variance.

6. Integrated the 2 models
Given the training data we used the markov chain model to predict which apps user will use in the future and the regression model to predict change in battery level given that the predicted apps were in use.

Notable Achievements-

1) Built all models from the ground up to really understand the math behind the models.
2) Used good software design structure inspired from Sci-kit learn and had a class for linear regression and markov chain model prediction with methods like model.fit and model.predict.
3) Even intermediate functions which graph errors and do validation were quite general