DM1*

Tim Mooren[1 [11710160]], Second Author[1 [1111−2222−3333−4444]], and Third Author[1 [2222−−3333−4444−5555]]

Vrije Universiteit Amsterdam, The Netherlands

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Task 1: DATA PREPARATION

1.1 TASK 1A: EXPLORATORY DATA ANALYSIS

Data properties & variables

Data was acquired in 'long form', where each row consisted of the following: a *time entry*, an *id*, a *variable* and a *value*. In total, there were 376912 entries in the dataset amassed by 27 users, of which 202 had explicitly indicated missing values (denoted 'NA'), belonging to the variables *circumplex.arousal* and *circumplex.valence*.

Time entries were not consistent across variables, since the variables had a different collection methods (eg. manual input, automatic collection), meaning that certain variables were collected much more frequently and with shorter time intervals between each entry. For example, a variable such as 'screen' received many entries since an entry was created every time a user was on his phone, whereas the 'mood' variable was manually collected around 4 times a day. This discrepancy in collection methods created a dataset with very inconsistent timestamps. Indeed, of the 376913 total rows, 336907 of them were unique time entries.

We can call 'time based' variables the ones that were automatically collected by user phone usage (ie. all *appCat* variables), 'score based' variables the ones in which the user inputs a score (eg. *mood*), and 'incidence based' variables the ones where a boolean value was automatically collected (*call* & *SMS*).

More details on each variable and their properties can be found in *Table...*

+ table, when I manage to format it right... fucking latex always being super cringe

Frequency distributions

To get an idea of the distribution of the data, each variable was plotted as a frequency distribution histogram. We can see that most of the time based variables follow roughly a power-law distribution, where the majority of the time entries are short lived, but there are a few very long ones. On the other hand, score based variables seem to be normally distributed, which makes sense

* Supported by organization x.

for this kind of data. It might be important to note that a normally distributed target variable (*mood*) may lead to class imbalance in later models, where low and high mood score are not as

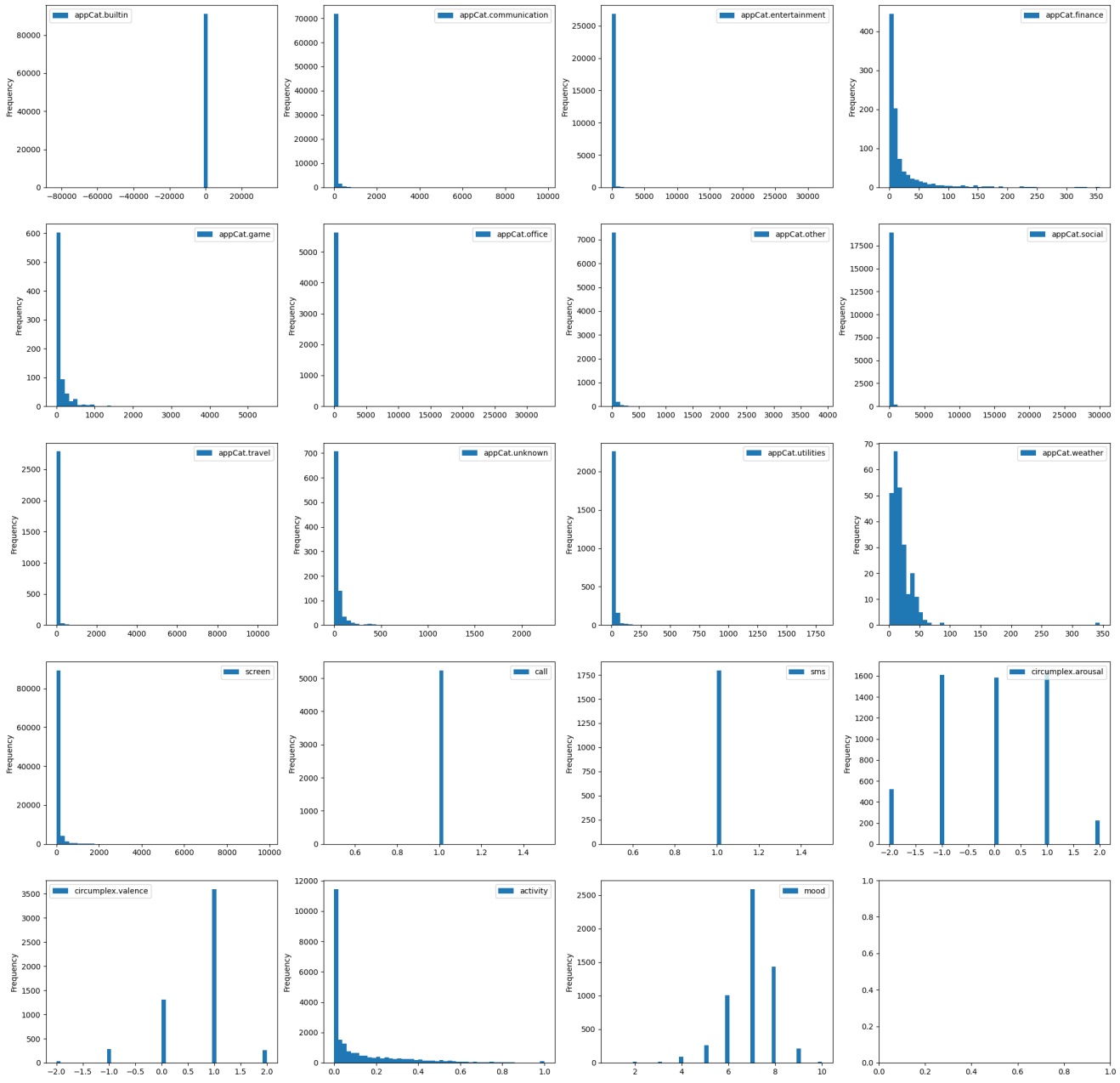

Fig. 1. Caption

Data trends over time

In order to get a better idea of the general time-series data trends, each variable was plotted over the (roughly) 4 month data collection period. For better visualization, the data was aggregated into days by taking the mean of all values in that day for each variable.

Overall, there seem to be no long term changes in *mood* or any of the other predictor variables. We also see that there seem to be no clear linear or seasonal trends, but some somewhat cyclical behavior for certain variables and mostly irregular behavior for others.

For a lot of the time based variables (*appCat* variables), the data seems to have sudden spikes in usage, with a quick return to baseline. Score variables such as *mood* or *circumplex* are more-so characterized by cyclical peaks and valleys. Incidence variables (*Call* and *SMS*) seem to drop off over time, but this could either be due to these events not occurring or these events not being recorded.

Additionally, we noticed that there was around a 14 day period at the start of the graphs where a large majority of variables values were missing, except for in *call* and *SMS*.

1.2 TASK 1B: DATA CLEANING

Extreme and Incorrect Values

Through the exploratory data analysis we discovered that 4 entries contained incorrect values in *appCat.builtin* & *appCat.entertainment*, ie. they were negative when expected to be positive. Entries containing these values were removed from the dataset.

Data aggregation

In order for the data to be in suitable form for the later steps, the data was first put in 'wide form', where each column contained the variables and each row contained the values. Since there were so many unique time entries, this created a lot of sparsity in the dataset, where a unique timestamp for one variable meant that each of the other variables would now have empty values.

Because of this, all data was aggregated per day. The data was first aggregated by day and by individual ('id'), where all time based variables were aggregated with the sum, as a cumulation of the daily time allocated to that variable, and where all score based variables were aggregated with the mean, since the score isn't supposed to cumulate.

After this first aggregation, we noticed that there were still a lot of empty values in the dataset and decided to aggregated a second time by merging the individuals together. Since the time based variables had already been cumulated by day, we could aggregate using the mean for all variables.

The remaining dataset was a table containing 113 rows, where each row corresponds to a date instance with all the values for all variables after both aggregations.

It could have been possible not to do a second aggregation round in order to have more instances but we deemed that this would still leave too many missing values to impute, so we decided to make the trade-off in favor of having less empty values. It may also be important to note that each day is not necessarily represented equally after the aggregation, meaning that some days had

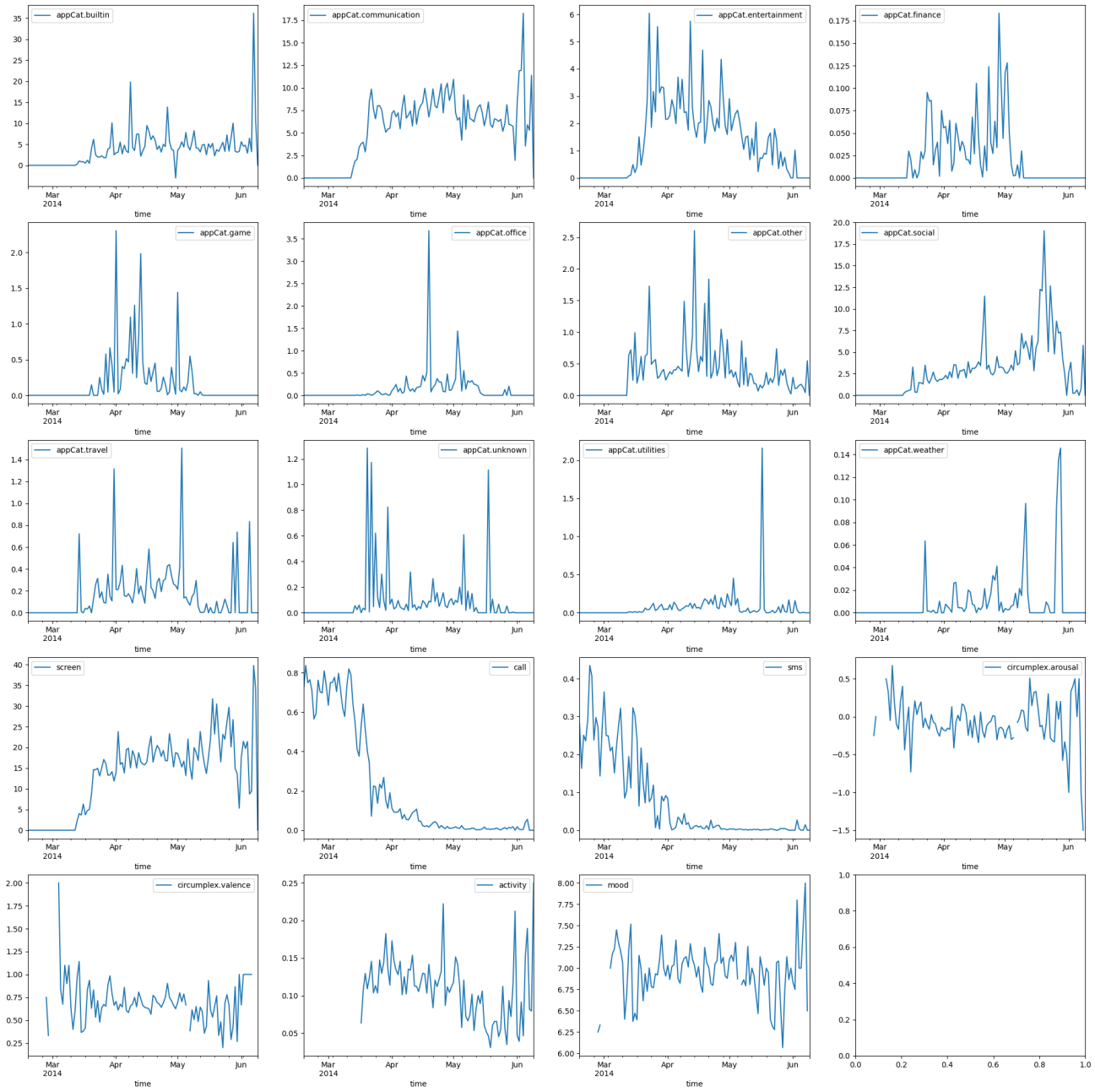

Fig. 2. Caption

a lot more values to aggregate over because of the date and method inconsistencies in the way the data was collected.

Imputation & removal

The missing values from variables 'circumplex.arousal' and 'circumplex.valence' were imputed using two methods. The first method involved replacing the missing values with the mean of the variable per participant per day. It was decided to use this method rather than the overall mean because it was expected that this would result in a closer approximation to the actual value of those entries.

second method is mean per person.

1.3 TASK 1C: FEATURE ENGINEERING

Frequency Count

Normalization

Weekdays
One feature that was added for the purpose of temporal mood prediction is the day of the week, which was constructed from the original data. This might be a valuable feature as it is reasonable that individual's mood may depend on the day of the week. For example, people may be in better mood on Saturday than on Monday.

Participant's Average Mood
PARTICPANT'S AVERAGE MOOD CAN BE IMPLEMENTED MAYBE?

2 TASK 2: CLASSIFICATION

2.1 TASK 2A: APPLICATION OF CLASSIFICATION ALGORITHMS

2.2 TASK 2B: WINNING CLASSIFICATION ALGORITHMS

3 TASK 4: Numerical Prediction

4 TASK 5: EVALUATION

4.1 TASK 5A: CHARACTERISTICS OF EVALUATION METRICS

Mean Squared Error (MSE) and Mean Absolute Error (MAE) are two common error measures used to evaluate the performance of a model in regression tasks. The corresponding formulae for these error measures are as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

where y_i denotes the true value of the target variable, $\hat{y}_i$ represents the predicted value of the target variable, and N is the number of samples in the dataset.

A researcher might choose to use one error measure over the other based on the characteristics of the problem at hand and the desired properties of the error metric. The key differences between the two error measures are outlined below:

Sensitivity to Outliers: The MSE metric is more sensitive to outliers than MAE. This is because the squared term in the MSE formula magnifies the errors for large deviations. As a result, MSE tends to penalize large errors more heavily. If the researcher wants to emphasize the importance of fitting large errors correctly, they might choose MSE over MAE. Conversely, if the goal is to minimize the influence of outliers, MAE may be preferred.

Differentiability: MSE is a differentiable function, while MAE is not differentiable at all points due to the absolute value operation. In the context of optimization algorithms, this difference is important. Algorithms that require gradient information, such as gradient descent, benefit from using MSE since its gradient can be easily computed. On the other hand, MAE might be used with optimization algorithms that do not require gradient information.

An example situation where using MSE or MAE would yield identical results is when the errors are either all positive or all negative, with equal magnitude. For instance, consider a dataset with true target values $y = [1, 2]$ and predicted values $\hat{y} = [2, 3]$. In this case, the errors are $[-1, -1]$, and the absolute values of the errors are $[1, 1]$. Computing MSE and MAE gives:

$$MSE = \frac{1}{2}\left((-1)^2 + (-1)^2\right) = \frac{1}{2}(1+1) = 1$$

$$MAE = \frac{1}{2}(|-1| + |-1|) = \frac{1}{2}(1+1) = 1$$

References

1. Author, F.: Article title. Journal **2**(5), 99–110 (2016)
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
5. LNCS Homepage, http://www.springer.com/lncs. Last accessed 4 Oct 2017