Data Mining Techniques - Assignment 1

Filip Muntean[1] and Marin Marian[1]

Vrije Universiteit Amsterdam
Group 21
{f.muntean,m.marian}@student.vu.nl

Abstract. The development of smartphone apps designed to assist those with depression is a major turning point in the quickly expanding field of mental health. These apps employ user-submitted mood ratings in addition to a range of sensory data to track user behaviour. An abstract of a large dataset that was collected from these apps is presented in this work. The dataset includes time-stamped user IDs, mood scores, and other behavioural variables including activity levels and app usage. This dataset's main goal is to make it easier to anticipate future mood states using past user data in an effort to provide a predictive lens on mood changes. Our dataset provides a comprehensive picture of user behaviour and its possible effects on mental health by encompassing a wide range of variables, from simple mood surveys to complex measures of app usage across multiple categories. The goal of the project is to turn this raw data into a reliable mood prediction model by means of rigorous data preparation, which includes cleaning, feature engineering, statistical analysis, and exploratory analysis. The ultimate objective is to improve mental health outcomes by utilising the predictive capacity of the information to enable early interventions and individualised support mechanisms. This project sheds light on the crucial relationship between technology and mental health, while also creating new opportunities for predictive health analytics study and implementation. A solid framework is provided by pertinent studies, such as those on the use of mobile phones in mental health by Torous et al.[Torous et al.(2017)] and the investigation of predictive models by Saeb et al. [Saeb et al.(2015)]. These studies highlight the advantages and disadvantages of using technology in mental health interventions.

Keywords: Data Mining · Users's Mood · Another keyword.

1 Introduction

This paper focuses on a novel dataset derived from such applications, specifically designed to track various behavioral and emotional metrics of individuals suffering from depression. The dataset is rich with sensory data and self-reported mood ratings, collected with the ultimate aim of predicting the user's mood for the subsequent day. This predictive capability holds the potential to transform mental health care by enabling preemptive support strategies and personalized

therapeutic interventions, thus offering a more dynamic and responsive approach to managing depression.

The dataset under analysis comprises time-stamped entries of user IDs, mood ratings, and other behavioral variables such as activity levels, screen time, and app usage across various categories. These entries provide a multifaceted view of the user's daily life, offering insights into the intricate relationship between mood and behavior.

The goal of our study is twofold: firstly, to perform an exhaustive exploration and cleaning of this dataset to prepare it for analysis, and secondly, to develop and evaluate predictive models capable of forecasting the next day's mood based on the data provided. By achieving this, we aim to contribute valuable knowledge to the fields of mental health and digital therapeutics, demonstrating the potential of smartphone applications as a tool for mental health care.

The paper the paper is organized as detailed below:

Section 2, Data Preparation, outlines the initial steps taken to render the dataset suitable for analysis. This section is subdivided into three parts: 2.1 Exploratory Data Analysis, where we examine the dataset to understand its characteristics, including the distribution of values and missing data; 2.2 Data Cleaning, which details the methodologies employed to address issues of noise, outliers, and missing values, thereby refining the dataset; and 2.3 Feature Engineering, discussing the creation of new data features from existing variables to enhance the predictive model's efficacy.

Section 3, Classification, shifts focus to the application of machine learning techniques for mood prediction. 3.1 Application of Algorithm presents the classification algorithms tested, along with the process of optimizing their parameters for best performance. 3.2 Winning Classification Algorithm reflects on the comparative effectiveness of the algorithms employed, identifying the most successful approach based on our analysis criteria.

Section 4, Association Rules, explores the identification of patterns within the dataset that predict mood states, employing association rule mining to uncover significant relationships between different behavioral indicators.

Section 5, Numerical Prediction, extends the analysis to numerical forecasting, applying regression models to predict the quantitative aspects of mood states, thereby complementing the classification approaches discussed previously.

Section 6, Evaluation, critically examines the evaluation metrics used to assess the performance of the predictive models. This section is divided into 6.1 Characteristics of Evaluation Metrics, discussing the theoretical underpinnings and practical considerations of the chosen metrics, and 6.2 Impact of Evaluation Metrics, which analyzes how different metrics influence the interpretation of model performance.

The paper concludes with Section 7, Conclusion and Future Work, summarizing the key findings of our research and outlining potential avenues for further investigation. Here, we reflect on the implications of our work for the development of mental health applications and the broader landscape of digital health

care, emphasizing the contribution of our study to the field and proposing directions for future research.

2 Data Preparation

2.1 Exploratory Data Analysis

To begin, we obtain a general overview of the dataset, which is comprised of user-specific entries each annotated with a timestamp, a variable indicative of a behavioral metric, and its corresponding measured value. Our data comprises 5 columns, with *Unnamed:0*, likely a byproduct of data indexing in its source format, which we elected to remove to streamline the dataset for analysis. Upon initial import, the dataset presents itself in a tabular form with a shape of 376912 $\times$ 5, denoting a substantial volume of 376912 individual records spread across 5 distinct columns. Our columns are described as such below:

1. **id**: A unique identifier for each user, ensuring individual-level analysis while maintaining anonymity. Within the dataset, each user is assigned a unique identifier following a standardized format, exemplifying the convention AS14., where * represents a sequential number unique to each participant, ranging from 1-33.
2. **time**: The timestamp of the recorded activity, precise to the second, which is crucial for tracking mood fluctuations over time.
3. **variable**: A categorical representation of the measured behavior, ranging from mood scores to various app usage metrics.
4. **value**: The quantified measure associated with each variable, serving as the critical data point for our predictive modeling.

	Unnamed: 0	id	time	variable	value
count	3.769120e+05	376912	376912	376912	376710.000000
unique	NaN	27	336907	19	NaN
top	NaN	AS14.01	2014-04-14 12:00:00.000	screen	NaN
freq	NaN	21999	91	96578	NaN
mean	4.501273e+05	NaN	NaN	NaN	40.665313
std	5.411519e+05	NaN	NaN	NaN	273.726007
min	1.000000e+00	NaN	NaN	NaN	-82798.871000
25%	9.422875e+04	NaN	NaN	NaN	2.025000
50%	2.274385e+05	NaN	NaN	NaN	7.029000
75%	5.160412e+05	NaN	NaN	NaN	29.356000
max	1.427711e+07	NaN	NaN	NaN	33960.246000

Fig. 1: Statistics of the dataset

Based on the statistics of the dataset, we can interpret the following:

1. **Count**: As mentioned beforehand, there are 376,912 entries for each of the id, time, and variable columns, indicating a complete dataset with no missing entries for these attributes. The value column contains some null entries, which will be handled during the cleaning process. Overall, the complete count for most columns suggests a high level of data collection comprehensiveness, integral to ensuring a quality analysis
2. **Uniqueness**: Out of the entire dataset, there are 27 unique user IDs, suggesting a modest but potentially sufficient number of subjects for robust analysis. The timestamp entries are notably diverse, with 336,907 unique time points, underscoring the fine granularity and potential for high-resolution temporal analysis. There are 19 unique variables captured, each likely representing a different aspect of user behavior or interaction with the mental health application. The high frequency of data points for certain users, like AS14.01, indicates that data density could vary significantly across individuals, which may influence the predictive model's performance and may necessitate stratified sampling or weighted analysis to normalize this effect.
3. **Temporal Granularity**: The most frequently occurring user ID is AS14.01, appearing 21,999 times, which may indicate more active participation or a longer tracking period for this user. The timestamp '2014-04-14 12:00:00.000' is the most common, with 91 occurrences, and 'screen' is the variable with the highest frequency at 96,578 instances. The granularity indicated by the 336,907 unique timestamps reflects detailed longitudinal data capture, which is beneficial for modeling time-dependent phenomena such as mood fluctuations
4. **Behavioural insights**: The mean value for the value column is approximately 40.67, with a standard deviation of 273.73, hinting at a broad spread of the data around the mean. This wide variation suggests the presence of high variability or outliers in the behavioral metrics recorded.

Columns Based on a deeper dive, we obtain the following percentages of unique values in our columns:

1. id - percentage: 0.007%. The low uniqueness in the 'id' column confirms that the data is rich in longitudinal measurements rather than cross-sectional, which can significantly enhance the model's capacity to make individualized predictions.
2. time - percentage: 89.386%. The high percentage of unique timestamps signifies a fine granularity in data collection over time. Such detailed tracking can serve our goal of predicting a user's mood for the next day.
3. variable - percentage: 0.005%
4. value - percentage: 34.652%. This moderate percentage of unique values indicates a substantial variety in the data captured by the behavioral metrics. It is neither too high to suggest sparse data nor too low to indicate redundancy.

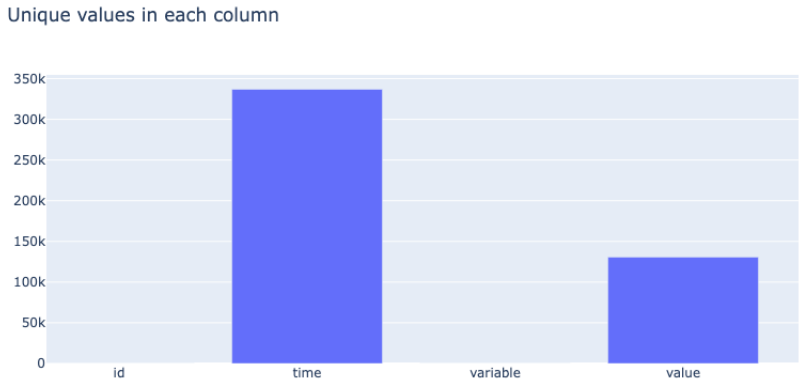


Fig. 2: Unique Values per Column

With regards to the column containing the values, we plot the latter over time to gain insights into how this column evolves.

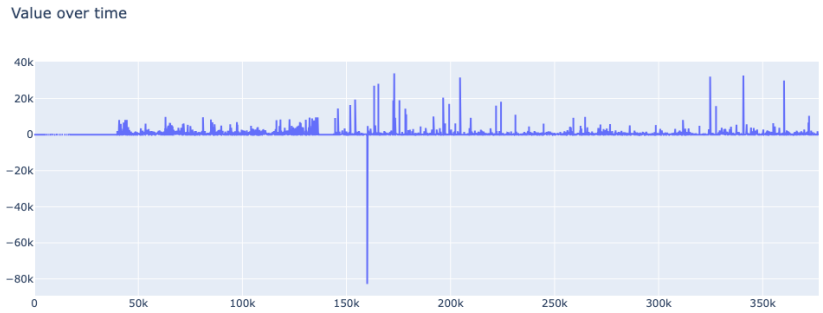


Fig. 3: Evolution of the Value over Time

Based on the plot above, we notice significant fluctuations over time with a constant frequency of spikes, suggesting regular changes in the variable being measured. Outliers or anomalies in the data, such as significant drops below zero, could be due to measurement errors or extreme events. There is no clear upward or downward trend across the entire dataset, suggesting no long-term increase or decrease over time. Data density indicates periods with more dense data points, suggesting more activity or frequent measurements. The values range from -80k to +40k, suggesting wide variability in the data. The time intervals, labeled

with numbers up to 350k, indicate uniform data collection over time. No clear seasonality is observed.

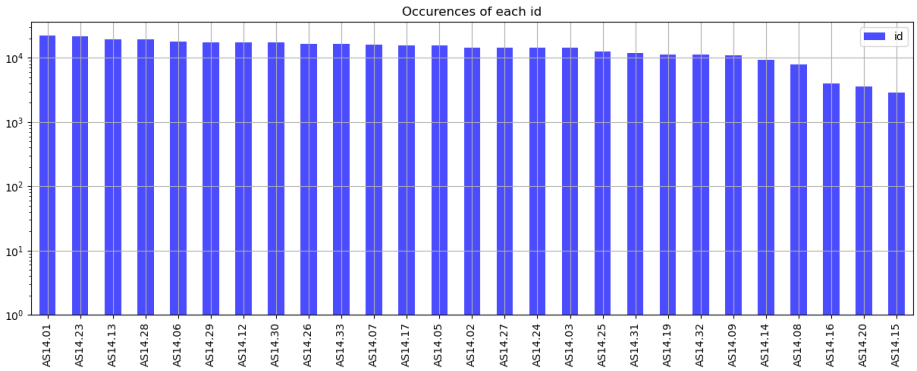


Fig. 4: Occurences of each ID

When focusing on the Id's, we have decided to visualize them on a logarithmic scale, since in this scenario it helps us to visualize the differences among smaller values while still accommodating larger values without letting them dominate the plot visually. In short, in our dataset, using such a scale is useful as it allows for better visibility of variations across the entire range of the data. We notice based on our distribution that a number of values have above 10^4 occurrences, signaling that these particular IDs might be associated with very common events or entities within our timestamps.

Variables Based on the statistics of our variables, we conclude the following:

1. **Count**: The 'count' column shows the number of non-null observations for each variable. The 'screen' variable has the highest count, indicating it's the most recorded/observed feature in the dataset.
2. **Mean**: The 'mean' value provides the average of each variable. For instance, 'mood' has a mean of approximately 6.99, while 'circumplex.arousal' has a slightly negative mean, indicating that on average, the recorded arousal levels are slightly below a neutral baseline set at zero.
3. **Min and Max**: These columns indicate the range of the data, from the lowest to the highest value recorded for each variable. For 'appCat.builtin', there's an extremely low minimum, which represents an outlier
4. **Percentiles**: These provide a view of the distribution of the data. For example, 50% (which is the median) of the 'activity' records are below 0.158, while 95% are below 278.783.

	count	mean	std	min	5%	10%	25%	50%	75%	95%	max
mood	5641.0	6.992555	1.032769	1.000	5.00000	6.0000	7.00000	7.000000	8.000000	8.000000	10.000
circumplex.arousal	5597.0	-0.098624	1.051868	-2.000	-2.00000	-1.0000	-1.00000	0.000000	1.000000	1.000000	2.000
circumplex.valence	5487.0	0.687808	0.671298	-2.000	-1.00000	0.0000	0.00000	1.000000	1.000000	1.000000	2.000
activity	22965.0	0.115958	0.186946	0.000	0.00000	0.0000	0.00000	0.021739	0.158333	0.531195	1.000
screen	96578.0	75.335206	253.822497	0.035	1.14400	1.9150	5.32225	20.044500	62.540250	278.783150	9867.007
call	5239.0	1.000000	0.000000	1.000	1.00000	1.0000	1.00000	1.000000	1.000000	1.000000	1.000
sms	1798.0	1.000000	0.000000	1.000	1.00000	1.0000	1.00000	1.000000	1.000000	1.000000	1.000
appCat.builtin	91288.0	18.538262	415.989243	-82798.871	1.00300	1.0560	2.02000	4.038000	9.922000	42.553600	33960.246
appCat.communication	74276.0	43.343792	128.912750	0.006	1.96400	3.0060	5.21800	16.225500	45.475750	149.610750	9830.777
appCat.entertainment	27125.0	37.576480	262.960476	-0.011	0.58800	0.8800	1.33400	3.391000	14.922000	156.836200	32148.677
appCat.finance	939.0	21.755251	39.218361	0.131	2.00490	2.9760	4.07200	8.026000	20.155000	90.180400	355.513
appCat.game	813.0	128.391615	327.145246	1.003	4.01080	5.1610	14.14800	43.168000	123.625000	496.865800	5491.793
appCat.office	5642.0	22.578892	449.601382	0.003	1.00200	1.0180	2.00400	3.106000	8.043750	54.038300	32708.818
appCat.other	7650.0	25.810839	112.781355	0.014	2.00645	3.1267	7.01900	10.028000	16.829250	71.381850	3892.038
appCat.social	19145.0	72.401906	261.551846	0.094	1.93600	3.0180	9.03000	28.466000	75.372000	267.712800	30000.906
appCat.travel	2846.0	45.730850	246.109307	0.080	1.41350	2.1160	5.08650	18.144000	47.227250	124.898000	10452.615
appCat.unknown	939.0	45.553006	119.400405	0.111	1.00590	1.7290	5.01800	17.190000	44.430500	155.771000	2239.937
appCat.utilities	2487.0	18.537552	60.959134	0.246	1.00800	2.0050	3.15850	8.030000	19.331000	54.324900	1802.649
appCat.weather	255.0	20.148714	24.943431	1.003	2.02350	4.1014	8.68400	15.117000	25.349000	45.795400	344.863

Fig. 5: Statistics of the Different Variables in the Dataset

2.2 Data Cleaning

Before proceeding with our data cleaning, we have first dropped the column *Unnamed: 0*, since, as mentioned above, we have not deemed it useful for our process. Moreover, we have sorted our data based on the columns *id, time, variable* which will ease our process further. Then, we reorganized our data so that each of our unique variables became a column. Furthermore, we renamed these columns for better readability and handling of these columns.

```
for feature in data['variable'].unique():
    data[feature] = data['value'][data['variable'] == feature]
data.rename(columns={
    'circumplex.arousal':'arousal',
    'circumplex.valence':'valence',
    'appCat.builtin':'builtin',
    'appCat.communication':'communication',
    'appCat.entertainment':'entertainment',
    'appCat.finance':'finance',
    'appCat.game':'game',
    'appCat.office':'office',
    'appCat.other':'other',
    'appCat.social':'social',
    'appCat.travel':'travel',
    'appCat.unknown':'unknown',
    'appCat.utilities':'utilities',
    'appCat.weather':'weather'}, inplace=True)
```

Remove extreme values From the statistics above, we can see that the screen, builtin, communication, entertainment, finance, game, office, other, social, travel,

unknown, utilities, and weather columns may have extreme outliers. So, we focused on them to move on with data cleaning. We have opted for 2 data cleaning approaches namely IQR and Isolation Forest. The initial dataset contained 376,912 observations. After cleaning with the IQR method, 35,669 observations considered outliers were removed, resulting in a dataset with 341,243 observations. The Isolation Forest method resulted in a slightly larger dataset with 343,779 observations after outlier removal, indicating a less aggressive removal approach, as can be seen in our result plots below. We have opted for the Isolation Forest method, as we wanted to capture all the intricacies from our dataset.

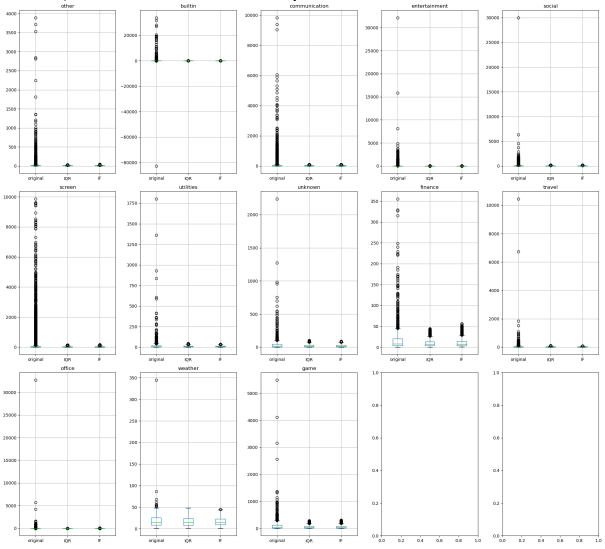


Fig. 6: Results of our Data Cleaning

Imputing missing values First, we wanted to visualize our missing values, which we have done as follows:

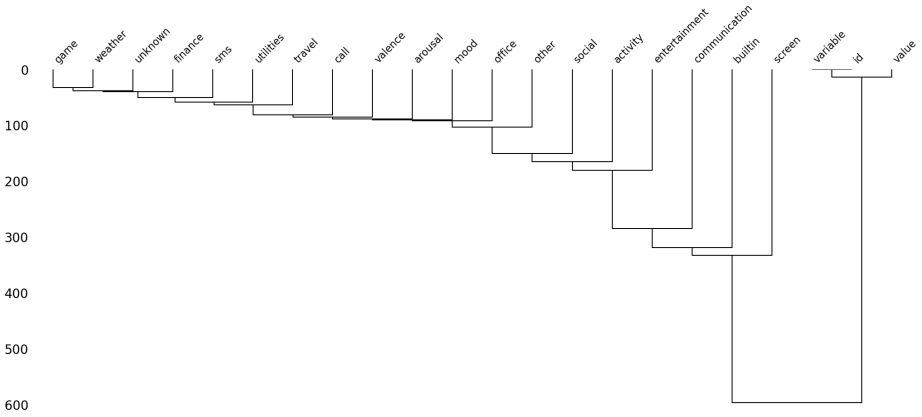


Fig. 7: Missing values

We dropped columns variable and value, since we have already transformed their contents in other columns, which we will use further for feature engineering. For imputing the missing values, we have opted, first for a SimpleImputer in which we impute missing values based on the 'most_frequent' approach, and second, for a time-series interpolation. Moving further, we have opted for the latter approach, as using the most frequent value to impute missing data does not account for time dependencies, while time-series considers the temporal order of data.

2.3 Feature Engineering

The feature engineering process begins by analysing the correlation matrix to identify highly correlated features and merging them to create broader categories reducing redundancy in the dataset.

The categories "call" and "sms" were merged into a new column "communication_activity" and "unknown", "finance", "screen", "game", "utilities", "weather" and "communication" were all merged into "other". Furthermore, the 'Day' column is created to indicate the day number for each user. The core transformation involves calculating a 3-day moving average for each variable in the dataset. This adds new features that capture broader trends in the data while smoothing out short-term fluctuations. Finally, the original variable columns were dropped, leaving the dataset with 13 columns (Fig. 9).

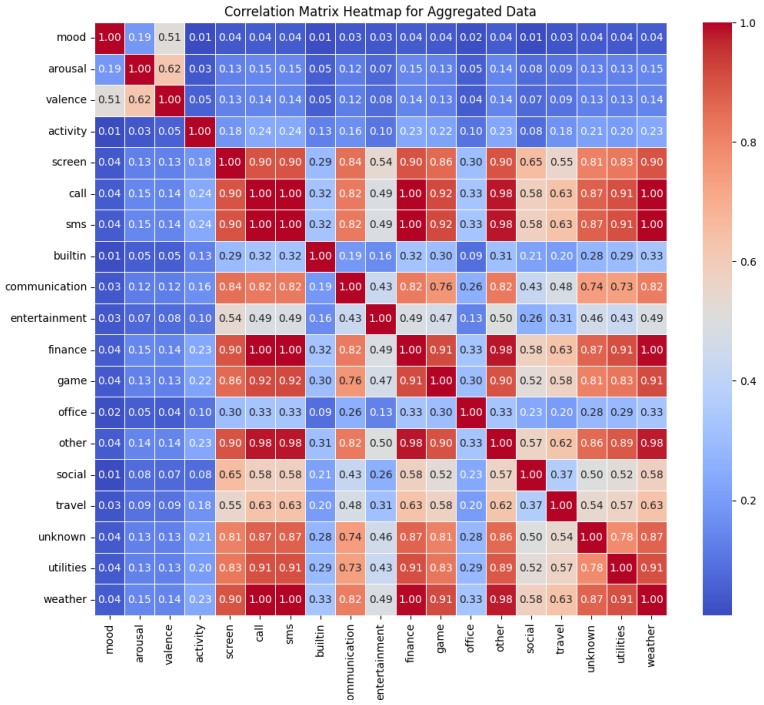


Fig. 8: Correlation of the different variables

id	Day	mood	mood_3day_avg	arousal_3day_avg	valence_3day_avg	activity_3day_avg	builtin_3day_avg	entertainment_3day_avg	office_3day_avg	other_3day_avg	social_3day_avg	travel_3day_avg	communication_activity_3day_avg
AS14.01	1	7.0	0.0	0.0	0.0	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.0
AS14.01	2	7.0	0.0	0.0	0.0	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.0
AS14.01	3	7.0	0.0	0.0	0.0	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.0
AS14.01	4	7.0	7.0	1.0	1.0	0.0	4.008	4.012	4.008	73.024	4.008	4.032	8.0
AS14.01	5	7.0	7.0	1.0	1.0	0.0	5.010	5.015	5.010	91.280	5.010	5.040	10.0

Fig. 9: Dataset after feature engineering

The rationale behind this approach is to simplify the dataset by addressing redundancy, to highlight temporal trends through the 'Day' column and moving averages, and to focus on underlying patterns by smoothing over daily variations. These transformations aim to make the data more suitable for the next mood prediction tasks, such as classification or regression.

3 Classification

3.1 Application of Algorithm

For the first part of this task, our goal is to predict the mood of the next day using the dataset formed after feature engineering. To begin with, we split the data into features and target variable ("mood") and, also, into a training set (80%) and a testing set (20%). Splitting the data allows us to train the model on one portion of the data and evaluate its performance on unseen data, helping prevent overfitting. The first chosen algorithm is Random Forest. Random forests belong to the family of ensemble methods, meaning they combine the predictions of multiple individual models (in this case, decision trees). Each decision tree in the forest is trained on a random subset of the data and a random selection of features. Trees split the data based on the most informative features, creating homogenous groups regarding the target variable. [Breiman(2001)] This randomness in data sampling and feature selection reduces the correlation between individual trees, making the forest more robust and less prone to overfitting. Furthermore, since we know that the performance of machine learning models can be sensitive to hyperparameter choices, we employ GridSearchCV to systematically explore different combinations of hyperparameters (like the number of trees in the forest, tree depth, etc.). Cross-validation within GridSearchCV ensures reliable performance estimation by avoiding overfitting to the training set.

The metrics used to evaluate our classification model are accuracy - 0.92, precision - 0.88, recall - 0.92 and f1 score - 0.89 (last three metrics are weighted). Also, we conduct feature importance analysis to understand which features contribute the most to the model's predictive power. These results suggest that the model performs well in predicting the mood. Moreover, 'valence_3day_avg', 'mood_3day_avg', and 'arousal_3day_avg' seem to be the most influential features, as you can see in Figure 10.

For the implementation of our Recurrent Neural Network, we have opted for an LSTM, because it is particularly well-suited to making predictions based on time series data, such as predicting a user's mood for the next day. LSTMs predict mood patterns and fluctuations based on temporal dependencies, avoiding the vanishing gradient problem. They use gating mechanisms to regulate information flow, deciding which information to keep or discard at each step. LSTMs can handle variable-length input sequences, learn from recent events, and understand real-life cycles. They can handle variable-length input sequences, make informed predictions, and learn from patterns in human moods, making them suitable for datasets involving user behavior over varying lengths. After training our model, we have obtained a MSE of 0.04 as well as a MAE of 0.08, meaning that our model's predictions are quite close to the actual observed values on average, indicating a high level of accuracy in predicting the user's mood for the next day. The low RMSE values on both training and test sets suggest that the model's predictions are generally consistent with the true mood scores, with a small average magnitude of error.

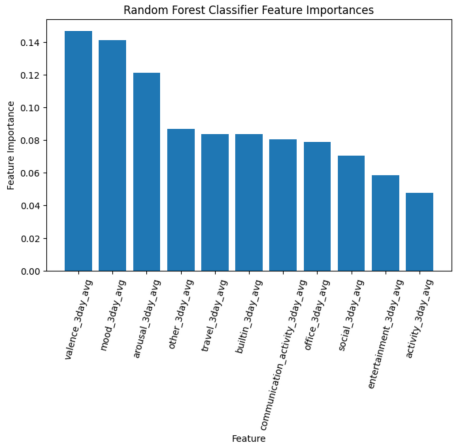


Fig. 10: Random Forest Classifier Feature Importances

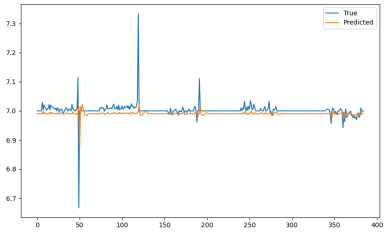


Fig. 11: True vs. Predicted Values - LSTM

3.2 Winning Classification Algorithm

The Otto Group Product Classification Challenge was held in 2015 on the Kaggle platform [Benjamin Bossan(2015)]. Participants were given a dataset containing over 200,000 product descriptions with 93 features. The task was to build a classification model that could accurately predict the correct product category (out of nine possible categories) for each item. The model's performance was evaluated using the multi-class logarithmic loss metric, where lower values signify better performance.

The winners of the Otto Group Product Classification Challenge were Gilberto Titericz Jr. and Stanislav Semenov with a score of 0.38242. Their core technique was ensemble learning with stacking (or meta-learning). This means they trained multiple base models and then used their predictions as features for another model that learned how to best combine them. They specifically used a 3-layer architecture where the first layer consisted of a wide variety of base models like Random Forests, XGBoost and Neural Networks. The second layer used meta-models trained on the first layer's predictions combined with some original

features. Finally, the third layer generated the ultimate prediction by weighted averaging of the second layer's model outputs.

The main idea behind this approach was to intelligently combine the strengths of diverse machine learning models to achieve greater accuracy than any single model could provide. They used a wide range of base models, from traditional algorithms to neural networks, to ensure that the ensemble captured different patterns and perspectives on the data. Predictions from these first-layer models became features for meta-models in a subsequent layer, allowing the system to learn the optimal way to combine these base predictions. The final step involved a weighted averaging scheme to further refine the outcome, giving more importance to the most reliable models within the ensemble.

The winning approach stands out significantly from more standard and the other non-winning methods. In contrast, the strategy adopted by the team ranked 16th, with a score of 0.40327, is simpler, focusing on a single model (XGBoost) and basic feature engineering with a genetic algorithm for ensemble selection. The winning solution uses a sophisticated ensemble design with a tiered architecture and meta-learning, allowing for a more nuanced combination of predictions. Most approaches include a few different model types. This solution employed a remarkably wide range of base models, increasing the potential that very different perspectives on the data were included.

4 Association Rules

Pattern recognition is a useful method for identifying common patterns and correlations in data, providing actionable insights and reducing data dimensionality. It can also serve as a basis for advanced analysis, such as association rule mining. However, it usually may lead to loss of information, arbitrary thresholds, correlation vs. causation, outlier sensitivity, scalability issues, bias towards frequency, contextual relevance, and dynamic data. The choice of thresholds which in our case is of 75%, can potentially lead to misleading patterns. Additionally, the method may not accurately reflect the state of the data at different points in time. Based on our results from applying the above-mentioned algorithm, the data shows a range of metrics, with the first row having all zeros, suggesting a normal state. The second row has a pattern where all metric values are high simultaneously, suggesting a specific event. The third row has a high activity metric, while the fourth row has a high mood metric. The final row indicates high arousal levels on some occasions.

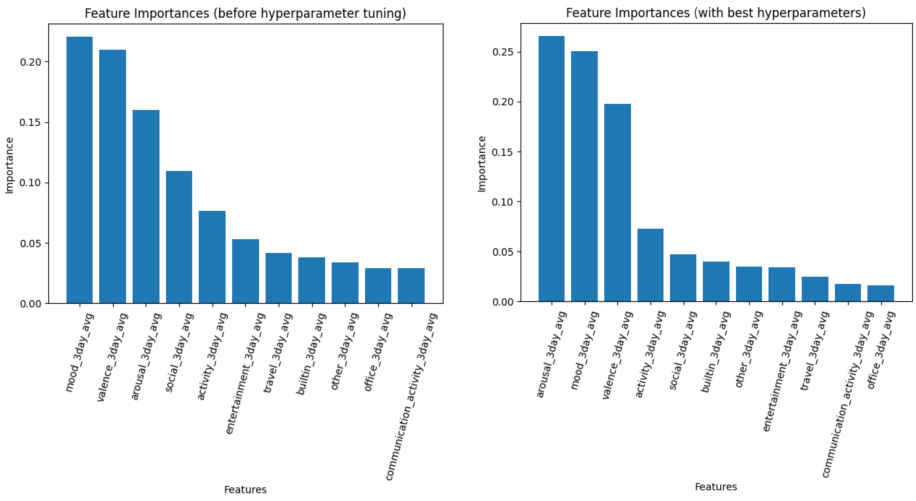
5 Numerical Prediction

For the task of predicting a numerical target variable, we, again, employ a Random Forest algorithm. The core principle of Random Forest for classification and regression is the same - combining decision trees to create a robust ensemble model. The difference lies in the nature of the target variable (discrete

vs continuous) and how individual tree outputs (class probability vs numerical values) are aggregated to arrive at the final prediction.

Once we train the model (similar to the RF classifier in task 2a), it can make predictions on the testing set. We evaluate how well the model has learned to predict the mood using mean squared error (MSE) and mean absolute error (MAE). Also, we analyse which features in the dataset are most influential in the RF model's decision-making (Fig. 12).

Next, we once again implement a hyperparameter optimization process that employs GridSearchCV, which systematically searches for the best combination of hyperparameters for our model. After the grid search is complete, we evaluate the model with the best hyperparameters by calculating MSE and MAE. Additionally, we perform another feature importance analysis using the best hyperparameters. As you can see in Figure 12, now the most important feature is arousal ("arousal_3day_avg").



Before hyperparameter tuning With best hyperparameters

Fig. 12: Random Forest Regressor Feature Importances

As for the evaluation metrics, we can see a slight improvement in model performance after hyperparameter optimization. The mean squared error decreased from 0.0022 to 0.0017, and the mean absolute error decreased from 0.0166 to 0.0142. This improvement suggests that fine-tuning the model's hyperparameters led to a better fit to the data and enhanced predictive accuracy.

For our second approach, we employed a Linear Regression model. Linear Regression is a classic statistical technique used for modelling the relationship between a dependent variable (target) and one or more independent variables (features). It assumes a linear relationship between the features and the target,

with the goal of minimising the difference between observed and predicted values. Unlike more complex models such as Random Forests, Linear Regression does not typically involve hyperparameter tuning. Linear Regression models are inherently simple and do not have a lot of hyperparameters that require optimization. Just like before, we use the training set to fit the Linear Regression model, while the testing set is used to evaluate the model's performance. Then, we calculate MSE and MAE to quantify the discrepancies between the predicted and actual values.

The results of the model evaluation indicate a mean squared error of 0.0018 and a mean absolute error of 0.0158. The relatively low values of both error metrics suggest that the Linear Regression model performs well in predicting mood based on the provided features created in task 1c.

6 Evaluation

6.1 Characteristics of Evaluation Metrics

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

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

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of samples.

MSE and MAE are two common regression evaluation metrics, but they have key differences. MSE squares the differences between predicted and actual values ($(y_i - \hat{y}_i)^2$), making it more sensitive to outliers. In contrast, MAE uses the absolute value of the differences ($|y_i - \hat{y}_i|$), giving less weight to extreme errors. MSE's results are harder to interpret in the problem's original context, while MAE's units match the target variable, aiding comprehension.

The choice between these metrics depends on the data and your goals. If the dataset contains outliers, MAE is often a more robust choice. If predicting large deviations accurately is a priority, then MSE is more appropriate. For scenarios where you need a clear interpretation, MAE is more preferable.

6.2 Example Situation

Implying that both MSE and MAE give identical results, means that $(y_i - \hat{y}_i)^2 = |y_i - \hat{y}_i|$, which results that the only solution for this case is for when $(y_i - \hat{y}_i)^2 = |y_i - \hat{y}_i| = c$ and $c = 1$ or $c = -1$.

In a concrete example, let's assume that we are dealing with temperature forecasting and we want to predict the daily temperatures over the span of one week. As such, in the case where the actual temperatures are $20, 22, 21, 23, 22, 21, 20$ and our predicted values are $21, 23, 22, 24, 23, 22, 21$, our MSE will be: $(1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2)/7$, while our MAE will be: $(|1| + |1| + |1| + |1| + |1| + |1| + |1|)/7$, which in both cases yield 1.

6.3 Impact of Evaluation Metrics

In task 4, we performed regression analysis using two different approaches: Random Forest Regression and Linear Regression. Both models were evaluated using MSE and MAE as evaluation metrics. Overall, both models showed similar behaviour in terms of the impact of MSE and MAE. They both had lower MSE values compared to MAE, suggesting a sensitivity to larger errors and that they might struggle a bit with outliers in the dataset.

References

[Benjamin Bossan(2015)] Wendy Kan Benjamin Bossan, Josef Feigl. 2015. Otto Group Product Classification Challenge. https://kaggle.com/competitions/otto-group-product-classification-challenge

[Breiman(2001)] Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.

[Saeb et al.(2015)] Sohrab Saeb, Mi Zhang, Chris Karr, Stephen M. Schueller, Marya E. Corden, Konrad Kording, and David C. Mohr. 2015. Mobile phone Sensor Correlates of Depressive Symptom Severity in Daily-Life Behavior: An Exploratory study. *JMIR. Journal of Medical Internet Research/Journal of Medical Internet Research* 17, 7 (July 2015), e175. https://doi.org/10.2196/jmir.4273

[Torous et al.(2017)] John Torous, Jorge A. Rodriguez, and Adam C. Powell. 2017. The new digital divide for digital biomarkers. *Digital Biomarkers* 1, 1 (June 2017), 87–91. https://doi.org/10.1159/000477382