Drunk Detector App

Toby Towler – 100395626

George Strugnell – 100344493

Group work 50% each

Report

Aims, Motivation and Background

DrunkDetector is a mobile phone Android[1] application written in Java with built in Android Auto[2] support.

To run the app all the user needs is an Android phone running Android 11 or later with an accelerometer and gyroscope. These requirements are satisfied by most consumer android devices released since 2021. However, if the Android Auto feature became optional, the Android version limitation would be removed, widening the capable devices to any smartphone with a gyroscope and accelerometer.

The aim of this project is to prevent accidents involving people who are drunk. We aim to complete this by alerting the user when they are likely intoxicated as well as a third party for extra safety. This app also has an emphasis on stopping drink driving, since alcohol impairs reaction time, coordination and judgement. It is a huge risk to not only the driver but passengers, other road users and even pedestrians not in a car. Preventing drunk driving can save lives and reduce unnecessary collisions on or off the roads.

According to the UK government, in 2021 "an estimated 6,740 people were killed or injured when at least one driver was over the drink-drive limit"[3] - around 5%. A large number like that could be easily reduced with better reminders to not drink and drive. This app aims to promote more responsible drinking habits and prevent future catastrophes like this.

Alcochange[4] is a similar product that uses a smartphone and breathalyser in an attempt to reduce alcohol consumption. The primary focuses of this product are liver health and decreasing alcohol over consumption. It brands itself as a digital therapy app with behavioral changing techniques. The app uses

Where this product differs from our application is the intended outcome. Alcochange seeks to The Most Versatile Compact Breath Alcohol Testing Device

[1] https://www.android.com/intl/en_uk/
[2] https://www.android.com/intl/en_uk/auto/
[3] https://www.gov.uk/government/statistics/reported-road-casualties-in-great-britain-involving-illegal-alcohol-levels-2021/reported-road-casualties-in-great-britain-involving-illegal-alcohol-levels-2021
[4] https://alcochange.com/

SCRAM Remote Breath Pro[5], is a digital breath alcohol monitoring tool with facial recognition systems and GPS to track alcohol use over time. The companion mobile app provides reminders and a check in service. This service aims to monitor individuals who experience alcohol issues acting as a reminder not to drink.

Similar to DrunkDetector, SCRAM's product tries to deter the user from drinking too much with scheduled reminders to constantly remind the user of their goal to reduce consumption. They both act as reminders to how much alcohol the user has had or how drunk they are.

Sensing

To determine the likelihood of a user being drunk, the system makes use of the built-in gyroscope and accelerometer sensors in an android phone. These give the accelerations in the x, y, and z axes, as well as the roll, yaw, and pitch values.

Some of the key symptoms of being drunk, also known as alcohol intoxication, include a loss of balance, loss of coordination, and staggering (Healthline: Alcohol Intoxication: Acute, Symptoms, Treatments, Signs, and More). These symptoms involve features such as sporadic accelerations, sudden changes in accelerations, sudden large rotations, and a loss of correlation in directional movement, especially when compared to the more predictable motions of a sober person. These features can be directly calculated from accelerometer and gyroscope data, and similarly the lack of these features can also be inferred from the same data. Thus, these two sensors are both necessary and sufficient to infer drunkenness in a user.

Almost all android devices come with an accelerometer and gyroscope, and the android operating system provides access to the sensors on the phone with very few restrictions (Sensors Overview | Sensors and location | Android Developers). In addition, 70.93% of smartphone users worldwide use an android device (Android Phone Statistics By Apps, Sales and Market Share). Thus, by making use of android as our system platform, we can reach a very large target audience with no additional expense required.

In our app, the accelerometer and gyroscope data are collected at a rate of 32Hz by a foreground service, and placed in two circular buffers, one for accelerometer readings and one for gyroscope readings. By handling the data collection with a foreground service, it is ensured that the operating system will not halt the collection of data, and with each buffer

[5] https://www.scramsystems.com/monitoring/scram-remote-breath-pro/

having a length of 128, this ensures the latest 4 seconds of movement data are always accessible.

The accelerometer data can be used to infer walking patterns, whether the user is walking with a consistent gait, or if they are walking unsteadily, with irregular steps. Large spikes can indicate lurching or falling, however other movements associated with drunkenness, like swaying or wobbling are harder for accelerometers to detect, as these consist of more rotational elements.

This is the advantage of the gyroscope, which can measure these patterns in movement that the accelerometer cannot. However, in turn, the gyroscope is not, in isolation, able to effectively detect walking patterns in the user in the way an accelerometer can. Thus, to catch the entire array of drunk behaviors, both sensors are required.

Another possible sensor which had been considered was a GPS sensor, to determine if the user was spending time in a bar. This could be used to calculate an increased likelihood of the user being drunk. However, this would require access to the Google Maps Platform, which is only free for limited use, and requires payment beyond that. It is also worth considering that a person can get drunk outside a bar, or can be sober despite going to a bar, so this sensor method would never be sufficient by itself

Data Collection

Data for this project was collected by both team members, through a series of mock scenarios, in which various activities were carried out, either approximating the movement of a drunk person, or the movement of a sober person. Collecting data while one of us was drunk had been considered, however we considered this to be not necessary, as we could make ourselves dizzy on purpose, which would result in similar symptoms of lack of coordination and stumbling. There was also the concern that the team member collecting the drunk data might have incorrectly recorded the data because of being drunk, which would have been counterproductive to our aims of collecting reliable data. The data was collected using the app from lab A3, where the “Walking-flat-surface" label was repurposed for “Not Drunk”, and the “Running” label was repurposed for “Drunk”. This was done to save time on creating an app that could fulfil the same purpose. The app from A3 recorded timestamped accelerometer and gyroscope data over a controlled period and stored the results in labelled files. This is all that was required for the purpose of our project, so we believe this to be a justified approach.

The final dataset consists of a series of “.txt” files containing timestamped accelerometer or gyroscope data. The files are labelled with the “activity”, either “Walking-flat-surface"(Not Drunk), or “Running” (Drunk), as well as the type of data (gyroscope or

accelerometer). The data for “Not Drunk” includes data collected while doing a wide variety of tasks, such as climbing stairs, walking, sitting, standing, kneeling, walking quickly, turning, bending over, and fidgeting. This variety of activities was used to ensure the model trained on this data did not inadvertently deem any ordinary physical activities as “drunkenness”. The data for “Drunk” represents activities such as staggering, stumbling, unsteady foot placement, swaying in place, suddenly stopping or starting, nearly falling, and swaying while walking. This was to ensure the model could pick up on the variety of unusual movement patterns that can arise from being drunk.

Data Processing

The raw data collected during the data collection stage is first pre-processed before extracting features. This begins by reading the data from the txt files, and processed in a manner similar to that in lab A3. For each session, the accel and gyro files are read from, and if either of these are missing, the session is skipped. The first and last 3 seconds of data are discarded to account for the time spent turning on/off data collection, and moving the phone to/from the pocket. Timestamps are then selected at a constant step between the new start and end points. These are used to interpolate the accelerometer and gyroscope data, to ensure that between the two data sources, the timestamps are consistent, and that the length of these data sources are the same. The length of the accelerometer data is then truncated to a multiple of 128, and the length of the gyroscope data is truncated to the same length. Finally, for each session, 3 arrays are created: an array for each of the sensor data values at each timestamp, an array for each of the labels for those data points, and an array for the timestamps of each of those data points.

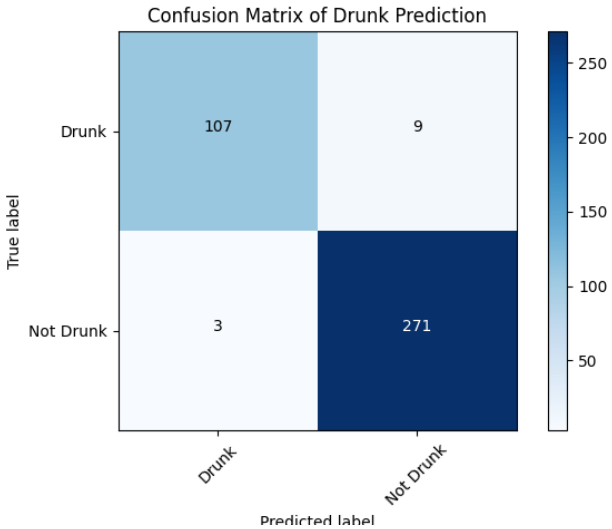
By running this pre-processing step, all incomplete data is removed, data is ensured to be the correct size for extracting the features, and the data values for each datapoint are consistent in time.

The next step is the feature extraction. Firstly, the total magnitude of accelerometer data and total magnitude of gyroscope data is calculated for each data point in the dataset, giving two new “raw features” to extract summative features from. These summative features are calculated using a sliding window of size 128. In each window, each of the “raw features” are used to calculate 14 summative features, and in addition, the correlations between the accelerometer readings (xy, yz, zx), and the gyro readings (roll/yaw, yaw/pitch, roll/pitch) are calculated. This yields a new dataset where each datapoint has 118 features. The features were selected based on the common symptoms of drunkenness. The features selected were mean, variance, maximum, minimum, kurtosis, skew, sum of squares, zero crossings, peak amplitude, power, mean amplitude, dominant frequency, entropy, and mean rate of change. These features were picked to

infer behaviors such as sudden stopping and starting, lack of coordination of movement, stumbling, staggering, unstable walking, leaning, lack of balance, inconsistent walking pace, and swaying. Statistical features such as kurtosis or variance can demonstrate whether a person's movement are varied and not stable, and frequency features such as entropy or peak amplitude can show if a person's motion is more periodic or chaotic. Features such as maximum or mean rate of change can show the presence of large movements, indicative of behaviors such as falling or stumbling, while mean or zero crossings can show general trends in a person's movement, such as swaying from side to side.

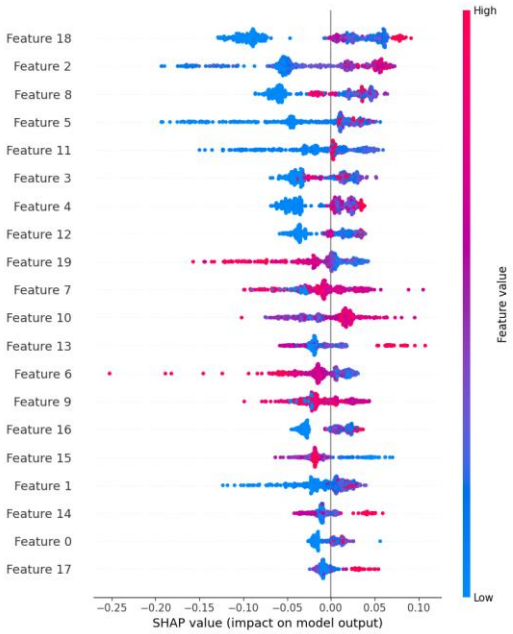
Each of the new data points has a label, either "Drunk" or "Not Drunk", and the data was split into training and test sets. These are used to train a Random Forest Classifier, and from this classifier, a reduced set of the most significant features were chosen, and a new classifier was trained using these features. The parameters of this classifier were chosen with a grid search using 5-fold cross validation, varying the number of estimators, max depth, minimum samples required to split a node, minimum samples required at a leaf node, and max features for a split. The search was scored using the accuracy metric, and the highest performing estimator was selected. The best model had no max depth, used square root to determine max features for a split, had minimum sample of 1 required at leaf nodes, 2 samples required to split at minimum, and 100 estimators. This model was then scored on the testing data set. The number of features selected for this reduced model was selected to be 20, as this was the number of features that resulted in an accuracy on the test data greater than 95%, which was our goal to avoid overfitting. From the results of this training, a confusion matrix and SHAP graph were generated to demonstrate the effectiveness of the model, and the effect of each feature.

This confusion matrix shows that the produced model performs very well on the test data. The classification report highlights strong values for precision, recall, f1 score, and accuracy. The cross-validation score for the model was 0.956, and the accuracy on the test data was 0.969 to 3 significant figures.



This diagram shows the SHAP values of each data point in the test set for each feature. For the top 8 features, high feature

values contribute to a ”Drunk” classification, and low feature values contribute against. These top feature values correspond to: mean rate of change of yaw, likely indicating an unsteady walking direction (reduced ability to walk in a straight line), skew of total acceleration, likely indicating a propensity towards exaggerated motions, mean jerk of y acceleration, possible indicating heavy steps (i.e. from a loss of balance), peak of z acceleration, most likely indicating a large stumble to the side, mean jerk of total acceleration, indicating the overall propensity towards sudden stops or starts in motion, mean jerk of x acceleration, indicating sudden stops or starts in the forwards direction, likely as a result of stumbling, and the subsequent recovery from the stumble, and mean jerk of z acceleration, which could be representative of the sudden sideways stumbling that occurs when drunk. Notable also is feature 19, the correlation between roll and pitch. In this case, lower values actually indicate drunkenness, as lower correlation between roll and pitch indicate less coordination in the person’s movements. Thus, it can be seen that the top features all directly correspond to known symptoms of drunkenness, and their use in prediction of drunkenness is sensible.



The remaining top features are, in order of appearance in the diagram: power of y acceleration, skew of z acceleration, mean of total angular velocity, skew of y acceleration, mean of z acceleration, mean rate of change of roll, minimum roll, kurtosis of total acceleration, peak of total angular velocity, maximum acceleration, mean amplitude of yaw.

System Output and Feedback

Data is presented to the user in numerous different ways meaning the user can always be aware of their current state.

The dashboard section of the app constantly outputs the likelihood of intoxication regardless of the predicted state of the user. This data is shown as a percentage and constantly updated in real time. Once the percentage is above the required threshold, we chose 60% certainty of drunkenness over the last 4 seconds, the user will receive a

notification alerting them they are likely drunk, this should serve as a reminder to be careful with their actions going forward and keep themselves safe.

If this does not work an SMS text message is sent to an emergency contact chosen by the user, for instance a friend or family member. The message alerts the recipient to the user's drunken state, asking for assistance in looking after the user and keeping them out of trouble. The emergency contact's phone number is inputted by the user to the app and can be changed at any moment with instant validation, checking the number is legitimate and UK registered, ensuring information is accurate if the desired caretaker changes.

Should the user attempt to get in a car and drive while drunk, the app automatically connects to their vehicles Android Auto service and displays a message alerting them that they are not safe to be behind the wheel and should instead find alternative transport. If the app does not detect drunkenness, a safe screen is shown telling the driver to drive safely. This is the only page of the app that has 2 "phases" rather than just changing the text as in the app itself. The state of the display is decided from the data before connection is made, as the user sitting in the car is likely to skew the data and give less accurate reading than their walk to the driver's seat. This means that by sitting still for a while once in the car, the app cannot be tricked into false sobriety, causing the "good message" to show.

Due to Google's security restrictions, it is not possible to test apps on a physical Android Auto head unit without official validation or a developer account that comes with a pay wall. However, apps can still be tested on the Desktop Head Unit[6] while functioning identically to if tested on the real thing. This method is how testing was carried out for this product.

Since drunk driving is a big issue in today's society, this feature aims to deter the user from driving at all, serving as a second reminder to what they're doing is both dangerous and illegal. In future implementations, this idea could be extended to connect to the vehicle itself and prevent the car from starting, completely taking away the ability to drunk drive, keeping the roads much safer.

This multitude of alerting methods should suffice to keep the user and their piers safe in a variety of situations.

For images of all outputs mentioned above, please see the screenshots section of the appendix

[6] https://developer.android.com/training/cars/testing/dhu

Conclusion

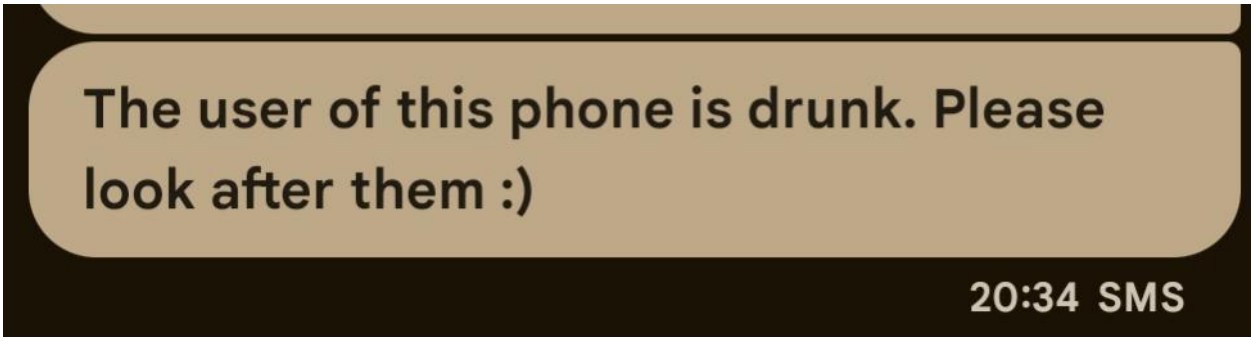
To conclude, DrunkDetector is an all-in-one application that serves to track users' movement through an array of sensors and directional movements, determines drunkenness through a custom engineered machine learning model and finally takes the necessary precautions to ensure the user's safety in several different ways.

DrunkDetector should act as a personal reminder to anyone who uses it that drinking too much can negatively alter their awareness, reaction time and perception of their surroundings, keeping in their mind they need to take extra care with their decisions and actions to keep themselves and those around safe.

The simple requirements of the core functionality of this app can have vast applications. Needing only a gyroscope and accelerometer to track movement, simpler versions could be made to run on much lower spec devices than a modern smartphone. The backbone logic could be applied to much simpler or even embedded systems e.g. on a smart watch or standalone movement tracking device. Whichever way it is, if development was continued, this product could be made accessible to a large audience with varying levels of technical literacy and wealth if pushed to run on cheap, simple devices.

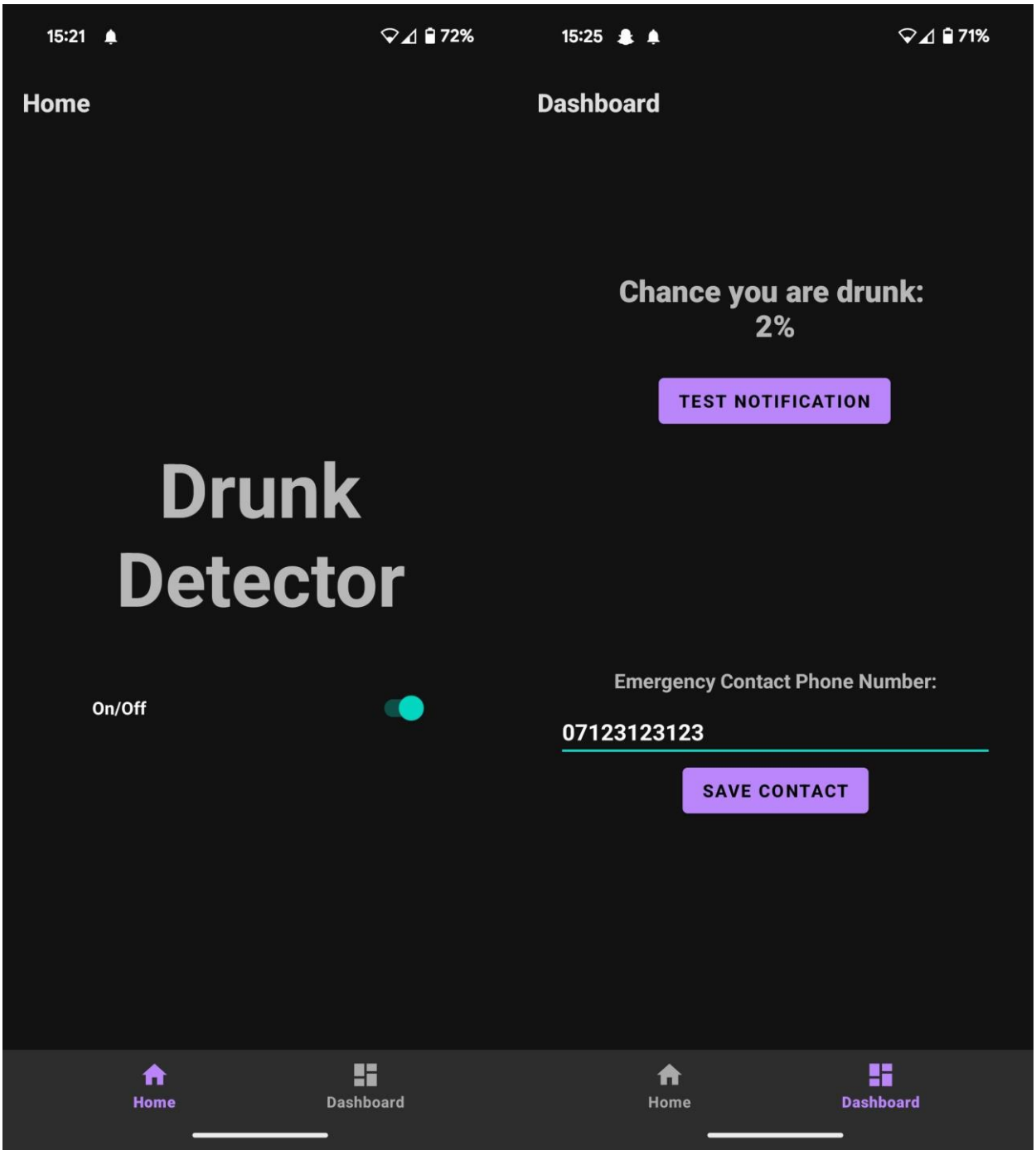
Appendix

UI Screenshots

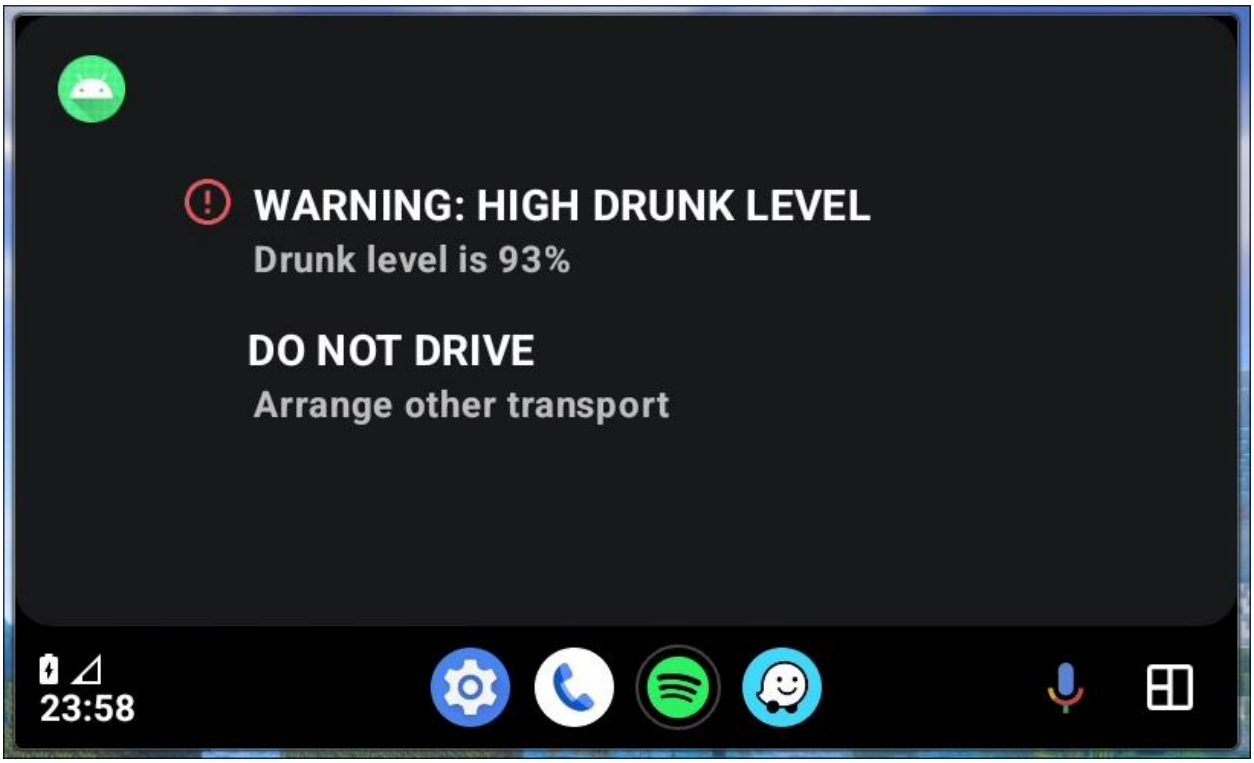


SMS message sent to emergency contact

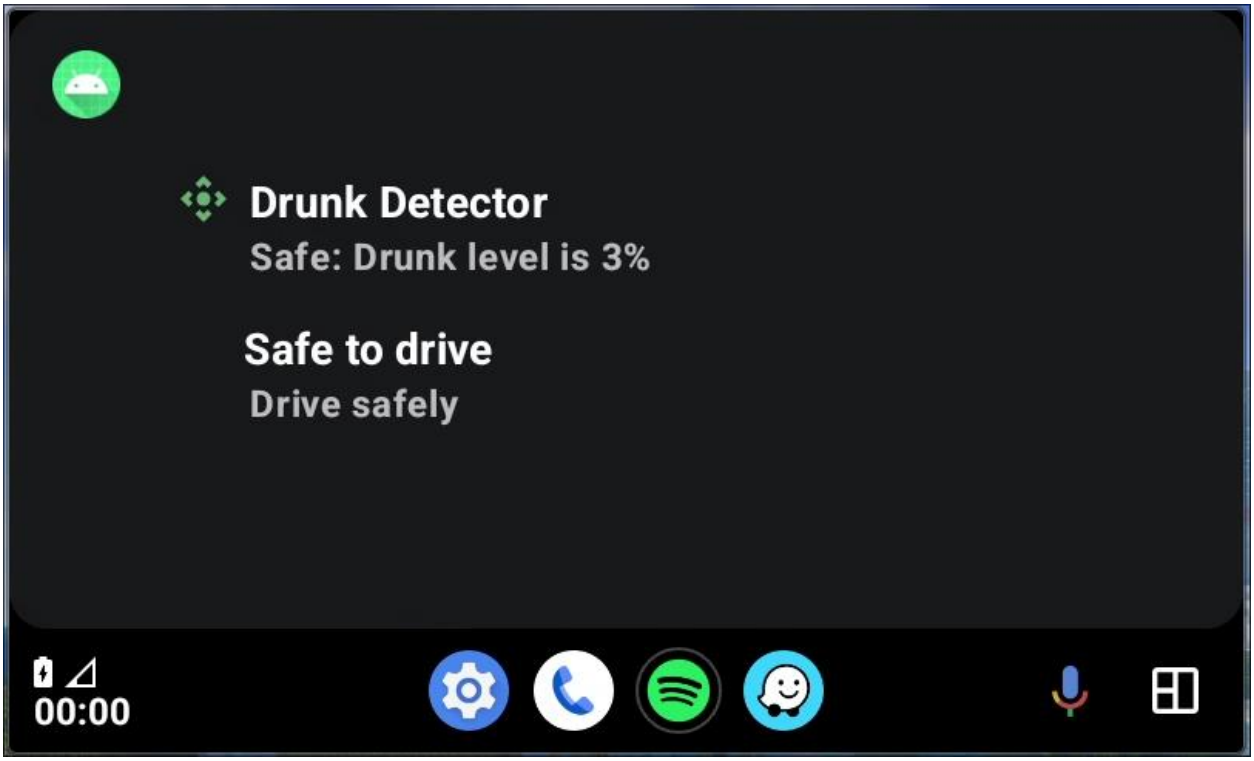
Nofification alerting the user to 90% drunkenness


15:21
72%
Home
Drunk
Detector
On/Off
Home
Dashboard
15:25
71%
Dashboard
Chance you are drunk:
2%
TEST NOTIFICATION
Emergency Contact Phone Number:
07123123123
SAVE CONTACT
Home
Dashboard

UI of the app



Android Auto drunk alert screen from Desktop Head Unit emulator



Android Auto safe screen from Desktop Head Unit emulator

	Precision	Recall	F1-Score	Support
Drunk	0.97	0.92	0.95	116
Not Drunk	0.97	0.99	0.98	274
Accuracy			0.97	390
Macro Avg	0.97	0.96	0.96	390
Weighted Avg	0.97	0.97	0.97	390

Classification report for the Random Forest Classifier model

References

1. https://www.android.com/intl/en_uk/
2. https://www.android.com/intl/en_uk/auto/
3. https://www.gov.uk/government/statistics/reported-road-casualties-in-great-britain-involving-illegal-alcohol-levels-2021/reported-road-casualties-in-great-britain-involving-illegal-alcohol-levels-2021
4. https://alcochange.com/
5. https://www.scramsystems.com/monitoring/scram-remote-breath-pro/
6. Alcohol Intoxication: Acute, Symptoms, Treatments, Signs, and More).
7. Sensors Overview | Sensors and location | Android Developers).
8. https://www.sci-tech-today.com/stats/android-phone-statistics/