ML Project Presentation

Emotion Recognition

Νικόλας Χελιώτης/cheliotisnick@gmail.com

Technical Details

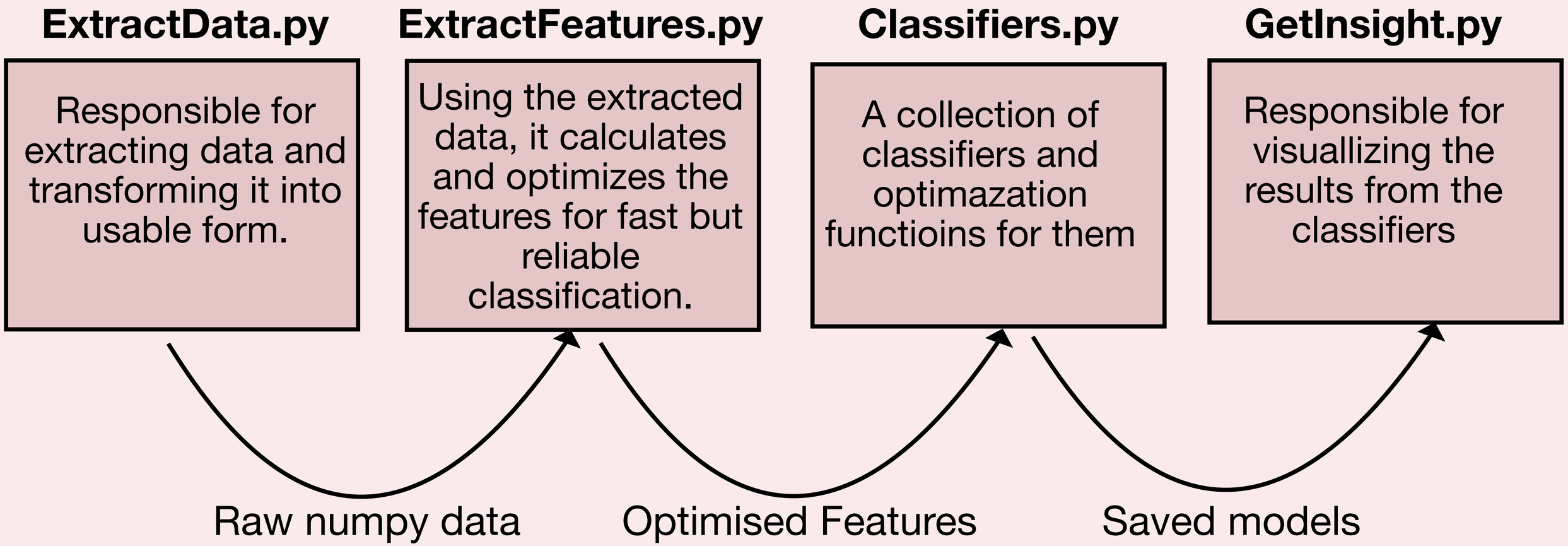
Technical Details

Tools used

- **Python version**: 3.11
- **Main libraries**: Sklearn, Numpy, Mathplotlib
- **Dataset**: FER2013
- **IDLE**: Pycharm

Technical Details

Code structure



Dataset/Features

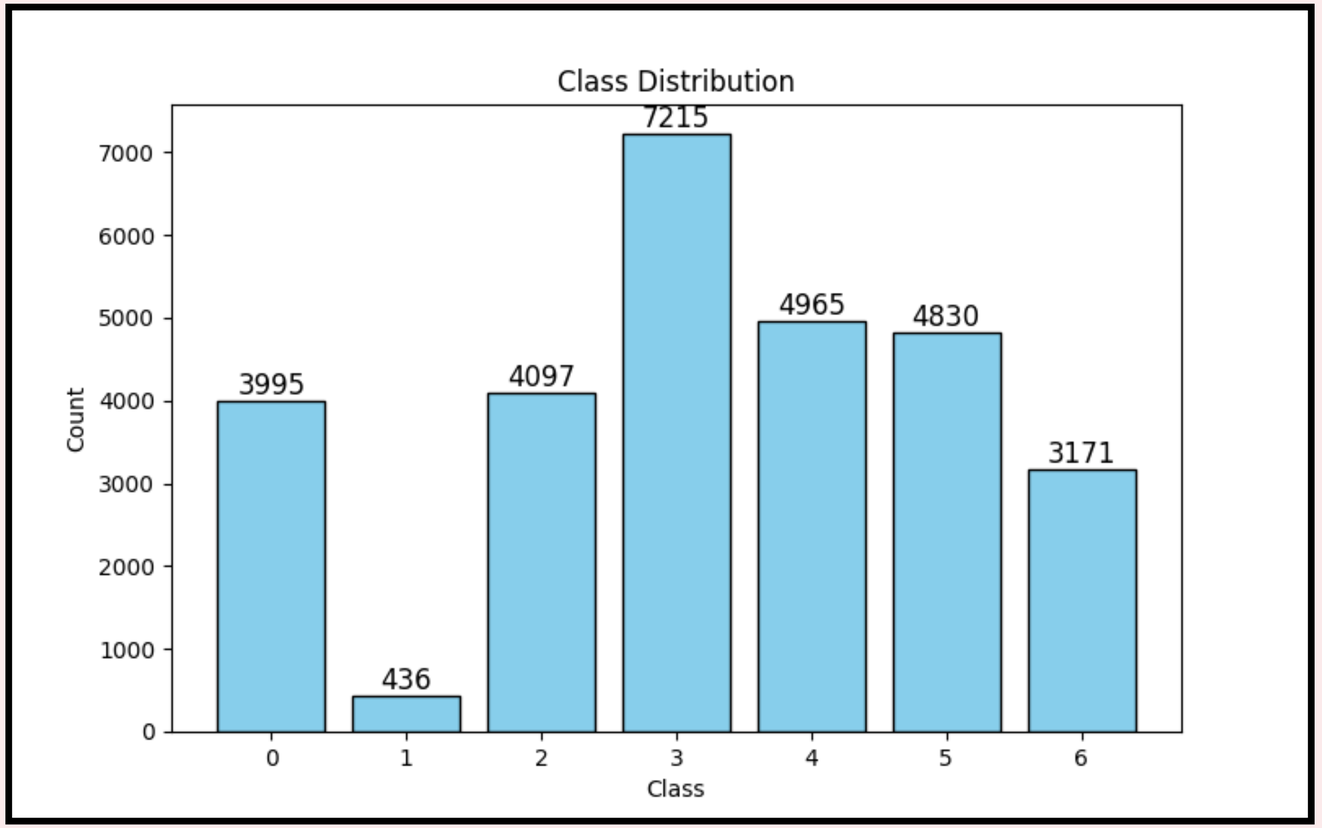
Dataset/Features

Dataset

The dataset used is the FER2013.Its collection of images depicting emotions (**Angry**, **Disgust**, **Fear**, **Happy**, **Neutral**, **Sad**, **Surpise**).The task is to classify an image to the correct emotion.**Sadly**, no features were given so we had to extract them ourselves!

Dataset/Features

Dataset class distribution

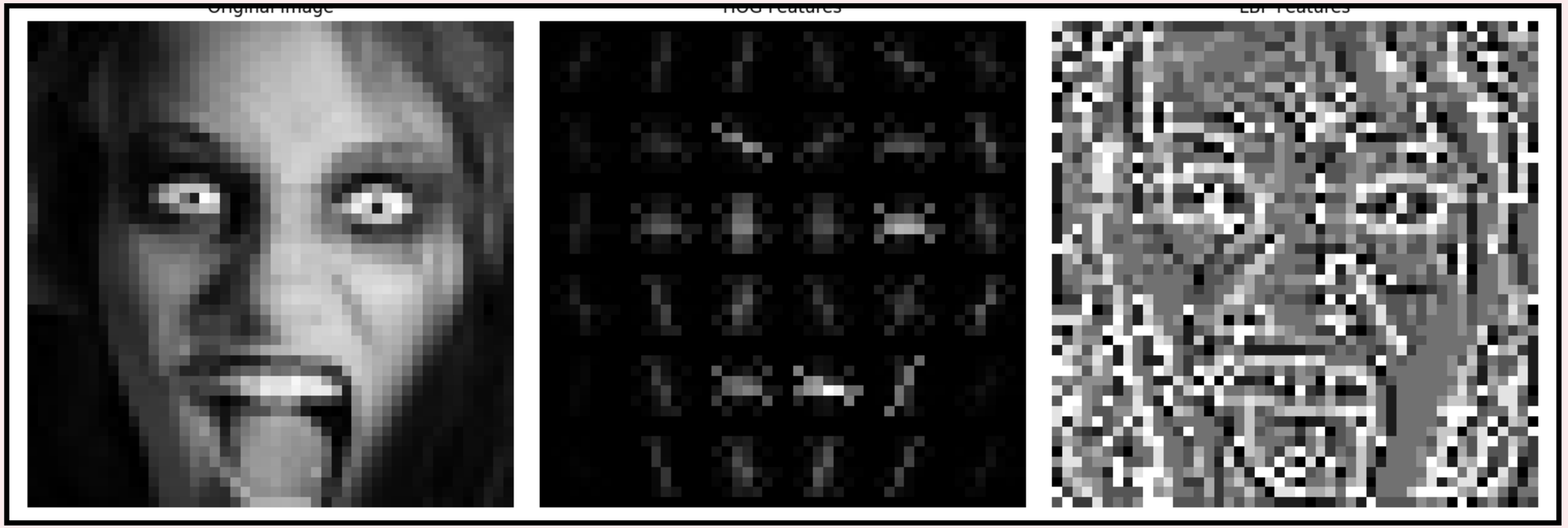


Its evident that class 3 (**Happy**) dominates the other classes.Class 1 (**Disgust**) is under-sampled, which I **fear** is gonna be a problem later on.

Dataset/Features

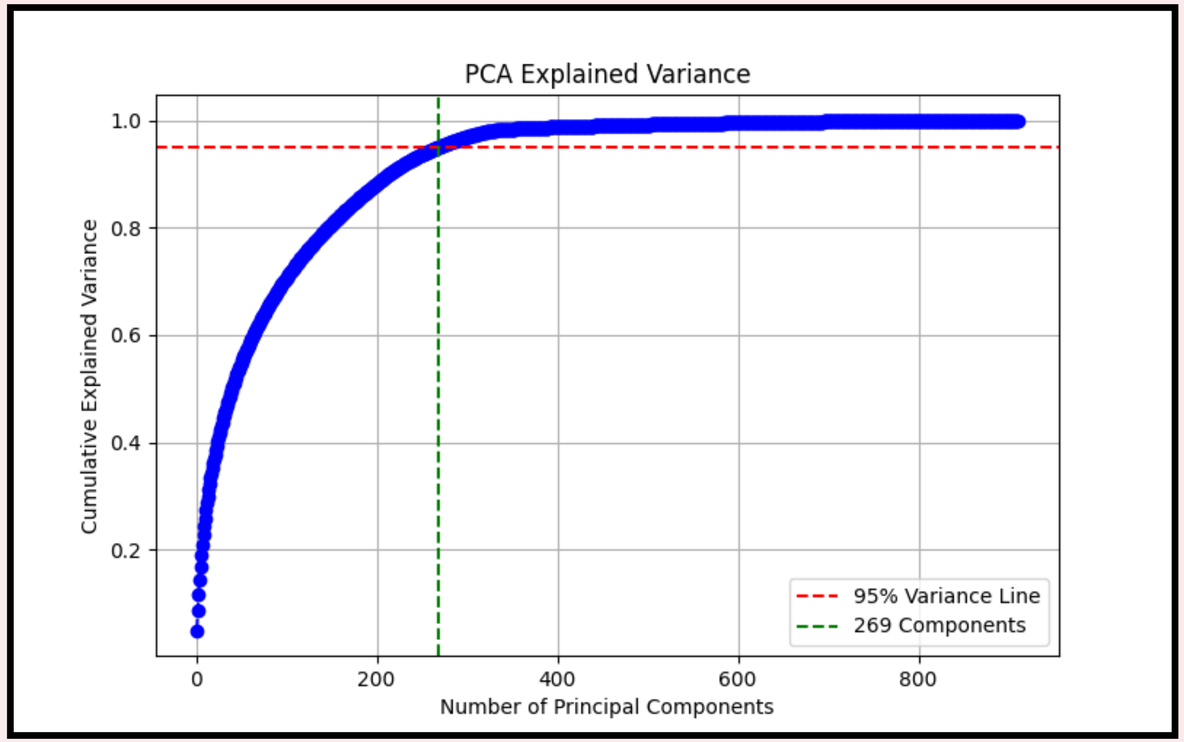
Features

HOG and **LBP** features were chosen.Why?Well, the main idea was that HOG catches shapes - like angled eyebrows when **angry**.LBP was used to detect texture patterns, like wrinkles around the eyes when smiling :)



Dataset/Features

Features



We started with 891 features…which are a lot.Since we didn't want to **scare** our SVM/ RandomForest we applied PCA keeping 95% of the variance.In the end we shrunk it down to 269 features.

Experimenting with classifiers

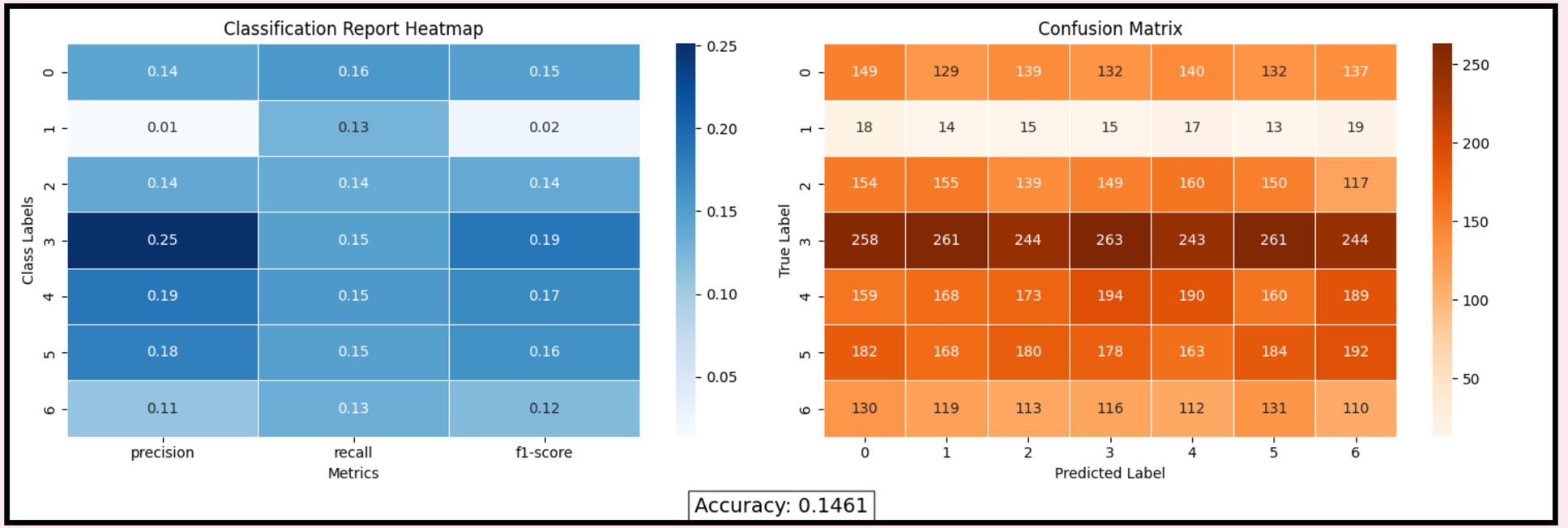
Experimenting with classifiers

Dummy classifier

We needed a **baseline** so we started with the dummy classifier.As expected it delivered a whooping 0.14% **accuracy** with the highest **precision score** (0.25%) belonging to the class with the most samples (class 3).

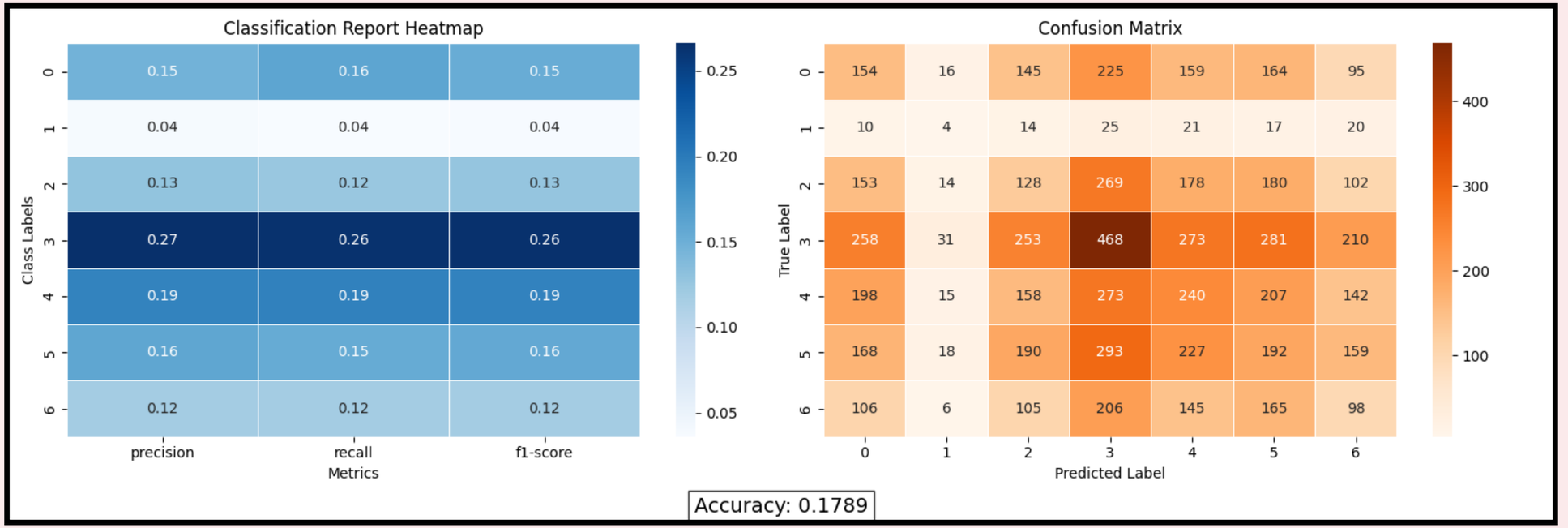
We also tested the **stratified strategy** which respects the class distribution instead of just guessing blindly.**Accuracy** increased to 17%.Progress is progress!

Dummy classifier



Accuracy: 0.1461

Dummy classifier with stratifed strategy



Accuracy: 0.1789

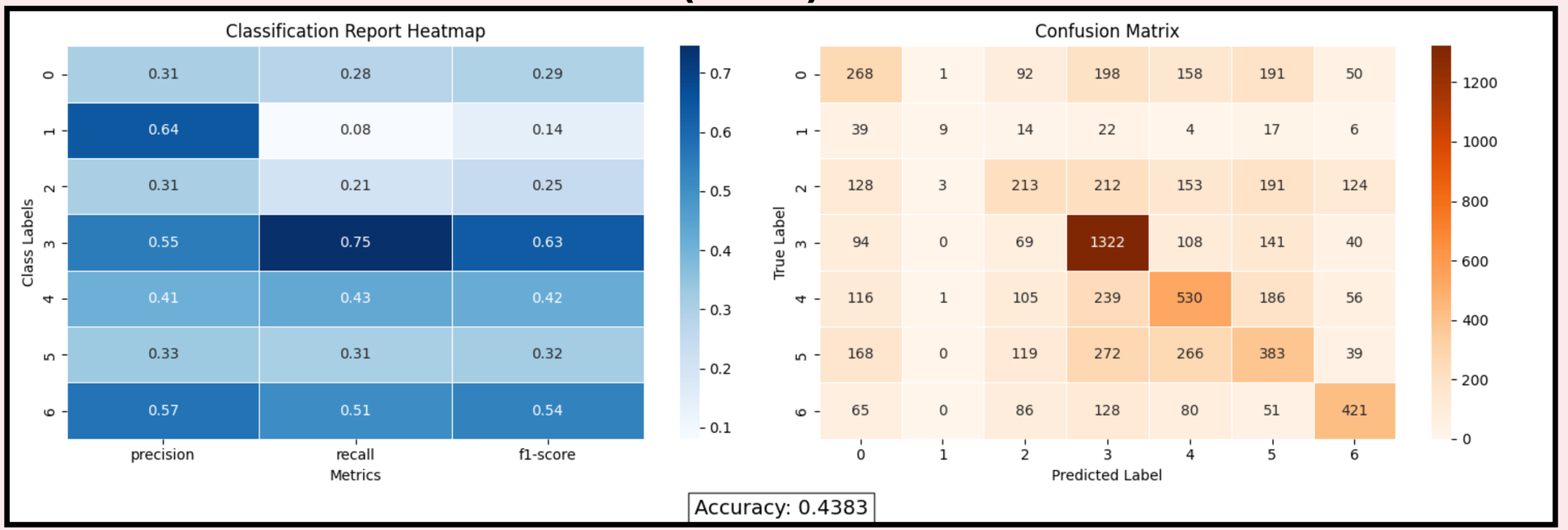
Experimenting with classifiers

Svm (linear kernel)

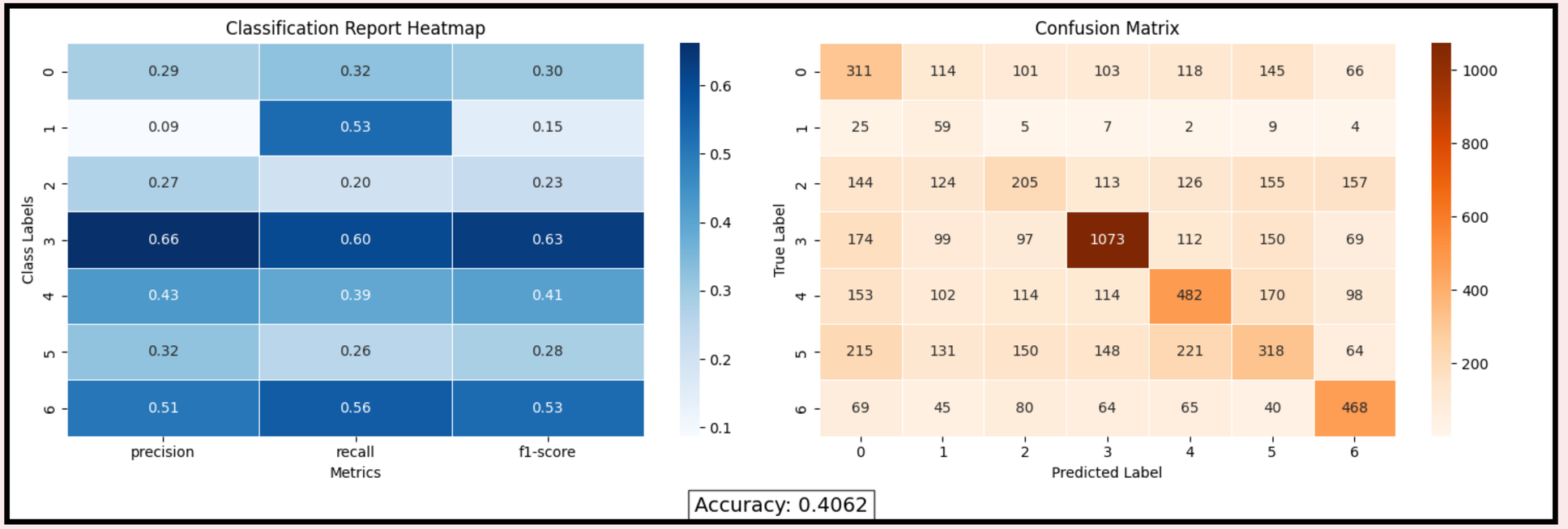
Next, we tested a **SVM** linear classifier.The results were **hopeful**.The model achieved a 43% **accuracy** which means the data actually carries useful patterns ,it's not just random noise. This tells us that the model can differentiate between emotions.However class 1 had a **recall** of 8% which means it barely ever gets predicted **:(**

To solve this problem, we added **weights** to the classes.Class 1's **recall** skyrocketed to 51% but overall **accuracy** dropped to 40% as well as Class1's **precision** which took a massive drop from 64% to 8%.But this was to be expected.

Svm (linear)



Svm (linear, weighted)

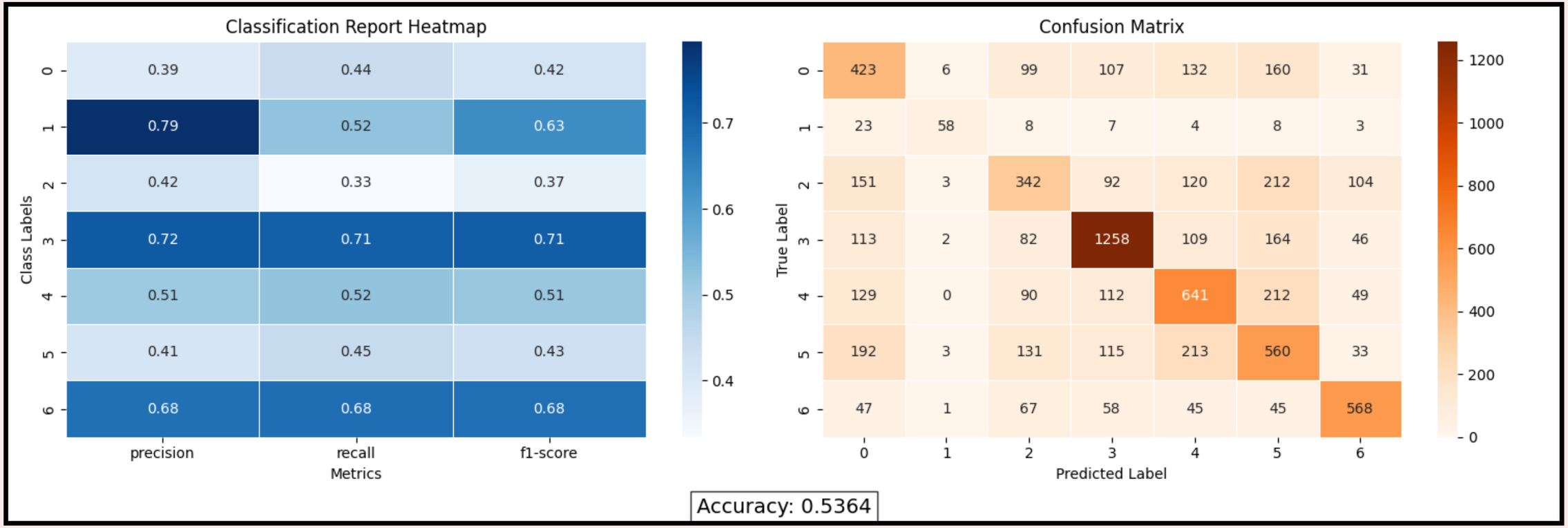


Experimenting with classifiers

Svm (rbf)

Our next experiment involved using the **RBF kernel**, which achieved an accuracy of **53%**! This indicates that the data is not easily linearly separable. All the metrics improved, but most notably, **Class 1's precision increased to 79%!**

Svm (rbf)


Classification Report Heatmap
Class Labels
0
1
2
3
4
5
6
0.39
0.44
0.42
0.79
0.52
0.63
0.42
0.33
0.37
0.72
0.71
0.71
0.51
0.52
0.51
0.41
0.45
0.43
0.68
0.68
0.68
precision
recall
f1-score
Metrics
0.7
0.6
0.5
0.4
Confusion Matrix
True Label
423 6 99 107 132 160 31
23 58 8 7 4 8 3
151 3 342 92 120 212 104
113 2 82 1258 109 164 46
129 0 90 112 641 212 49
192 3 131 115 213 560 33
47 1 67 58 45 45 568
0 1 2 3 4 5 6
Predicted Label
1200
1000
800
600
400
200
0
Accuracy: 0.5364

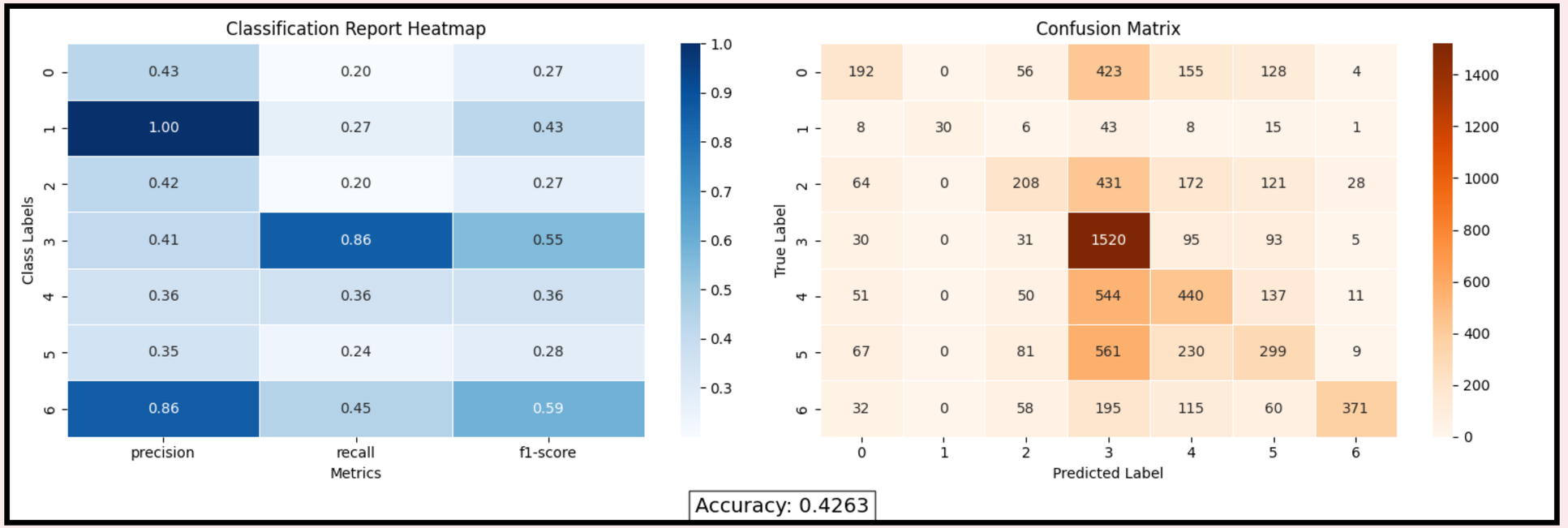
Experimenting with classifiers

Random Forest

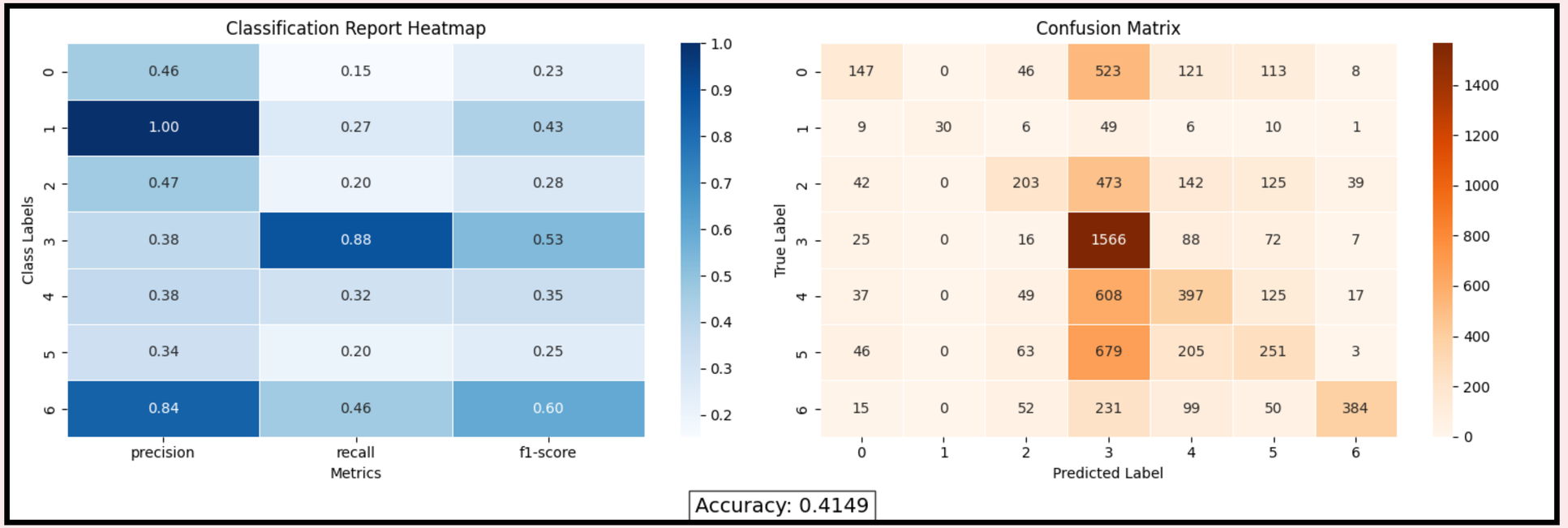
Our next classifier was Random Forest, which achieved an **accuracy** of 42%. Most notably, Class 1's **precision** was 100%, but its **recall** was only 27%. Additionally, Class 3's **recall** increased to 86%, compared to SVM's 71%.

These results motivated us to try Random Forest with class weights, but unfortunately, the results were **disappointing**—there was no significant improvement in any metric.

Random Forest



Random Forest (weighted)



Experimenting with classifiers

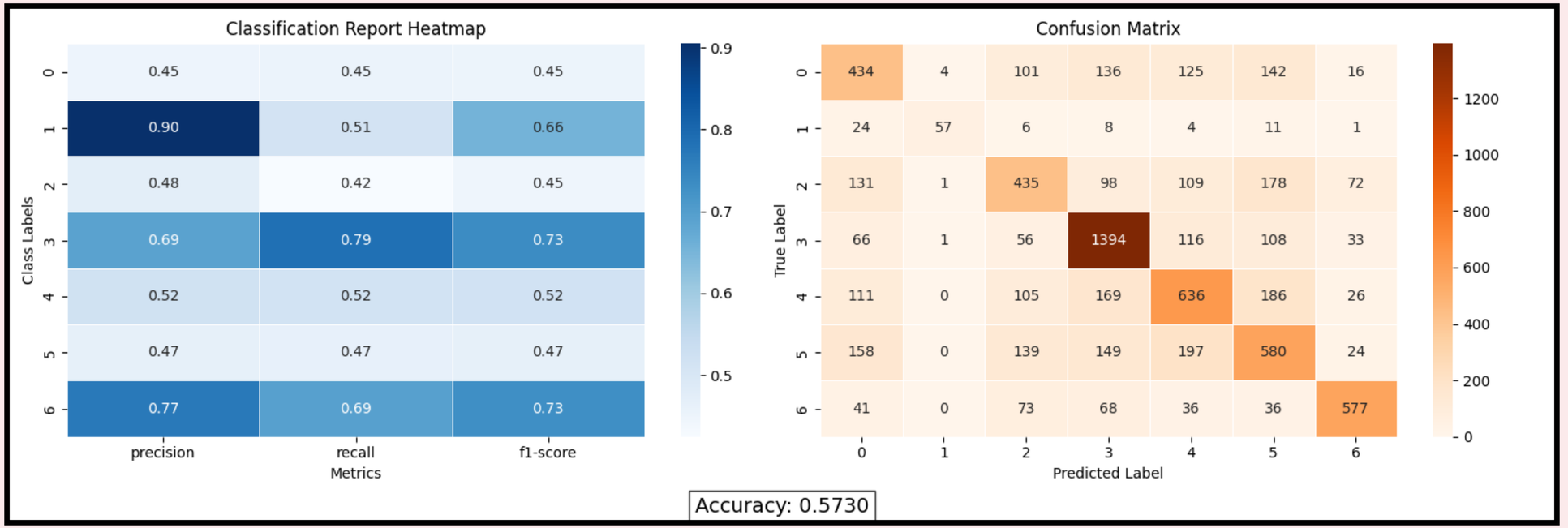
Hyperparameter tuning

The final test aimed to push the limits of emotion recognition using traditional ML methods. For this, we fine-tuned SVM and Random Forest to find their optimal hyperparameters.

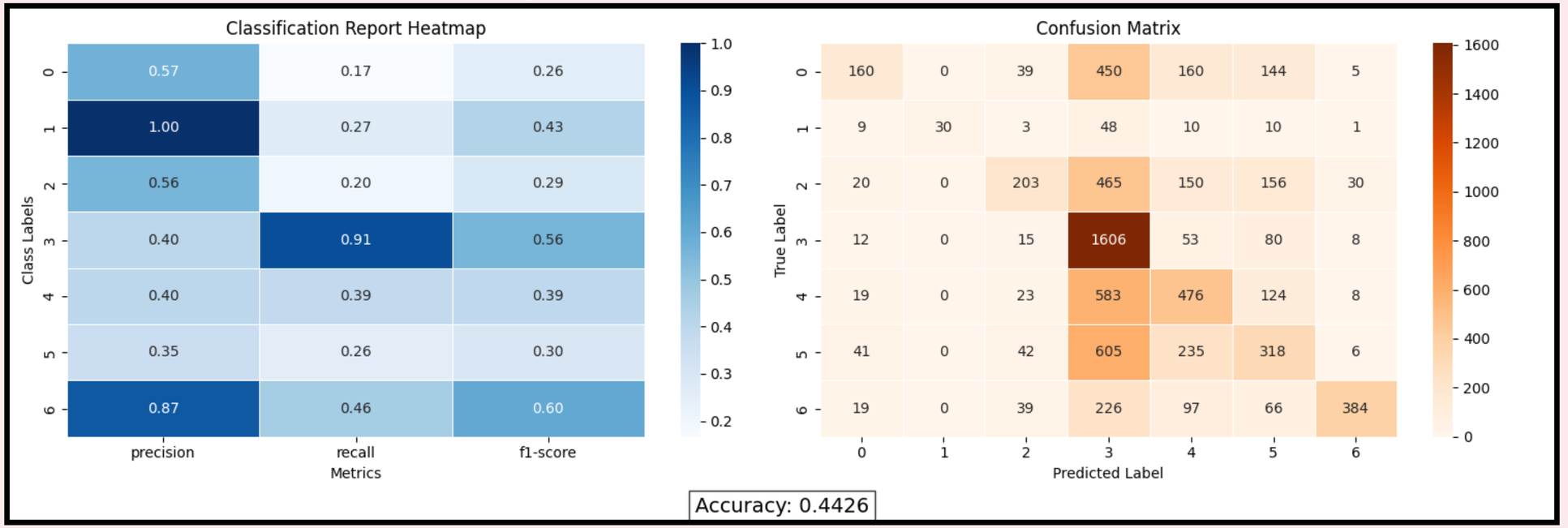
The optimal SVM model achieved 57% **accuracy**, showing an overall improvement across all metrics.

On the other hand, the optimal Random Forest model reached only 44% **accuracy**, which was **disappointing**. There was no significant improvement in the other metrics, making it clear that Random Forest struggles with this task.

Optimized Svm



Optimized Random Forest



To be answered

To be answered

1. **Does age introduce noise?**(older people have wrinkles regardless of whether they are smilling or no)

2. **Does ethnicity impact model classification?** (Facial structure variations across different ethnicities)

3. **Does emotion intensity affect recognition?** (subtle smile vs big smile)

4. **Does gender affect accuracy?** (Maybe the dataset contains more smiling women, leading the algorithm to learn a biased pattern: *“If woman → predict smile.”*)

Comparison and Conclusion

Comparison and Conclusion

Computer vision, and especially emotion recognition, is a difficult task. Our results aren't particularly impressive, with the best model achieving **57% accuracy**.However, comparing with other studies:

1. To our best knowledge, the benchmark (without extra training data) for the FER2013 dataset is **73.28%** (using VGGNet architecture) by Khaireddin & Chen (2021).

2. Zhang (2015) reached **75.1% accuracy** by integrating auxiliary data and advanced feature extraction.

3. For the FER2013 dataset, an **accuracy** of **66.3%** is considered very acceptable for CNN methods (Hanya 2024).

Overall Happiness meter

Questions?

Thank!