Shoot for the Stats, AIm for the Moon:

Developing a pixel AImbot

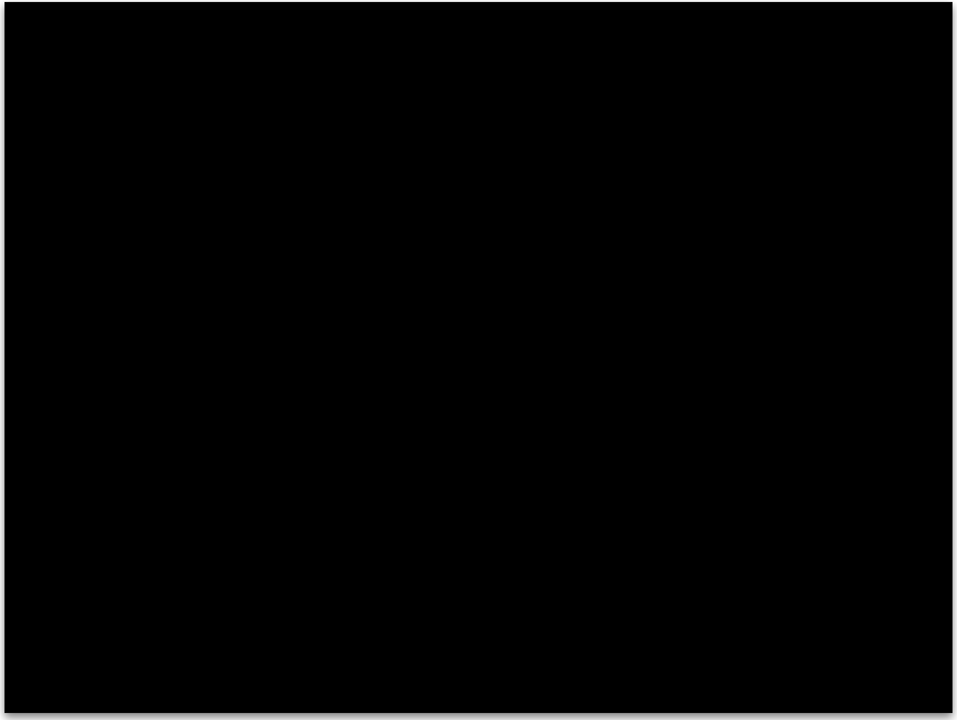
Alessio Barboni - Francesco Redaelli

Project aim

Developing an *AI agent* able to
play a *FPS Aim Trainer game*

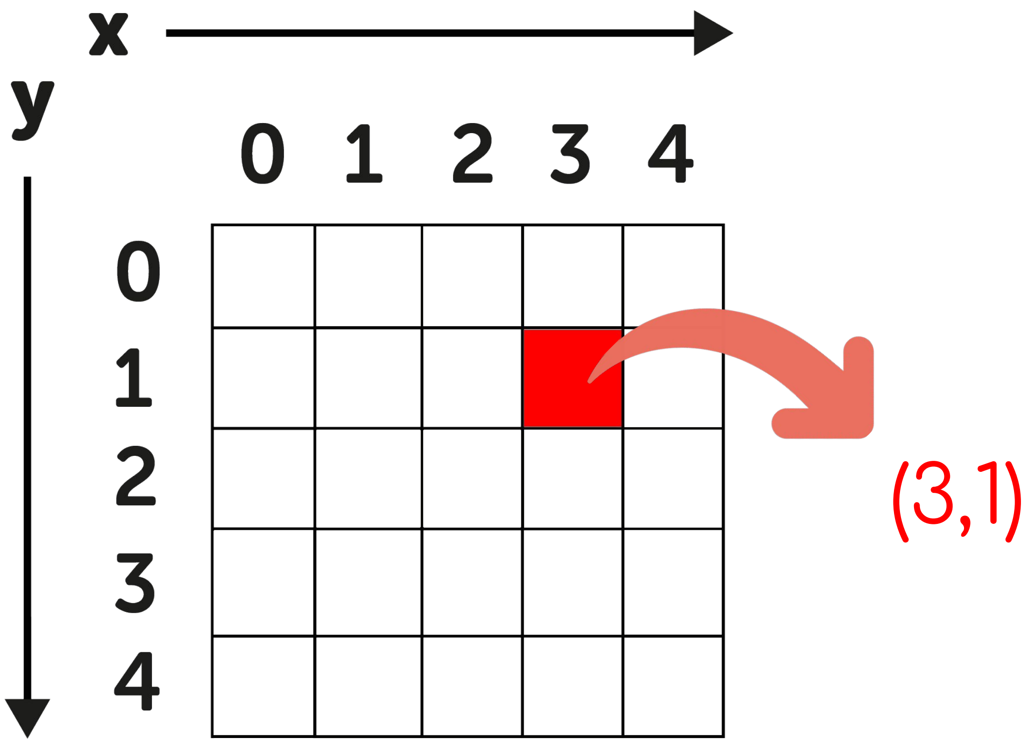
The agent would exploit only **visual information** (i.e., screen pixels
color values) in order to properly **move the mouse pointer**
live, in a ***human-mimicking*** *fashion*

The game level



- Goal:
  Chase the target
- Duration:
  60 seconds
- Score:
  # target hits

Interlude: mouse position on the screen



In our project:

$$mp = (X, Y)$$

$$X \in [0,1920]$$
$$Y \in [0,1080]$$

Data Collection

We played the target level multiple times and collected a data point every **1/20th** of a second

Data Point (time t)

Screenshot$_{t-1}$ (1920x1080)

- ➢ Mouse Movement on the X axis: $X_t - X_{t-1}$
  - ○ Int (px) ∈ [-1920,1920]
- ➢ Mouse Movement on the Y axis: $Y_t - Y_{t-1}$
  - ○ Int (px) ∈ [-1080,1080]

Data Collection "Hack"

Although we aimed at training a model that would reproduce our *far-from-being-perfect playstyle*, we nonetheless laid down some *data collecting rules* in an attempt to **improve the quality** of the input data:

- **Do not miss!**
  - *Only **100%** shooting accuracy levels data saved*

- **Move!**
  - *No standing still data point - **(0,0)** movement vector - allowed*

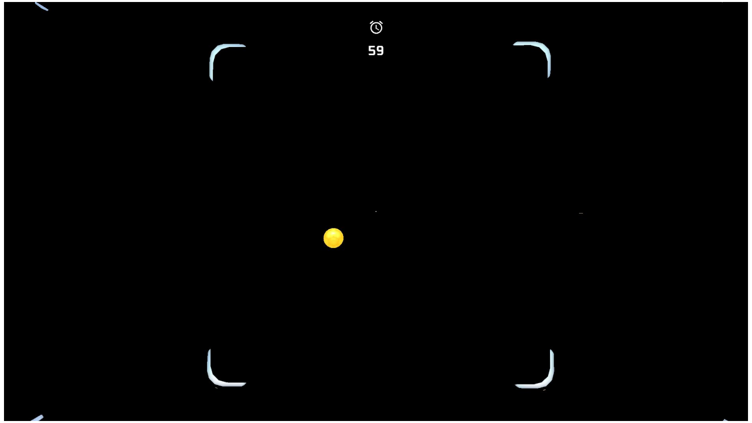
Data preprocessing

Data preprocessing - Masking

BGR Boundaries

- Lower = [0,150,150]
- Upper = [255,255,255]

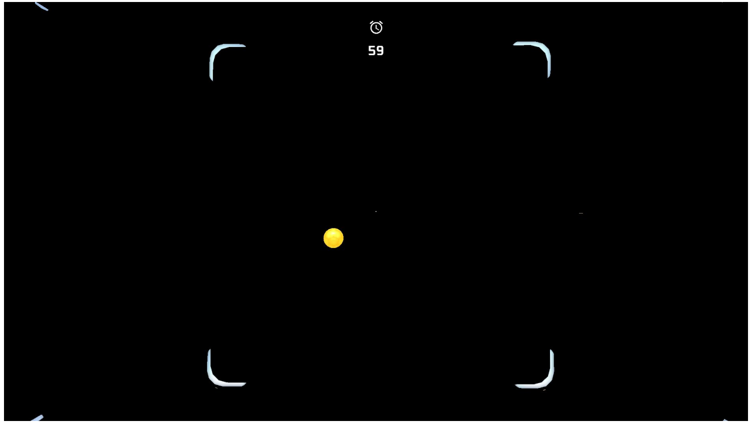
Original Screenshot



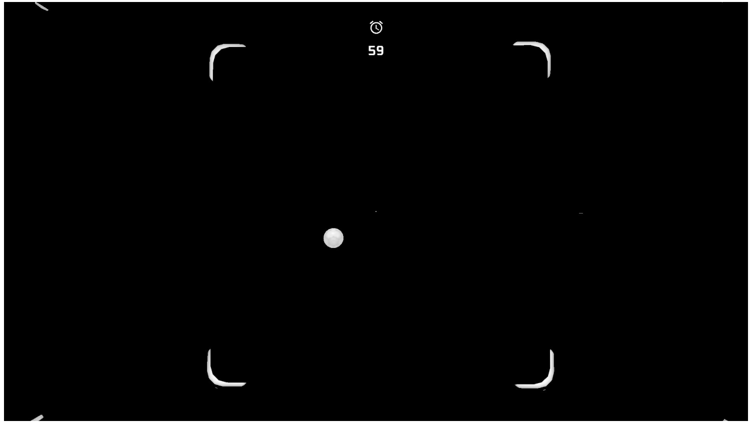
Masked Screenshot

Data preprocessing - From BRG to Grayscale

$$Y = 0.299\ R + 0.587\ G + 0.114\ B$$



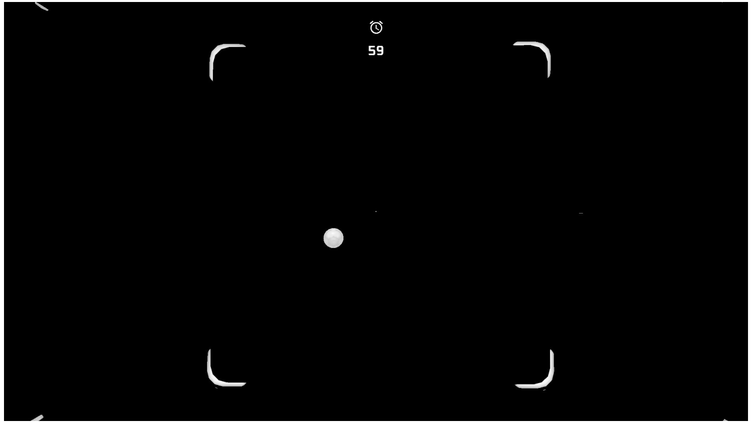
Masked Screenshot



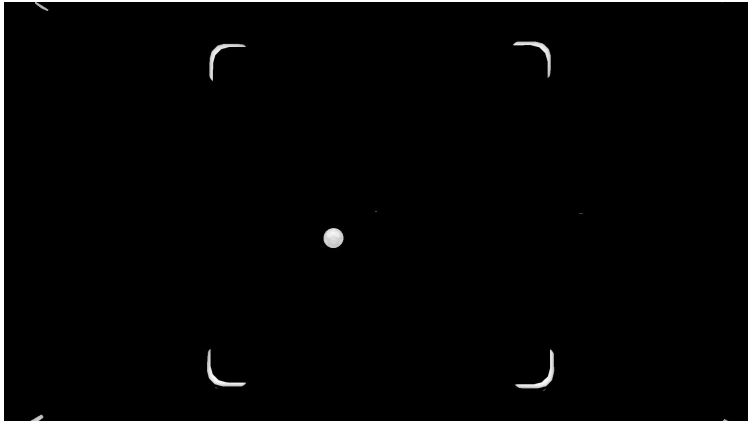
Grayscale Screenshot

Data preprocessing - Removing clock & time

Blackened the corresponding pixel (set their value to 0)



Grayscale Screenshot



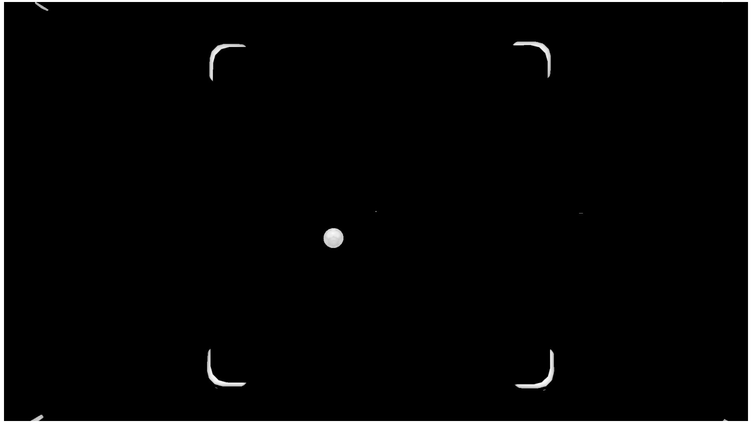
Preprocessed Screenshot

Mouse movement: regression model

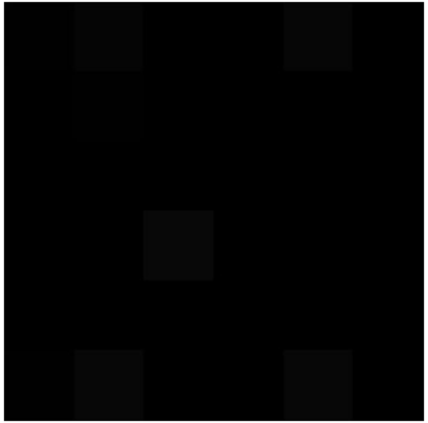
Data preprocessing - Regression: resizing

New size: (6x6) - Grid search approach

Algorithm: INTER_AREA (resampling using pixel area relation)

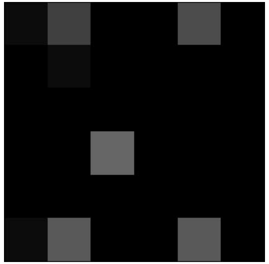


Preprocessed Screenshot (1920x1080)

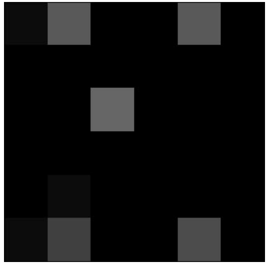


Resized Screenshot (6x6)

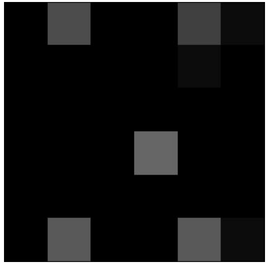
Data augmentation - Regression



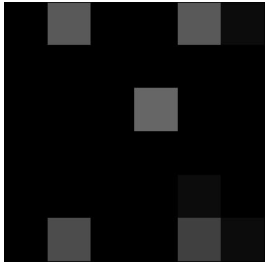
Original
(X_vec, Y_vec)



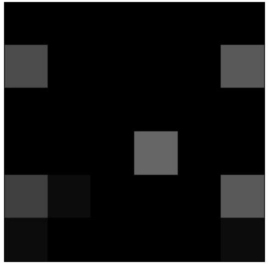
Flip around X-axis
(X_vec, -Y_vec)



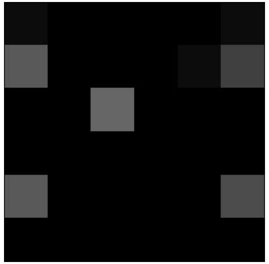
Flip around Y-axis
(-X_vec, Y_vec)



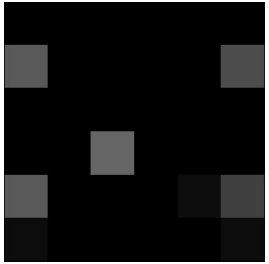
Flip around both axis
(-X_vec, -Y_vec)



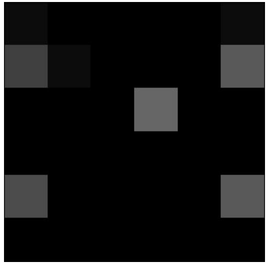
Rotate 90°
CounterClock
(Y_vec, -X_vec)



Rotate 90°
Clock
(-Y_vec, X_vec)

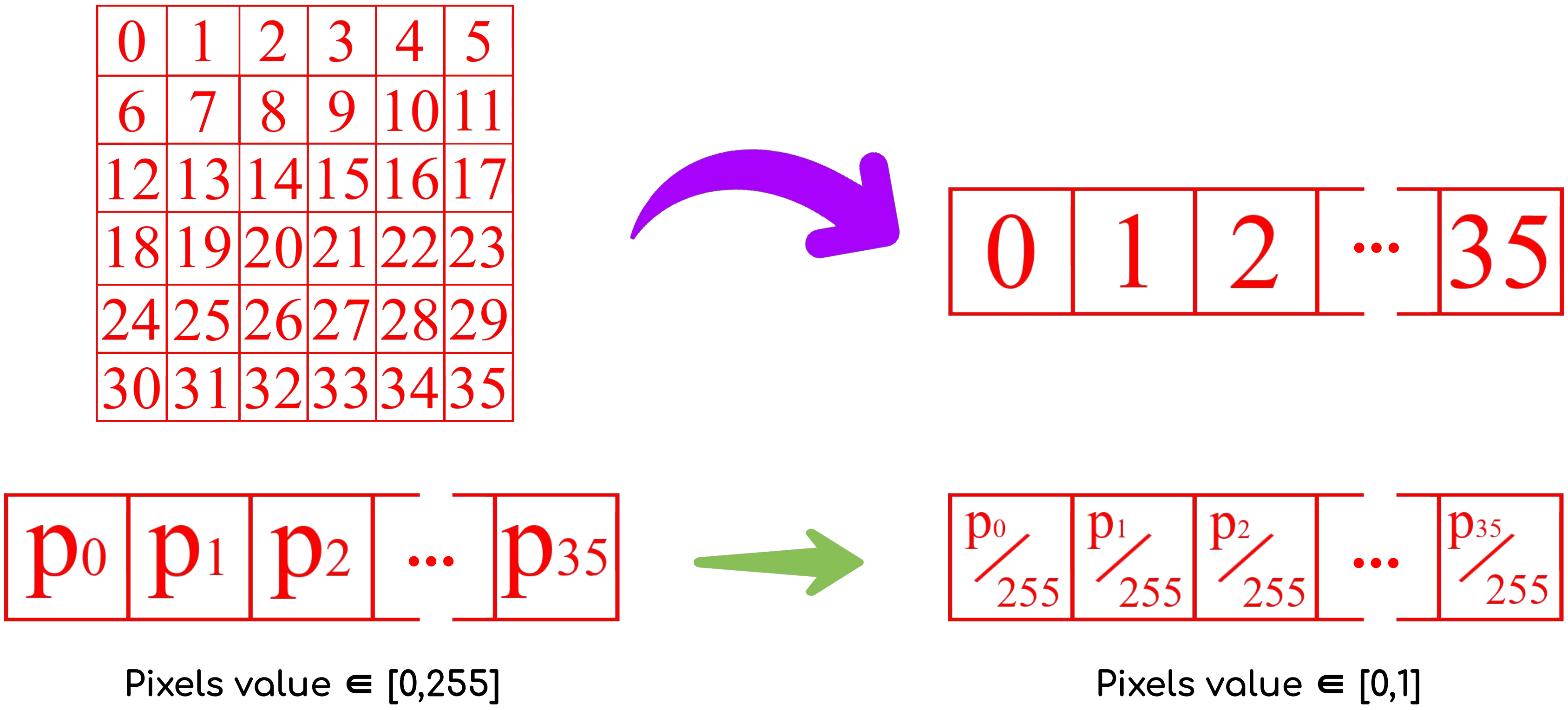


Flip around X-axis &
Rotate 90° CounterClock
(-Y_vec, -X_vec)



Flip around X-axis
& Rotate 90° Clock
(Y_vec, X_vec)

Data preprocessing - Regression: flattening & scaling



Target variables description

	x_mov	y_mov
count	20398.000	20398.000
mean	0.323	-0.208
std	59.260	50.298
min	-272.000	-170.000
25%	-25.000	-20.000
50%	0.000	0.000
75%	24.000	16.000
max	326.000	298.000

Before data augmentation

	x_mov	y_mov
count	163184.000	163184.000
mean	0.000	0.000
std	54.962	54.962
min	-326.000	-326.000
25%	-22.000	-22.000
50%	0.000	0.000
75%	22.000	22.000
max	326.000	326.000

After data augmentation

Model Training: Train/Test Split

75% Train - **25%** Test Size

Mouse movement model: Random Forest regression

INPUT: 36 pixel values

OUTPUT: Two-dimensional vector **(X_mov, Y_mov)**

Multi-Output Strategy:

Fitting **one regressor per target** (sharing the *same parameters*, each one having as target a *different output component*: **X_mov** and **Y_mov**)

Score the model according to the **arithmetic average** of the individual regressor scores (different scoring strategies might be possible)

Random Forest Fitting: 5-Fold CV

Parameters: *n_estimators = 100*

Scoring Metric: $$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Mouse movement model: grid search approach

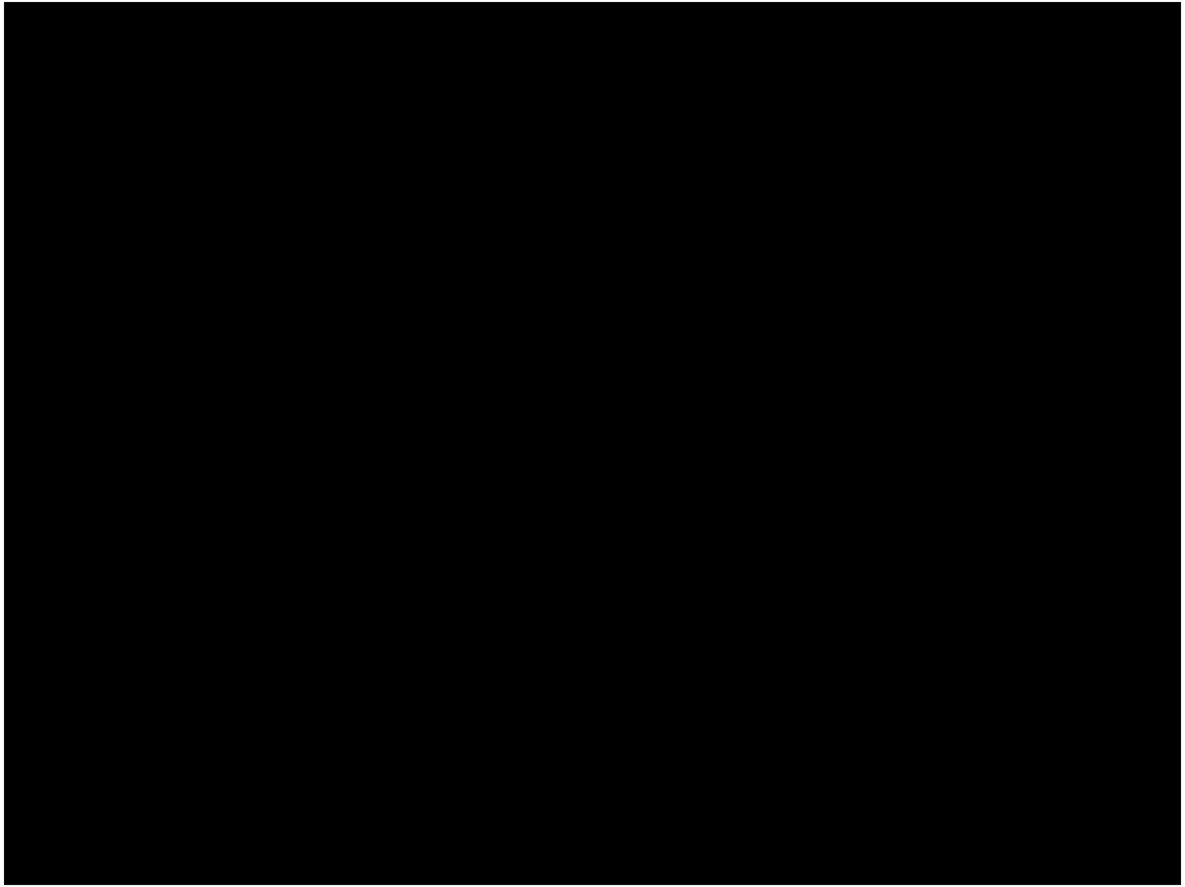
Input screenshot size	RMSE score
4,4	35.1
5,4	37.7
4,5	36.2
5,5	38.2
6,6	33.0
7,7	35.2
8,8	34.4
10,10	35.8

Best Model

Input size = (6x6)

5-Fold CV RMSE = 33.0

Qualitative model assessment - Live Action!



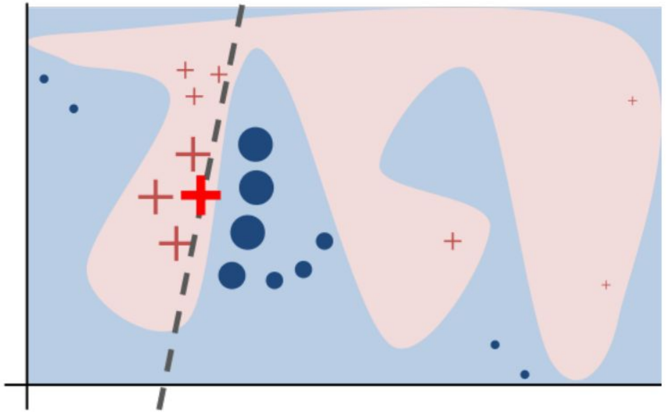
Interpretable Machine Learning (IML)

"Why Should I Trust You?"

An analysis based on the work and paper by *Marco Tulio Ribeiro et al.* (2016)

Local Interpretable Model-agnostic Explanations (LIME)

Goal: *identify an **interpretable model** over the interpretable representation that is **locally faithful** to the classifier/regressor*



$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \; \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Explanation
(intelligible)

Unfaithfulness
(as opposed to local fidelity)

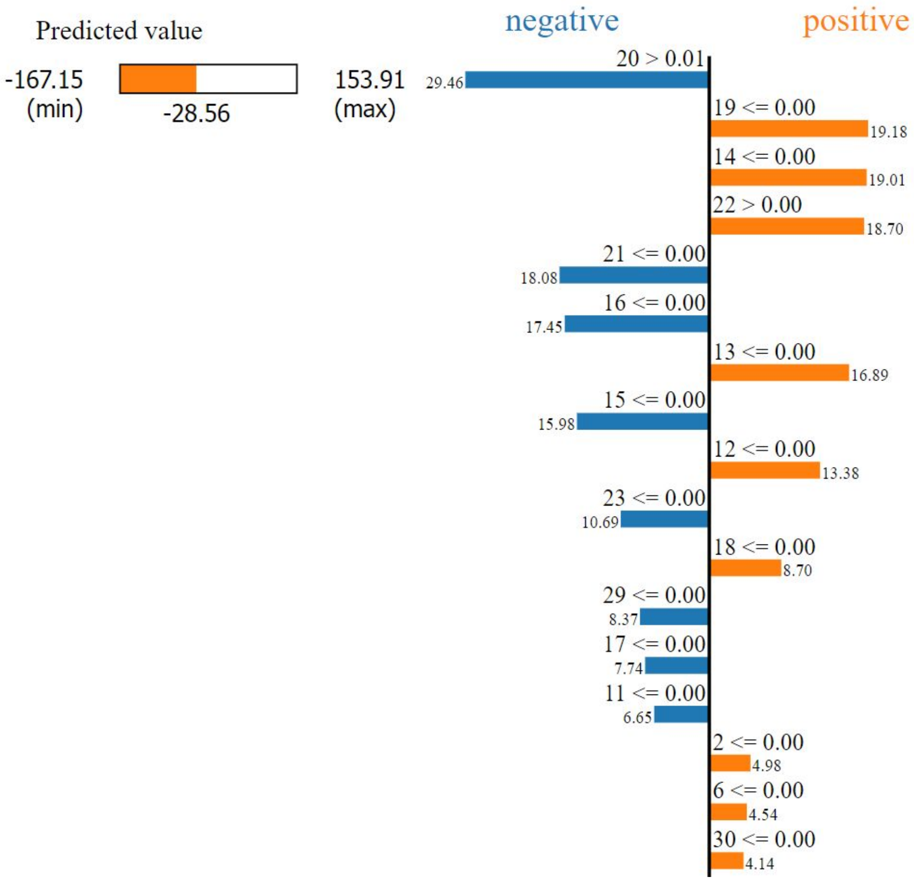
Complexity
(as opposed to interpretability)

LIME in action

RF Prediction: (-28.56, 1.54)

Actual Response: (-84, 0)

LIME in action: X-axis



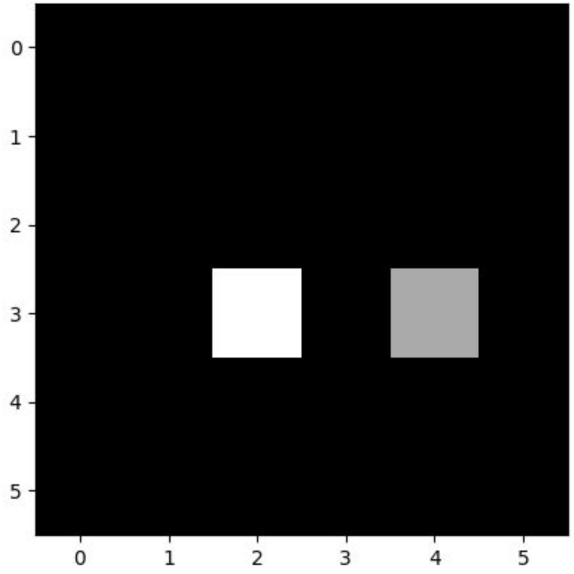
Feature	Value
20	0.04
19	0.00
14	0.00
22	0.02
21	0.00
16	0.00
13	0.00
15	0.00
12	0.00
23	0.00
18	0.00
29	0.00
17	0.00

Intercept: -13.41

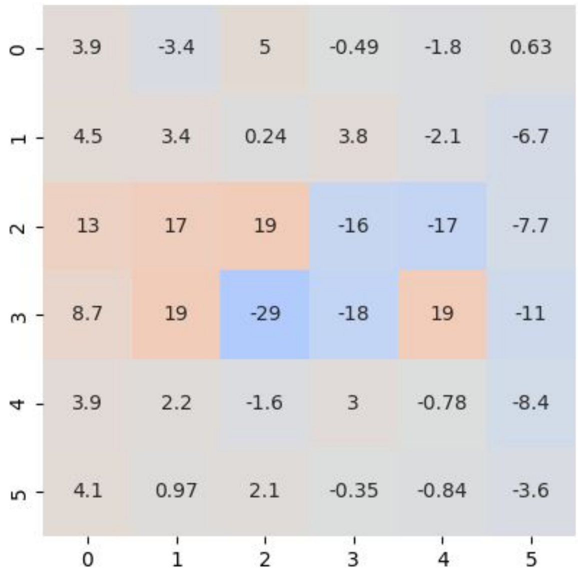
Score (R^2): 0.296

LIME Prediction: -9.22

LIME in action: X-axis

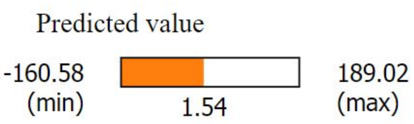


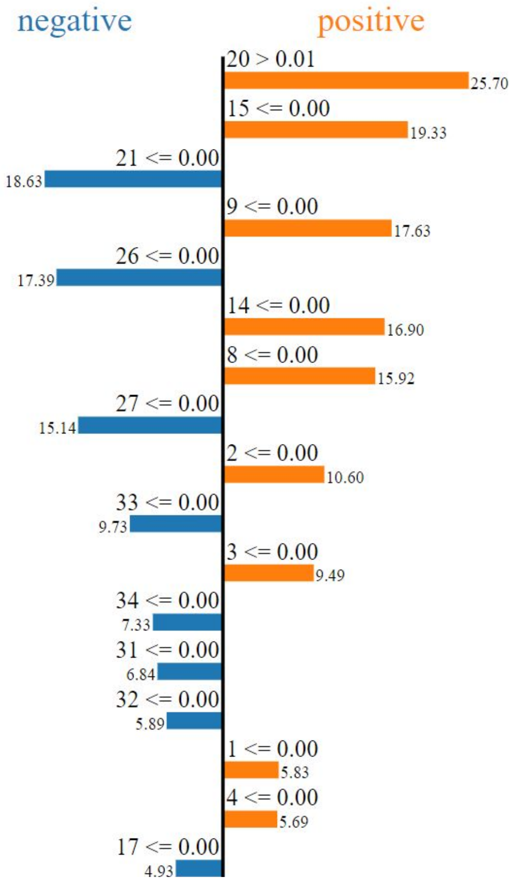
Input Image



LIME Explanation

LIME in action: Y-axis





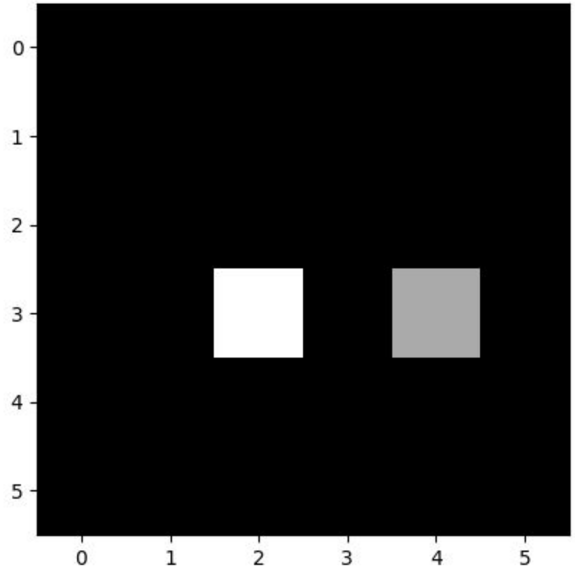
Feature	Value
20	0.04
15	0.00
21	0.00
9	0.00
26	0.00
14	0.00
8	0.00
27	0.00
2	0.00
33	0.00
3	0.00
34	0.00
31	0.00

Intercept: -13.29

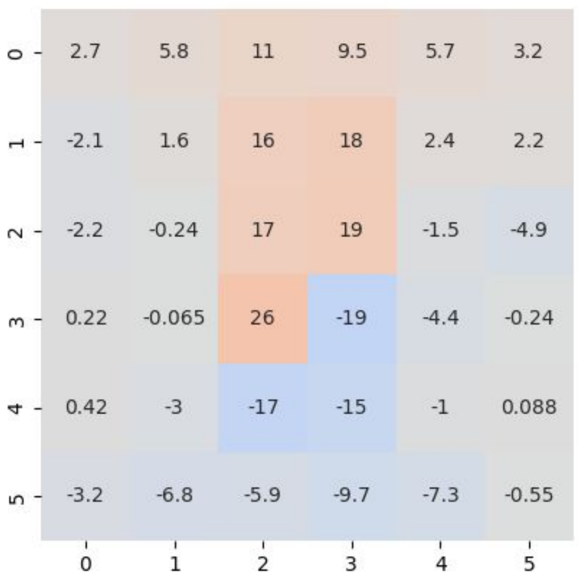
Score (R^2): 0.269

LIME Prediction: 22.3

LIME in action: Y-axis



Input Image



LIME Explanation

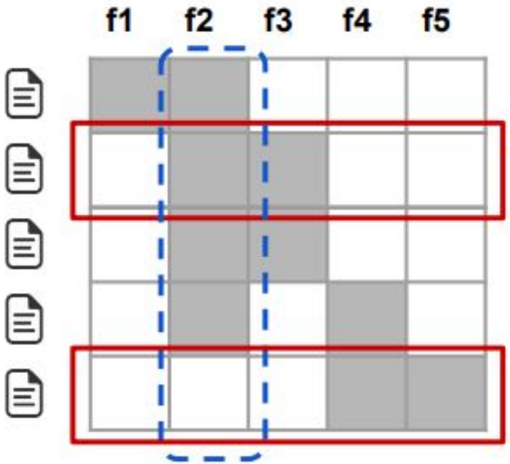
Submodular Pick method for explaining models

Goal: pick a *diverse, representative* **set** of **non-redundant explanations** that represent how the model behaves **globally**

$$Pick(\mathcal{W}, I) = \underset{V, |V| \leq B}{\operatorname{argmax}}\, c(V, \mathcal{W}, I)$$

NP hard problem!

Greedy approach



***Explanation matrix W**: represents the **local importance** of the interpretable components for each instance*

I knew I should not have trusted you...

Model issues:

- **LIME explanations** showed that the model at times failed in discerning whether a **grey pixel** in input corresponded to a **target** or to one of the **corners**
- **Qualitative assessment** highlighted some difficulties in reaching the actual target when the cursor was in its **neighborhood** right **above** or **below** it

Our implemented preprocessing solution:

- *Double Masking + Binary Pixel Color Conversion*
- *Higher resolution input image*

Data preprocessing - Double masking + BPCC

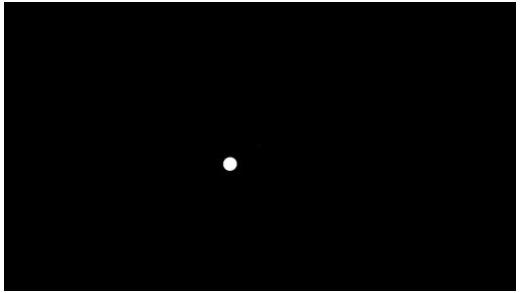
Original Screenshot

BGR Boundaries

Upper = [0,0,252]
Lower = [255,255,255]

+

Binary Pixel
Color Conversion



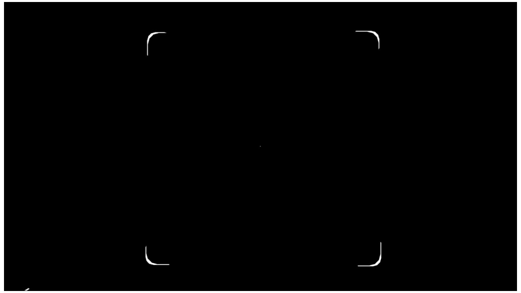
Target

BGR Boundaries

Upper = [240,100,0]
Lower = [255,255,255]

+

Binary Pixel
Color Conversion



Corners

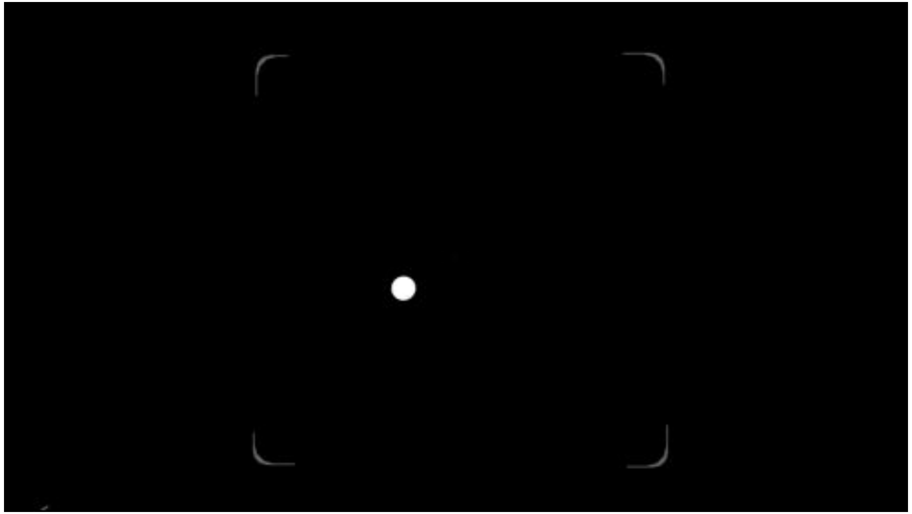
Binary Pixel Color Conversion

$$y = \begin{cases} 255 \quad if \quad x > 0 \\ 0 \quad if \quad x = 0 \end{cases}$$

Data preprocessing: Double Masking

Formula: $$Target + Corners * \frac{k}{255}$$

$$k = 100$$

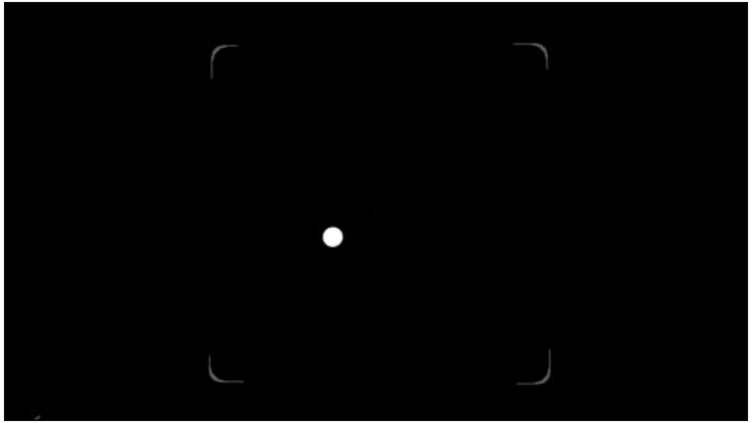


Final Screenshot

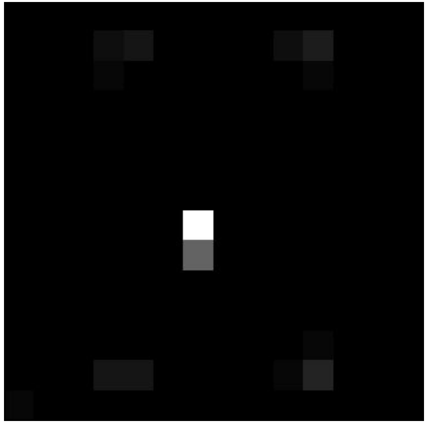
Data preprocessing - Regression: resizing

New size: (14x14) - Grid search approach

Algorithm: INTER_AREA (resampling using pixel area relation)



Preprocessed Screenshot (1920x1080)



Resized Screenshot (14x14)

Mouse movement model: grid search approach

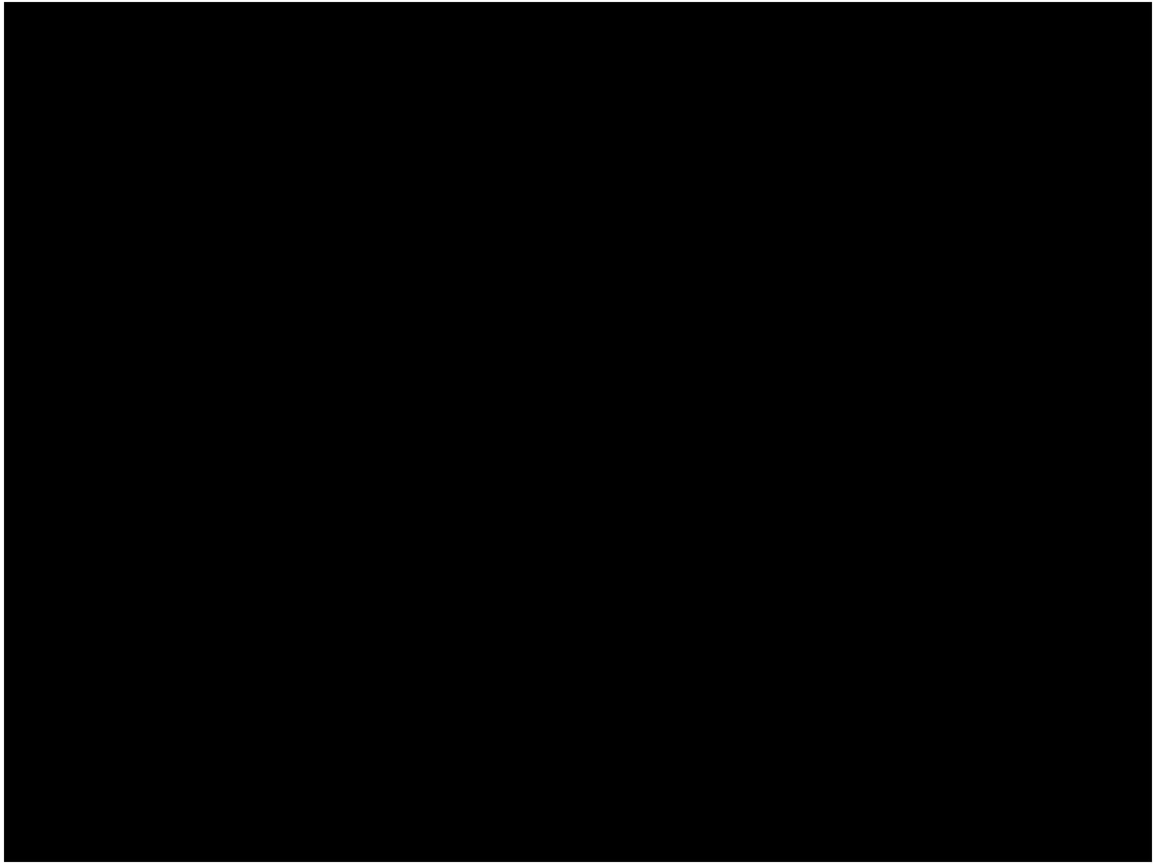
Input screenshot size	RMSE score
10,10	30.1
12,12	29.3
14,14	28.8
16,16	29.0
20,20	29.4
24,24	29.7

Best Model

Input size = (14x14)

5-Fold CV RMSE = 28.8

Qualitative model assessment - Live Action!

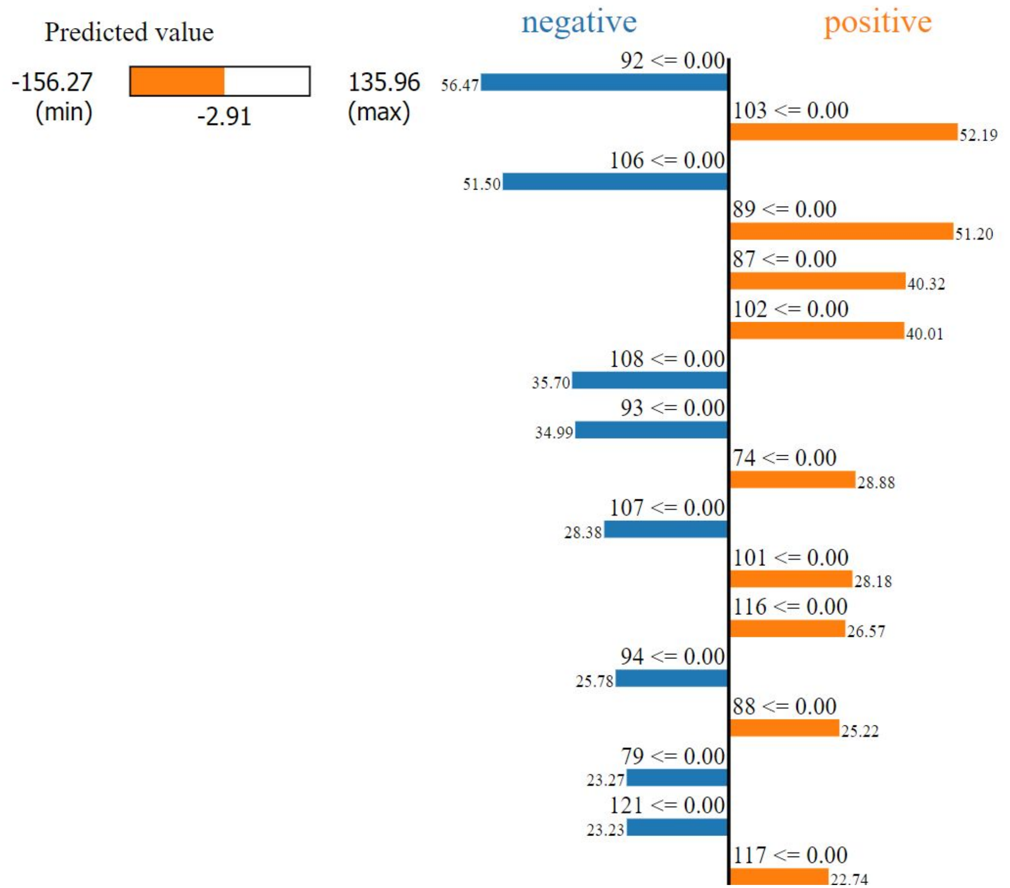


LIME in action

RF Prediction: (-2.91, -19.96)

Actual Response: (-10, -31)

LIME in action: X-axis



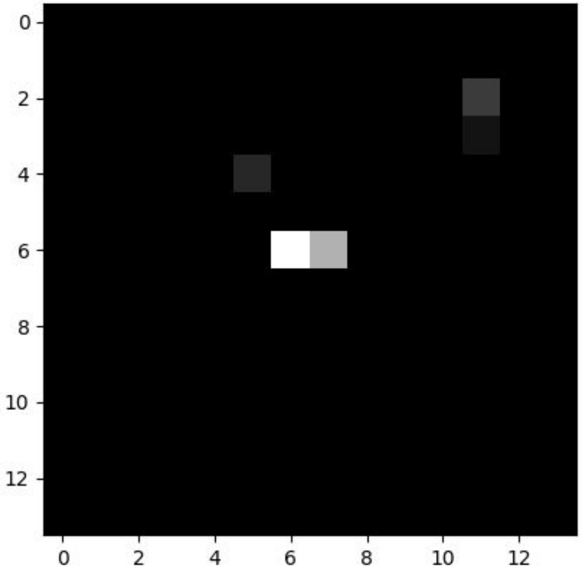
Feature	Value
92	0.00
103	0.00
106	0.00
89	0.00
87	0.00
102	0.00
108	0.00
93	0.00
74	0.00
107	0.00
101	0.00
116	0.00
94	0.00

Intercept: -78.87

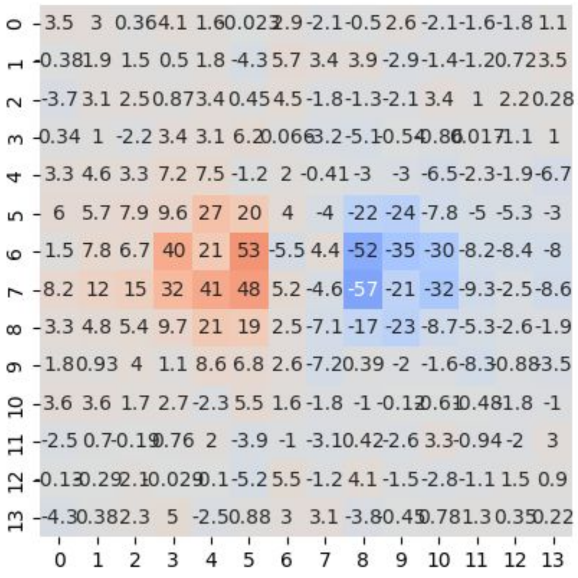
Score (R^2): 0.598

LIME Prediction: -8.65

LIME in action: X-axis



Input Image



LIME Explanation

Conclusion

By analysing the **explanations** provided by the **LIME IML** algorithm and applying **domain specific knowledge** to modify the data preprocessing pipeline, we obtained:

- **Better quantitative performance**
    - *Lower RF RMSE*
- **Better qualitative performance**
    - *Smoother playstyle*
- **Less of a "black box" and more trustworthy model**
    - *Greater explainability and higher R^2 score of LIME model*