Local explainability with SHAP

EXPLAINABLE AI IN PYTHON

Fouad Trad
Machine Learning Engineer

Global vs. local explainability

Global explainability

- Overall model behavior
- Doesn't explain individual instances

Local explainability

- Explains prediction for specific data point
- Crucial for sensitive applications

[1] Images generated by DALL-E

Heart disease dataset

age	sex	chest_pain_type	blood_pressure	ecg_results	thalassemia	target
52	1	0	125	1	3	0
53	1	0	140	0	3	0
70	1	0	145	1	3	0
61	1	0	148	1	3	0
62	0	0	138	1	2	0

`knn` : KNN classifier predicting risk of heart disease

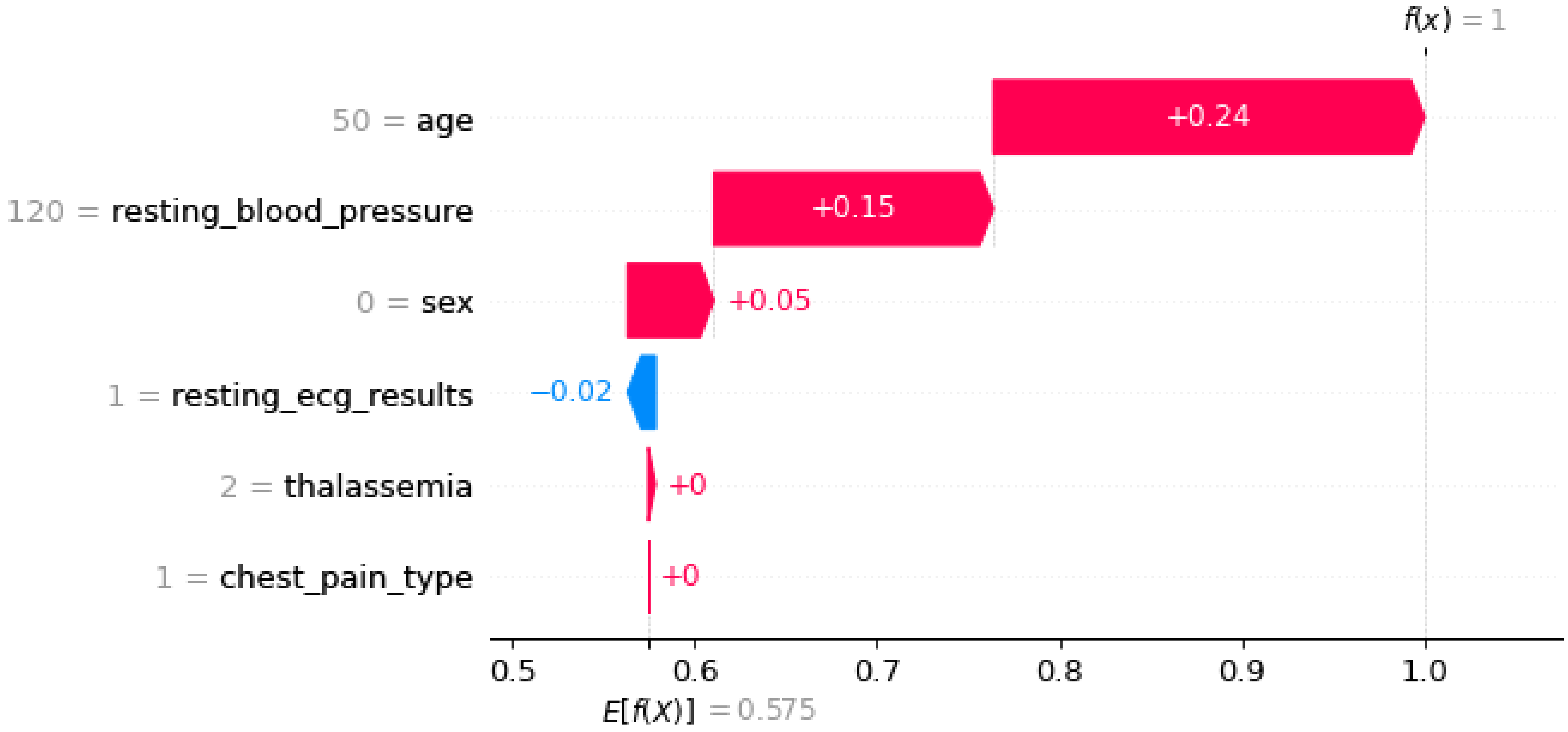
Local explainability with SHAP

```
explainer = shap.KernelExplainer(knn.predict_proba, shap.kmeans(X, 10))

test_instance = X.iloc[0, :]

shap_values = explainer.shap_values(test_instance)

print(shap_values.shape)
```

```
(6, 2)
```

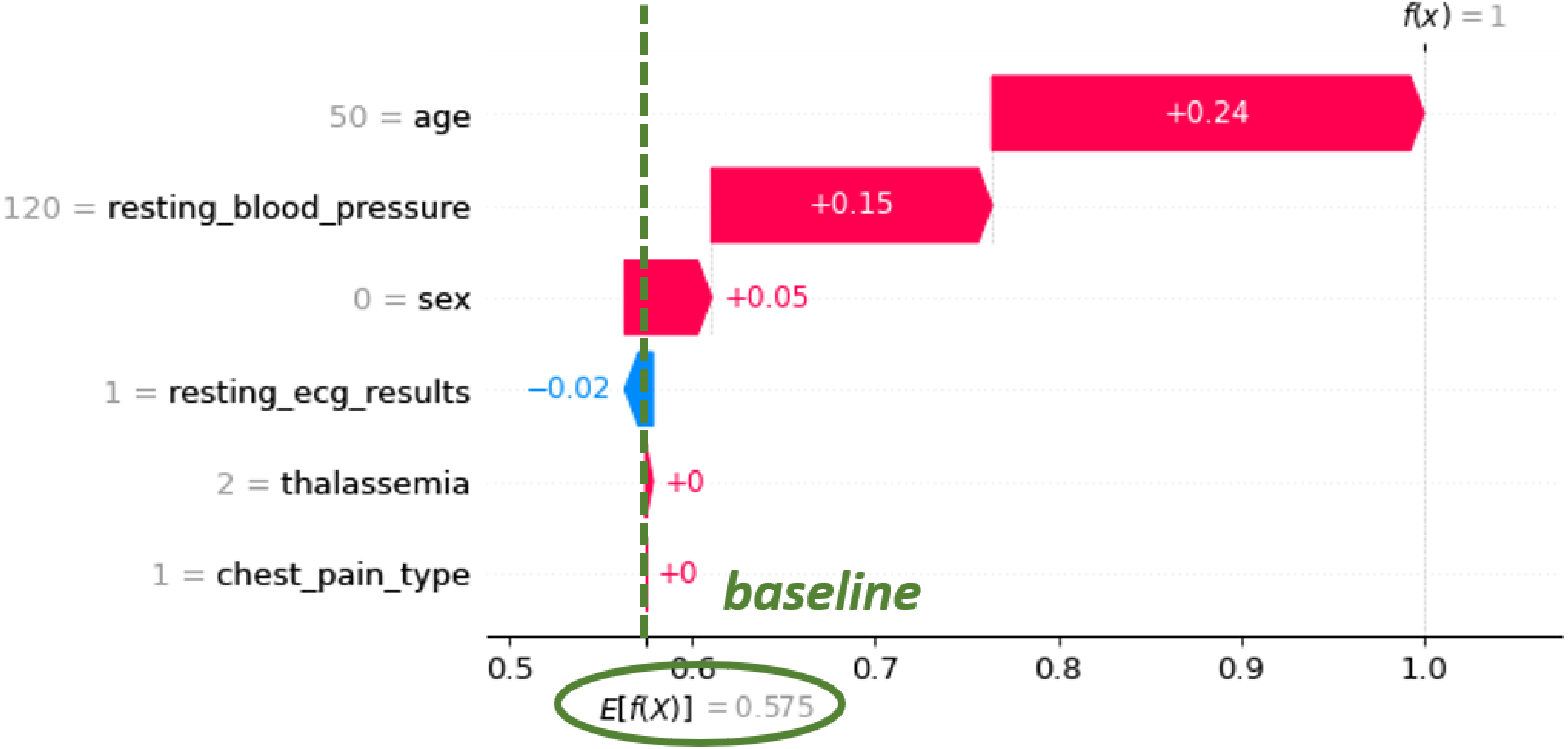
SHAP waterfall plots

- Shows how features increase or decrease model's prediction



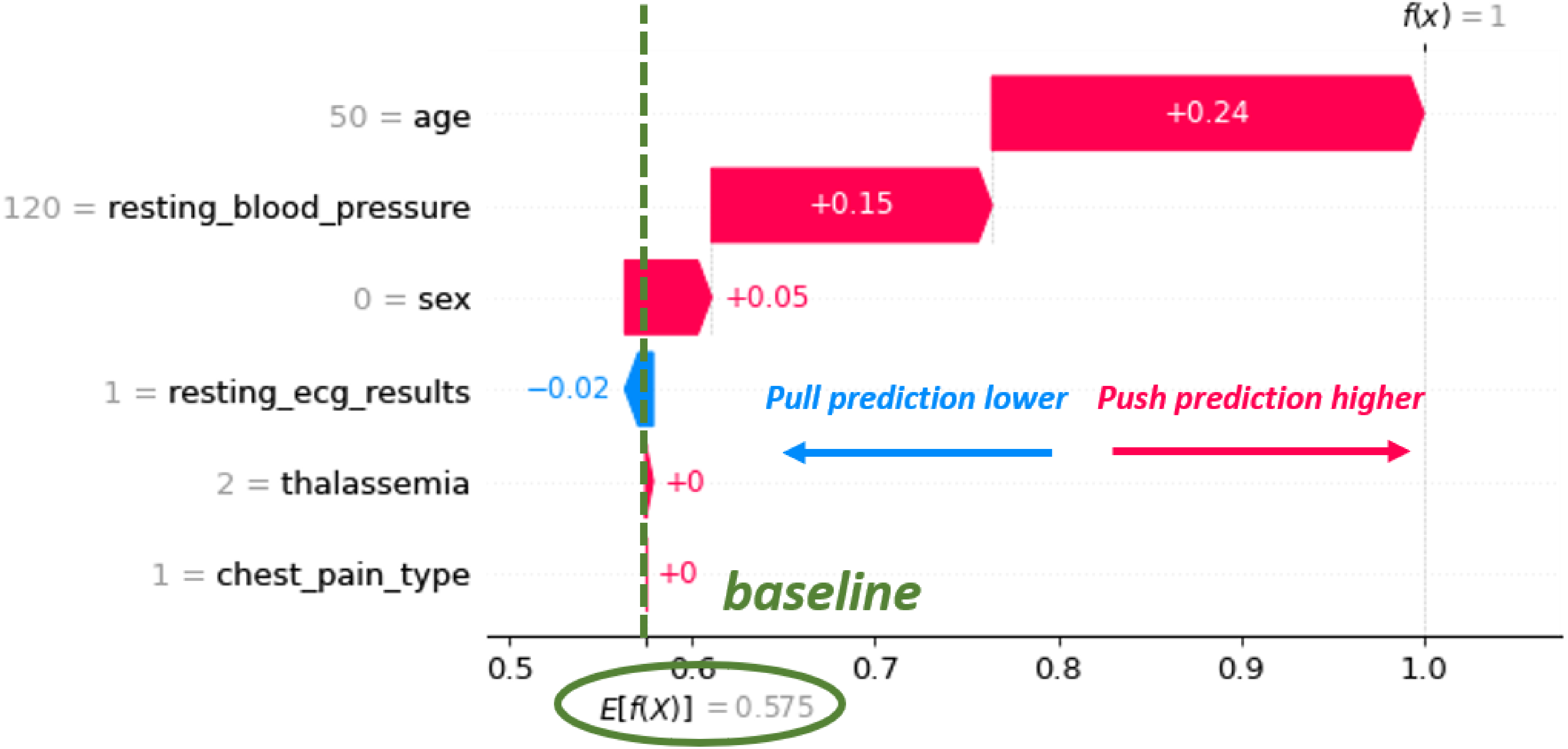
SHAP waterfall plots

- Shows how features increase or decrease model's prediction



SHAP waterfall plots

- Shows how features increase or decrease model's prediction



Pull prediction lower

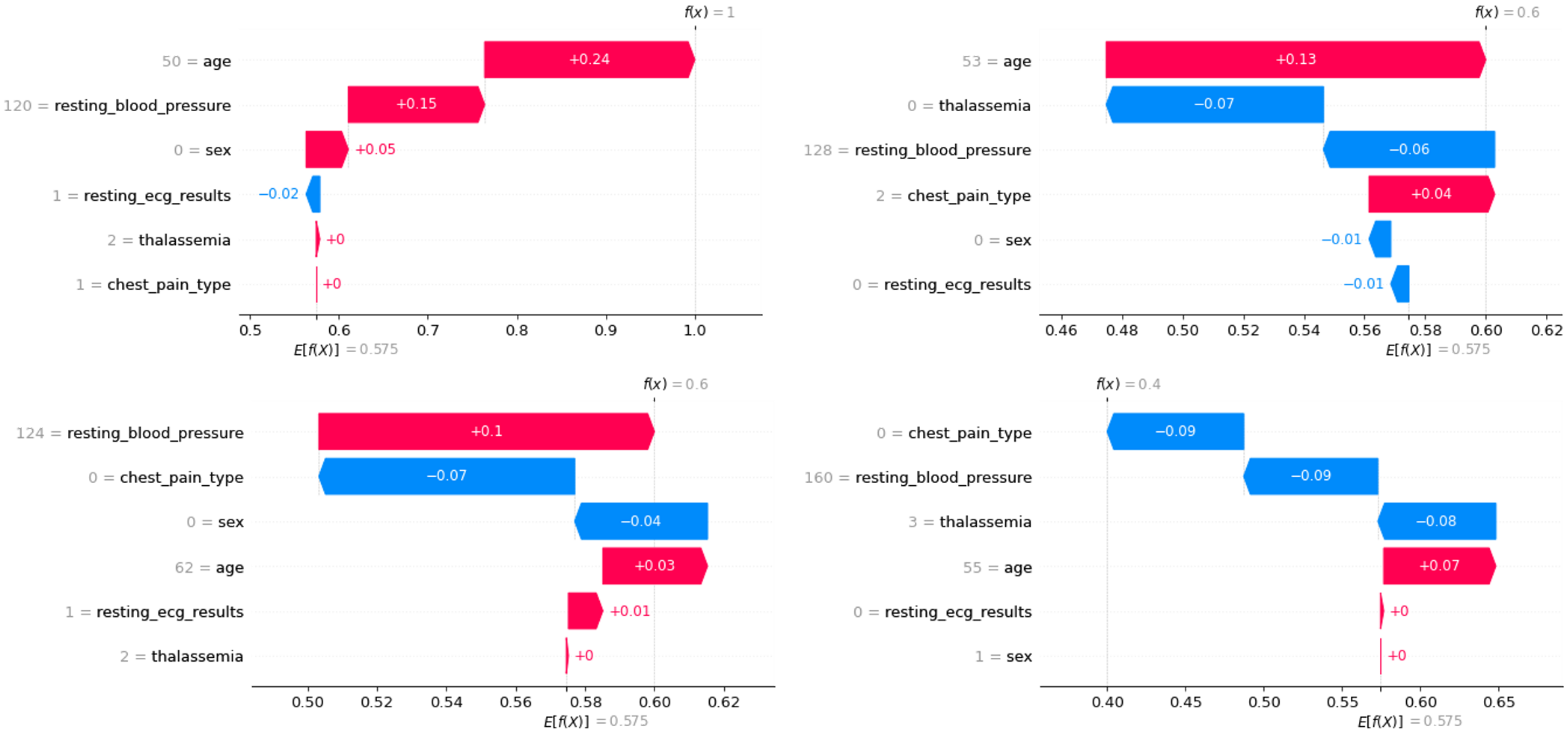
Push prediction higher

baseline

Creating waterfall plots

```
shap.waterfall_plot(
  shap.Explanation(
    values=shap_values[:,1],
    base_values=explainer.expected_value[1],
    data=test_instance,
    feature_names=X.columns
  )
)
```

Waterfalls for several instances



Let's practice!

EXPLAINABLE AI IN PYTHON

Local explainability with LIME

EXPLAINABLE AI IN PYTHON

Fouad Trad

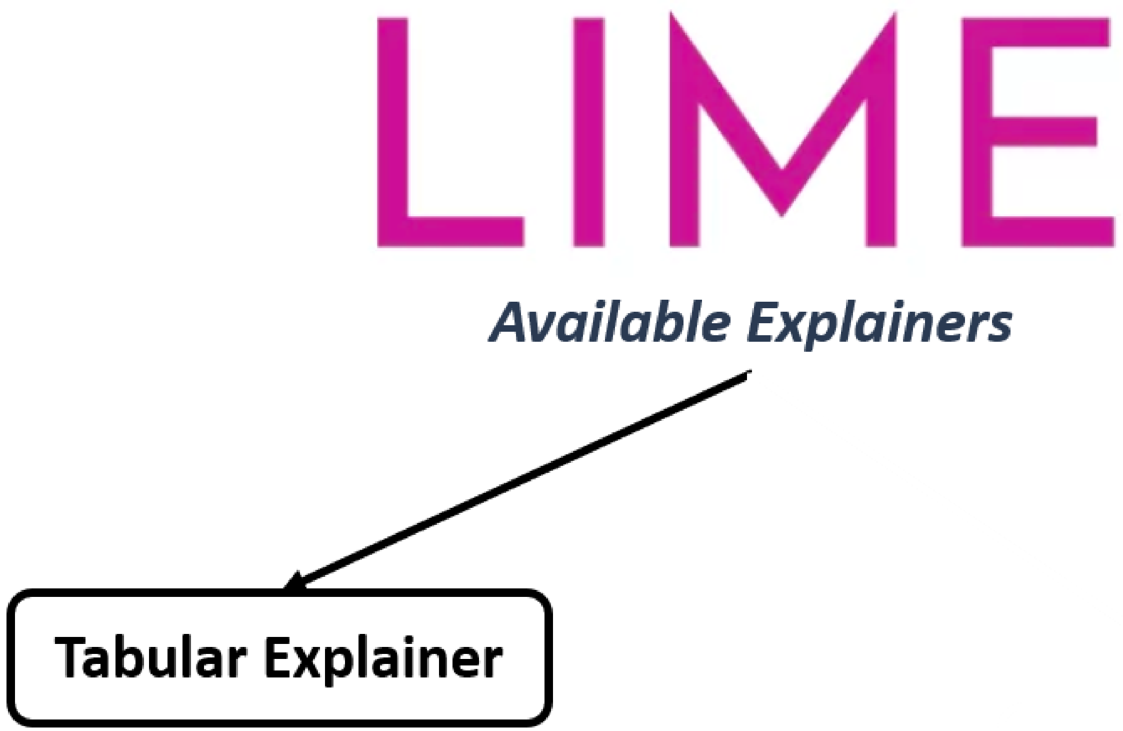
Machine Learning Engineer

- **LIME** → **L**ocal **I**nterpretable **M**odel-Agnostic **E**xplanations
- Explains predictions of complex models
- Works on individual instances
- Agnostic to model type

LIME

Lime explainers

- Tailored to different kinds of data



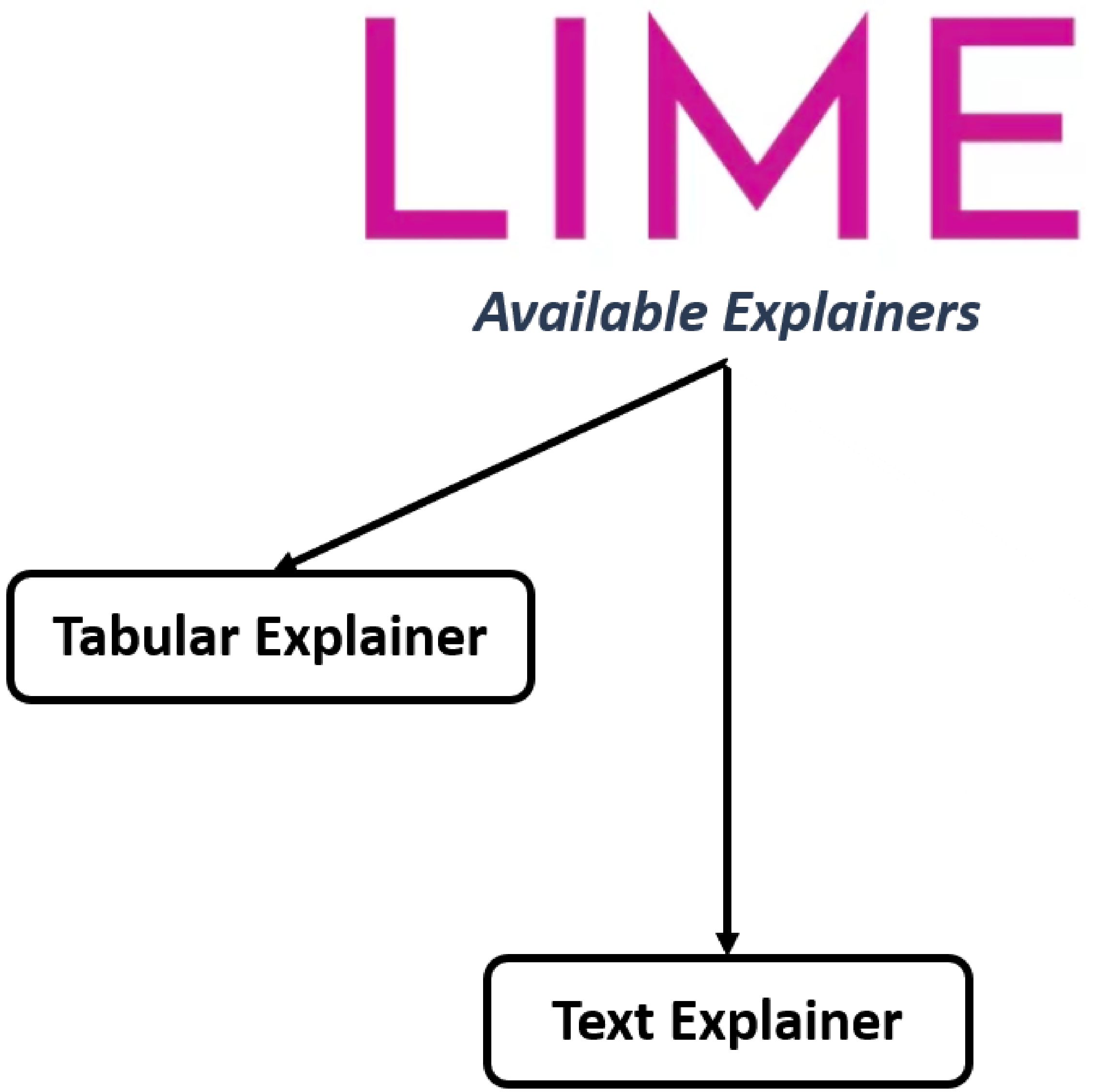
LIME

Available Explainers

Tabular Explainer

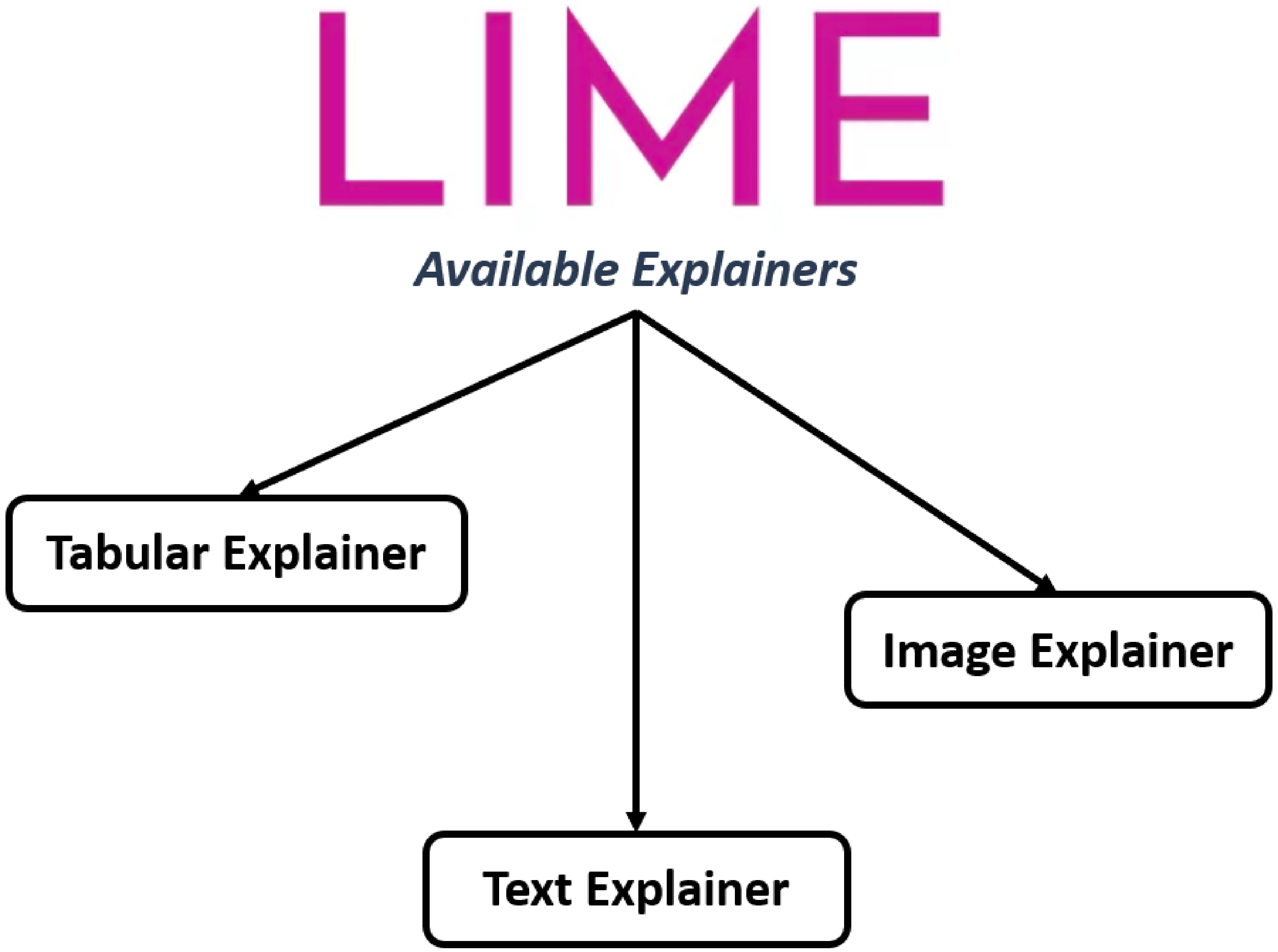
Lime explainers

- Tailored to different kinds of data



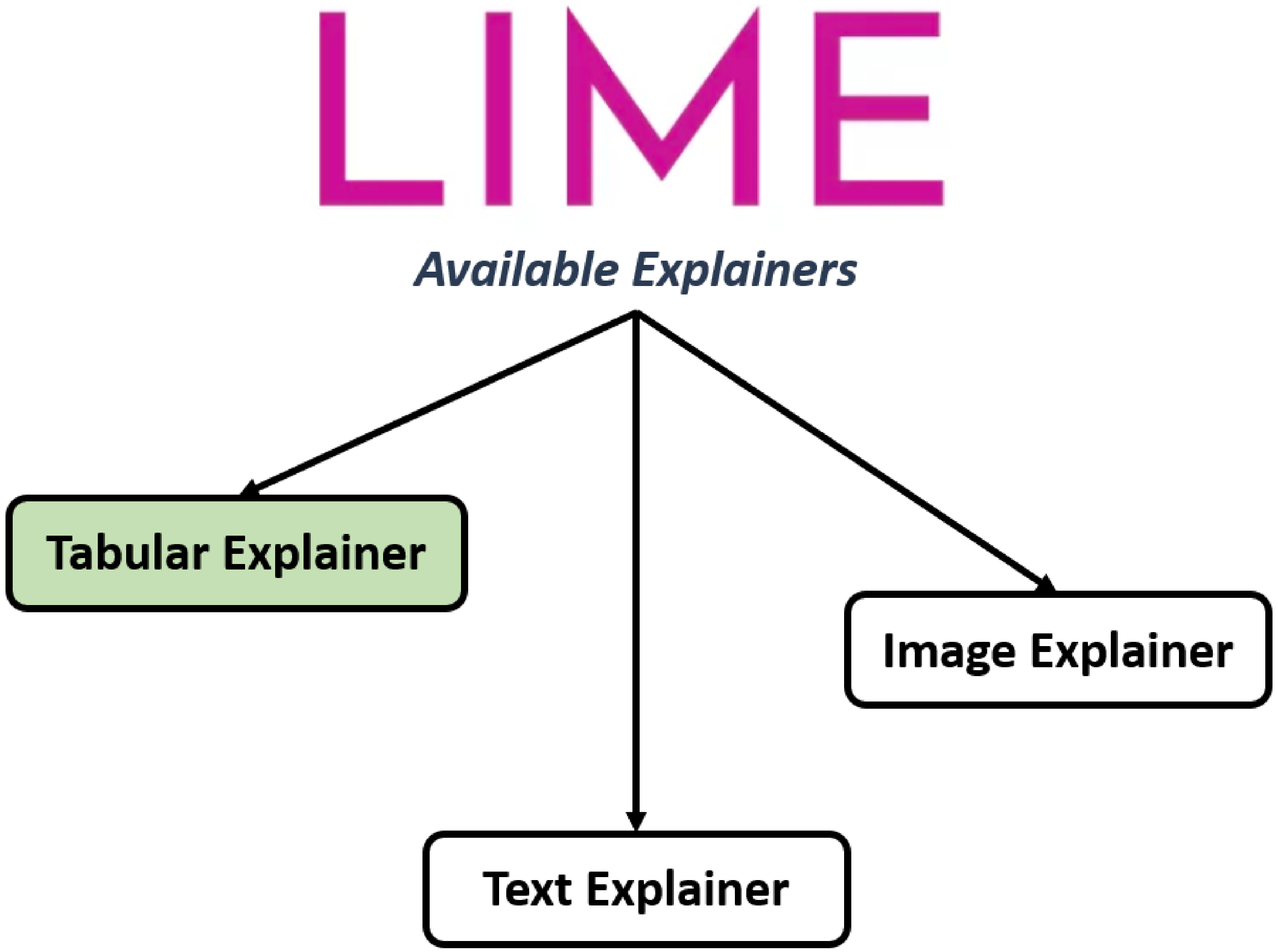
Lime explainers

- Tailored to different kinds of data
- Generates perturbations around a sample
- Sees effect on model's output
- Constructs simpler model for explanation



Lime explainers

- Tailored to different kinds of data
- Generates perturbations around a sample
- Sees effect on model's output
- Constructs simpler model for explanation



Admissions dataset

GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Chance of Admit	Accept
337	118	4	4.5	4.5	9.65	0.92	1
324	107	4	4	4.5	8.87	0.76	1
316	104	3	3	3.5	8	0.72	1
322	110	3	3.5	2.5	8.67	0.8	1
314	103	2	2	3	8.21	0.45	0

- `regressor` : predicts chance of admit
- `classifier` : predicts acceptance
- Features in `X`

Creating tabular explainer

Regression

```python
from lime.lime_tabular import LimeTabularExplainer
instance = X.iloc[1,:]

explainer_reg = LimeTabularExplainer(
  X.values,
  feature_names=X.columns,
  mode='regression'
)

explanation_reg = explainer_reg.explain_instance(
  instance.values,
  regressor.predict
)
```

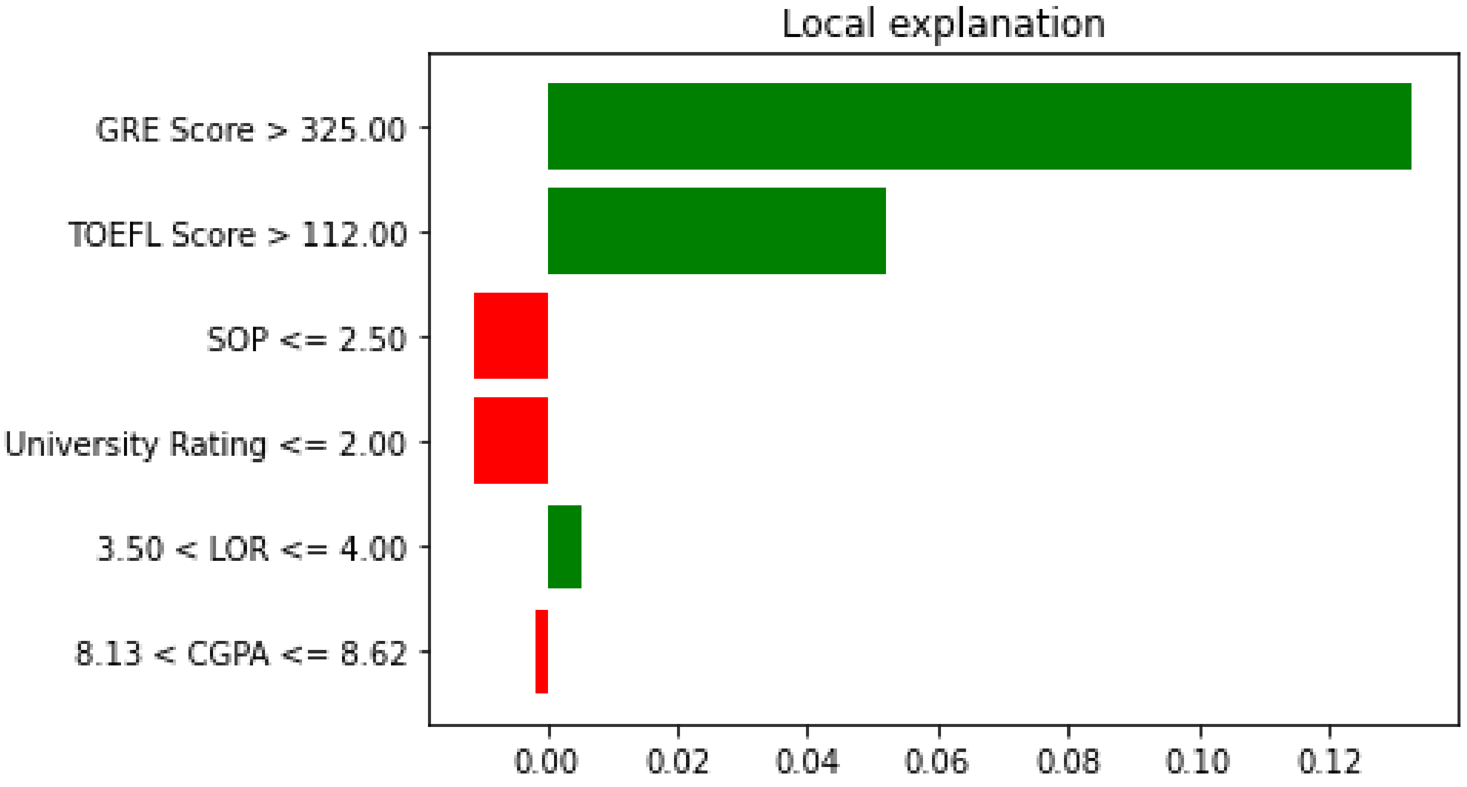
Classification

```python
from lime.lime_tabular import LimeTabularExplainer
instance = X.iloc[1,:]

explainer_class = LimeTabularExplainer(
  X.values,
  feature_names=X.columns,
  mode='classification'
)

explanation_class = explainer_class.explain_instance(
  instance.values,
  classifier.predict_proba
)
```

Visualizing explanation

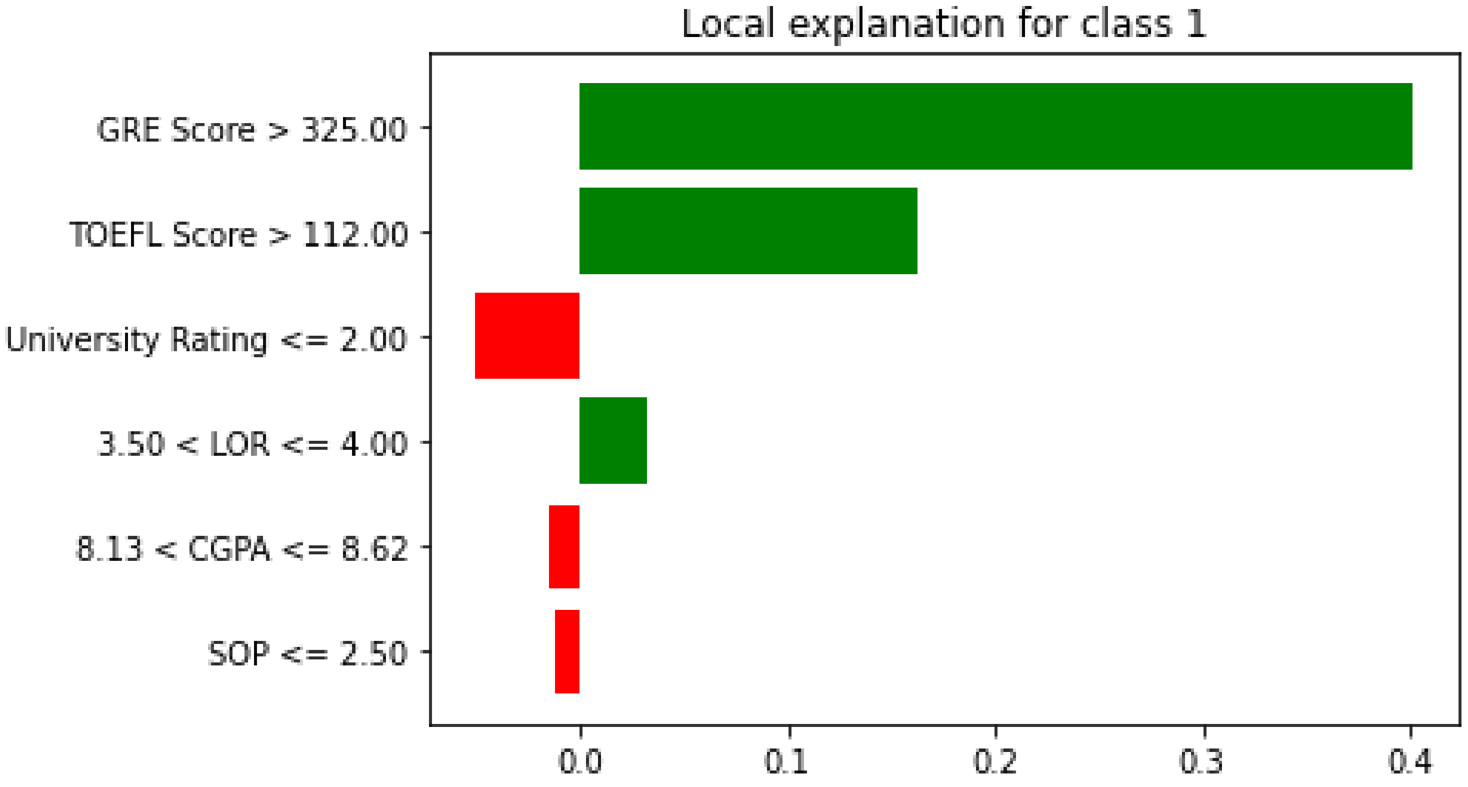
Regression

```
explanation_reg.as_pyplot_figure()
```



Classification

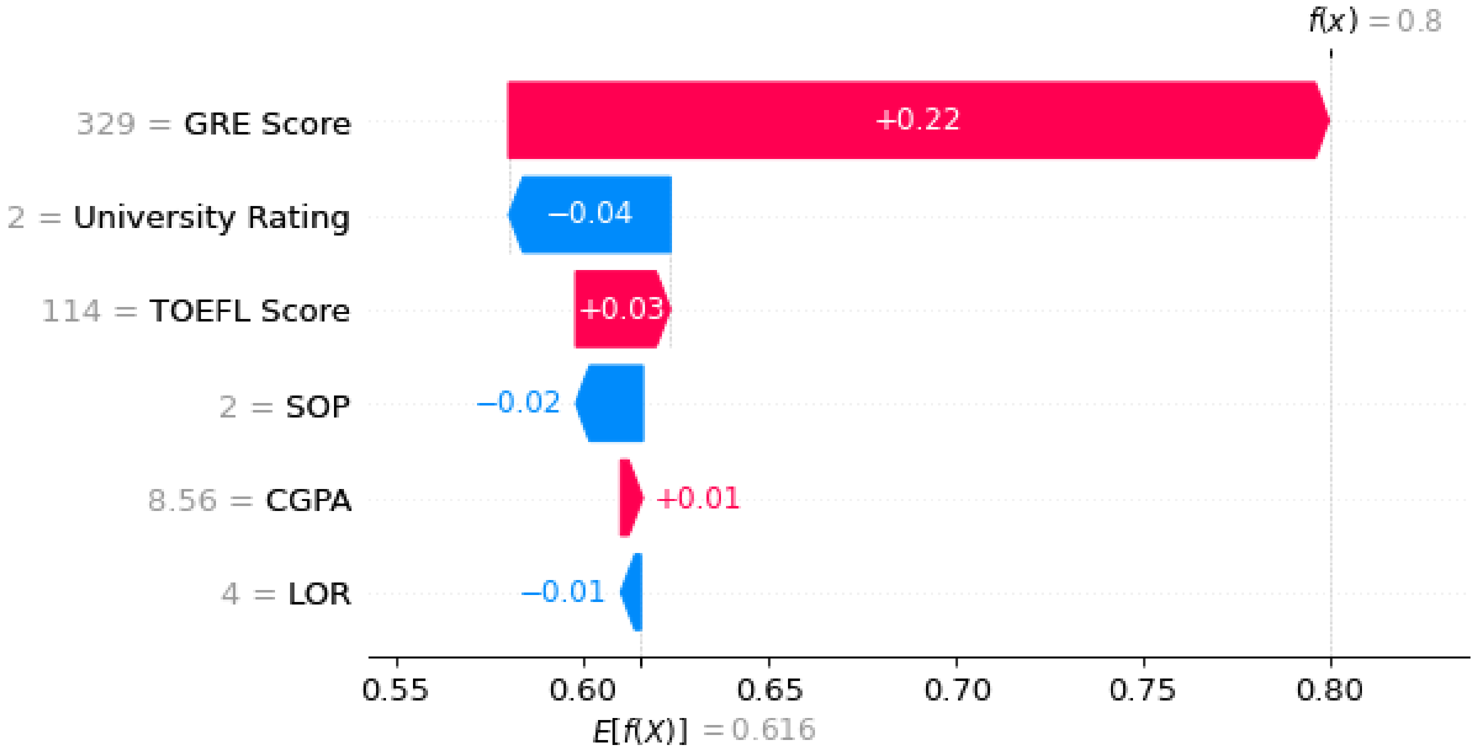
```
explanation_class.as_pyplot_figure()
```



SHAP vs. LIME

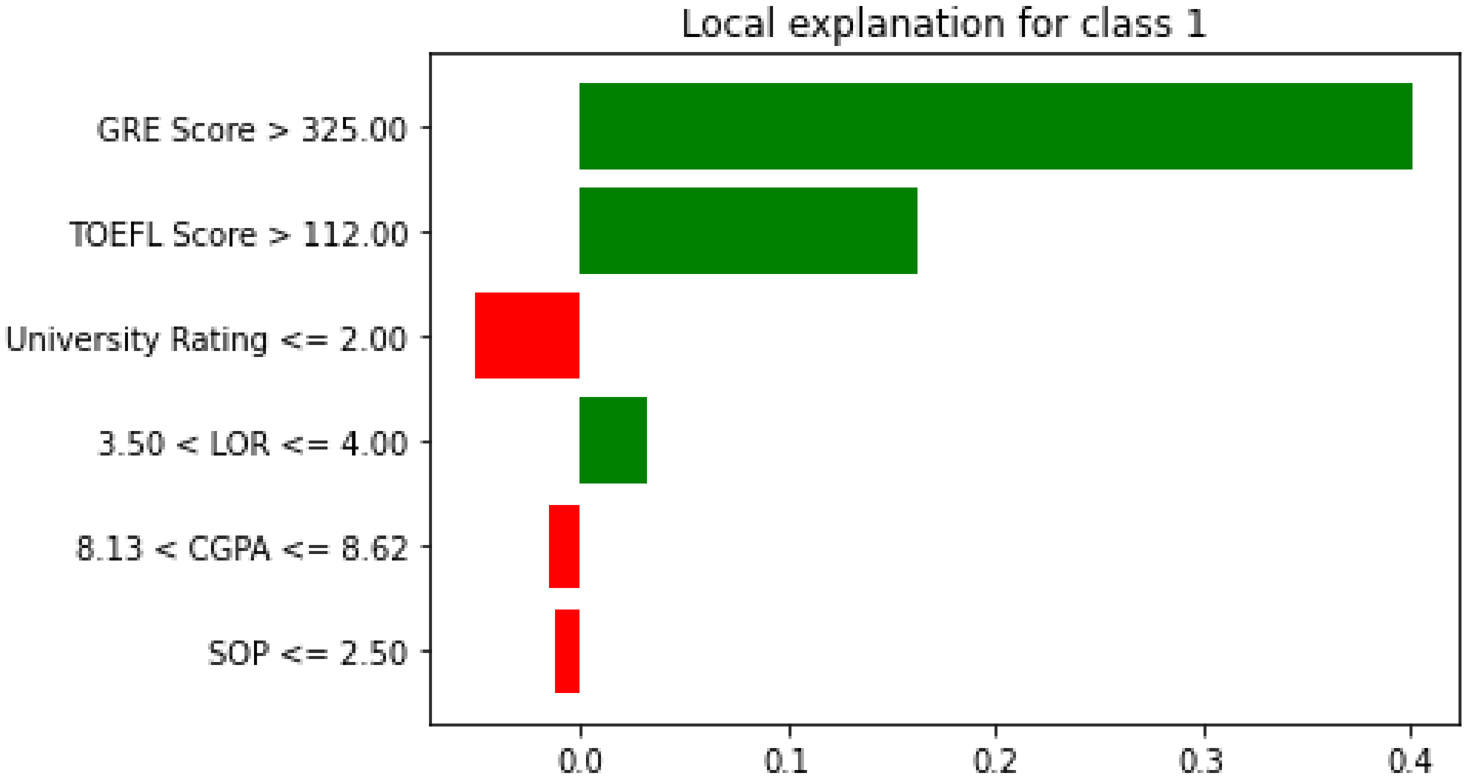
SHAP

```
shap.waterfall_plot(...)
```



LIME

```
explanation_class.as_pyplot_figure()
```



Let's practice!

EXPLAINABLE AI IN PYTHON

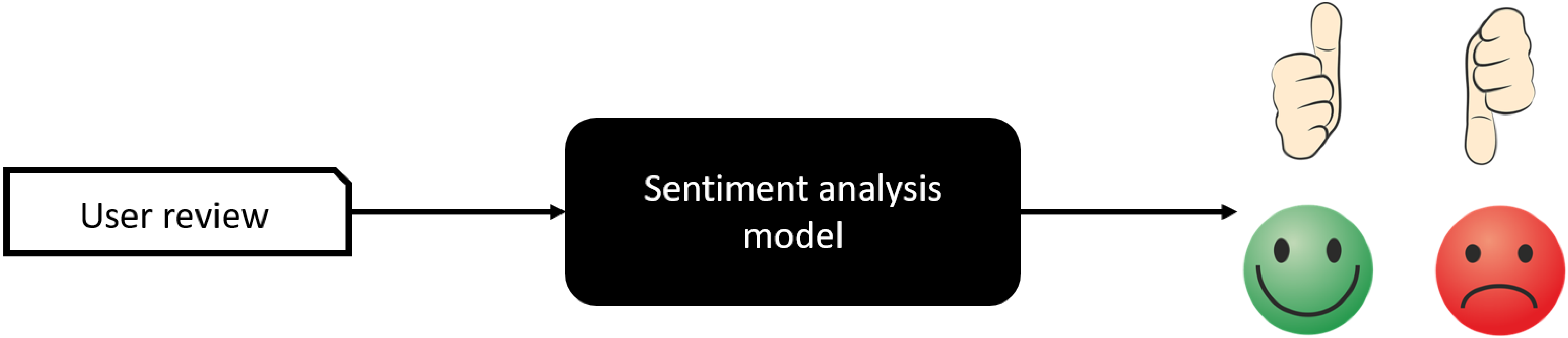
Text and image explainability with LIME

EXPLAINABLE AI IN PYTHON

Fouad Trad
Machine Learning Engineer

Text-based models

- Process and interpret written language
- Example: Sentiment analysis
- Black box models
- `LimeTextExplainer` explains such models
  - Finds how each word impacts prediction



User review

Sentiment analysis model

LIME text explainer

```python
from lime.lime_text import LimeTextExplainer

text_instance =
"This product has great features but a poor design."

def model_predict(instance):
  ...
  return class_probabilities

explainer = LimeTextExplainer()
exp = explainer.explain_instance(
  text_instance,
  model_predict
)

exp.as_pyplot_figure()
```

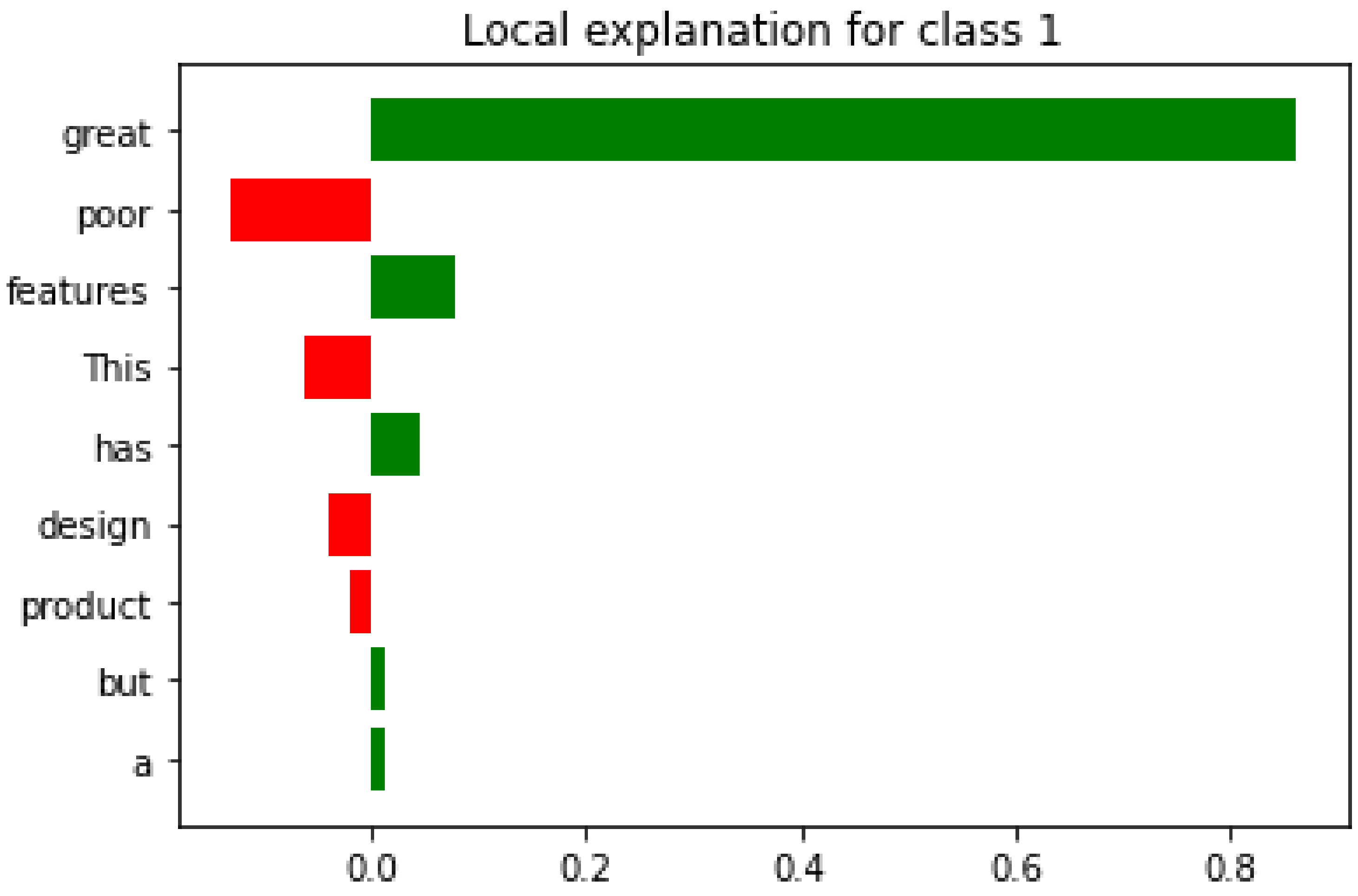
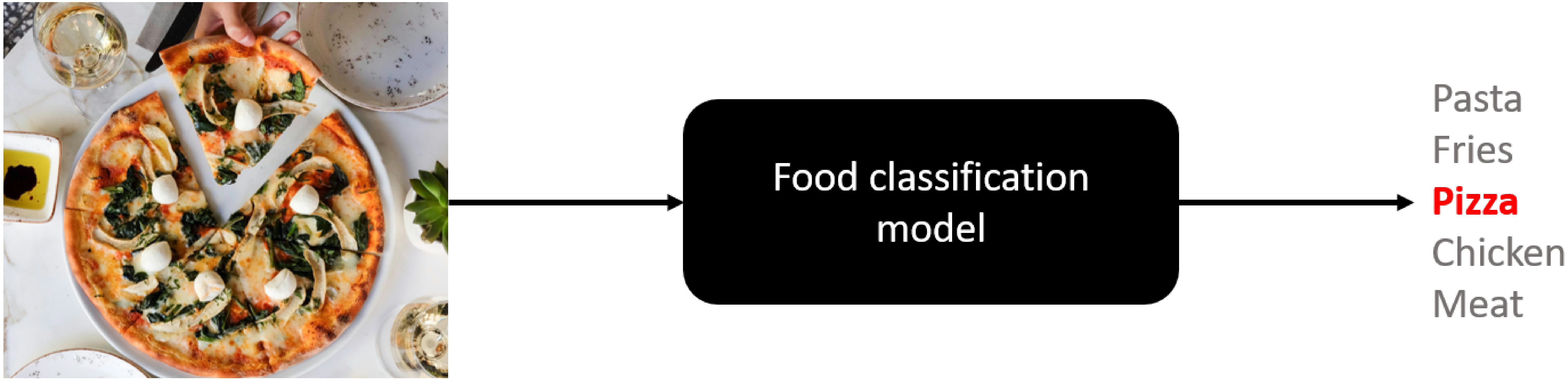


Image-based models

- Highly complex
- Interpret visual data
- Example: Food classification
- `LimeImageExplainer` explains such models
    - Finds which parts of image impact predictions



Food classification model

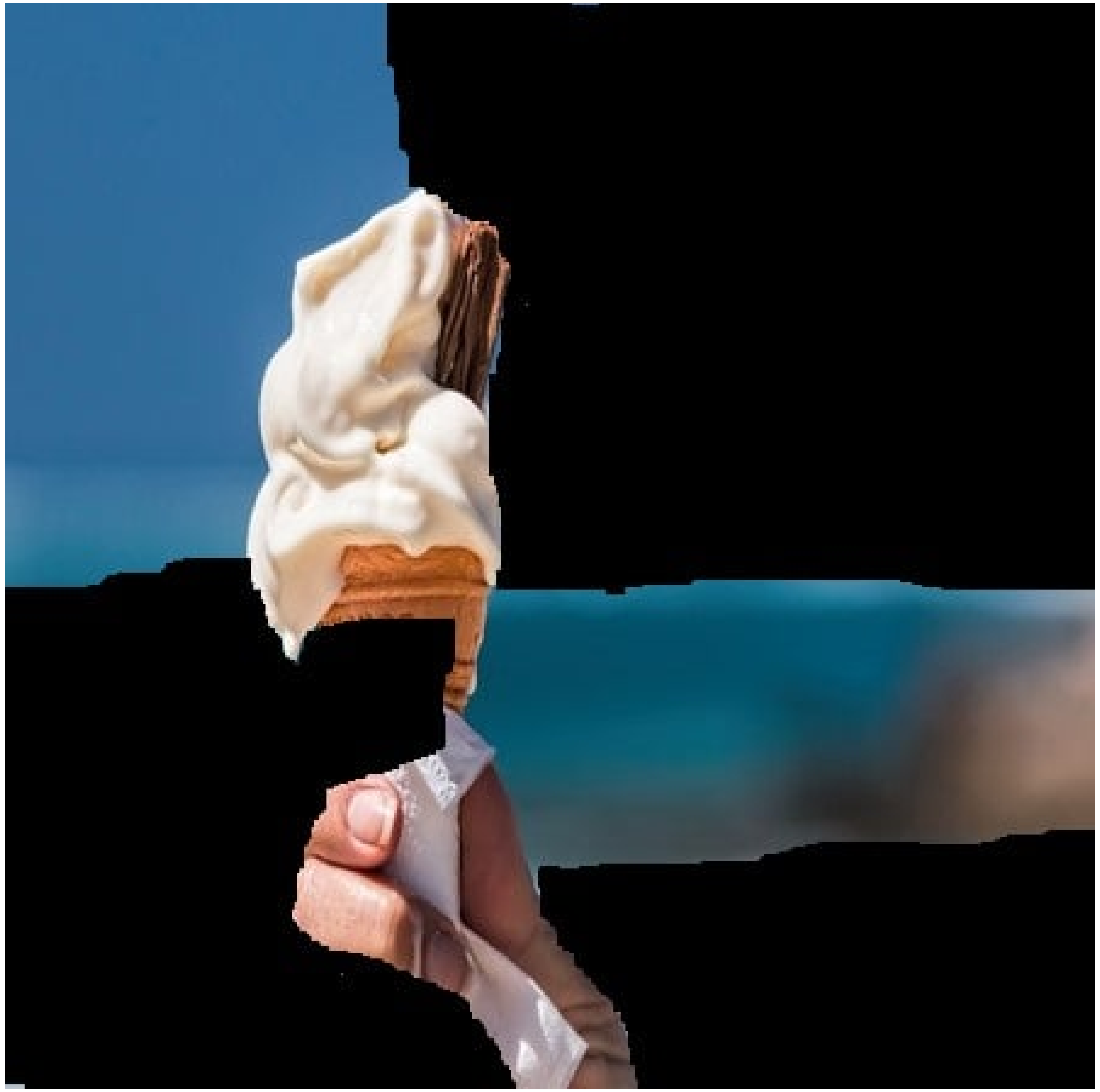
Pasta
Fries
Pizza
Chicken
Meat

LIME image explainer

```python
from lime.lime_image import LimeImageExplainer

explainer = LimeImageExplainer()
explanation = explainer.explain_instance(
  image,
  model_predict,
  num_samples=50
)

temp, _ = explanation.get_image_and_mask(
  explanation.top_labels[0],
  hide_rest=True
)
```

LIME image explainer

```
from lime.lime_image import LimeImageExplainer

explainer = LimeImageExplainer()
explanation = explainer.explain_instance(
  image,
  model_predict,
  num_samples=50
)

temp, _ = explanation.get_image_and_mask(
  explanation.top_labels[0],
  hide_rest=True
)

plt.imshow(temp)
```



Let's practice!

EXPLAINABLE AI IN PYTHON