

Librerie Python per Explainable Artificial Intelligence

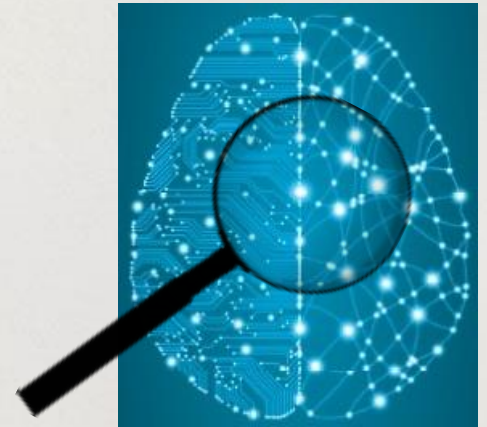
Ph.D. Roberto Marmo

www.robertomarmo.net info@robertomarmo.net

<https://it.linkedin.com/in/robertomarmo/it>

Agenda

1. Intelligenza Artificiale
2. Machine Learning
3. Explainable Artificial Intelligence
4. Librerie Python
 - 4a. LIME
 - 4b. SHAP
 - 4c. ELI5
 - 4d. Explainerdashboard
 - 4e. tf-explain
5. Conclusioni e bibliografia



1. Intelligenza Artificiale (AI)

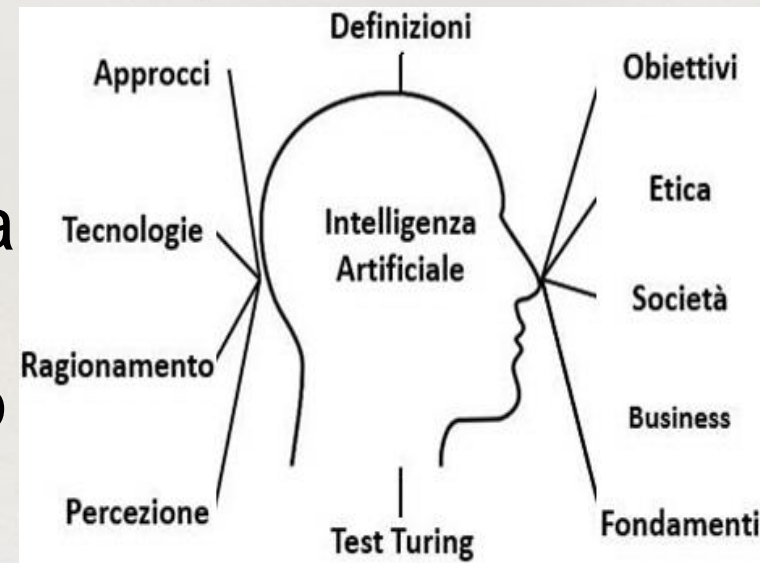
il cervello umano è un risolutore intelligente per alcuni problemi



algoritmi con funzionamento ispirato alla struttura del cervello umano e modo di ragionare



intelligenza artificiale è scienza di fare con computer cose che richiedono intelligenza quando sono fatte da esseri umani



2. Machine Learning (ML)

Il computer può studiare, apprendere in automatico e migliorare come uno studente a scuola:

1. all'inizio non sa fare niente
2. comincia a creare la conoscenza dei dati
3. prova a fare i primi esercizi
4. il maestro spiega dove ha sbagliato
5. corregge i suoi meccanismi interni
6. la prossima volta sbaglia di meno
7. viene promosso e mandato a lavorare



2. Apprendimento supervisionato

Obiettivo: gestione ventilatore con ML

Input: Umidità, Temperatura

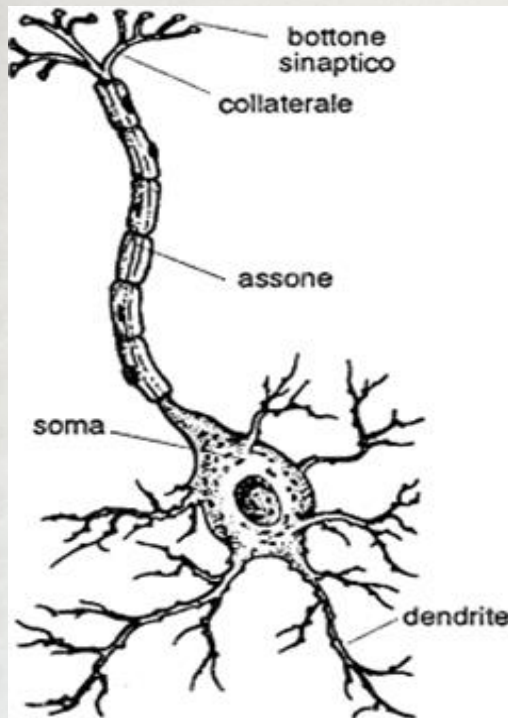
Output: Accendere SI o NO?

1. esperto prepara (input, input, output)
esempi come (2,1,SI) (1,1,NO)
(0,1,SI) (2,5,NO) da sua esperienza
2. ML impara a collegare input e output
da grande quantità di esempi
3. esperto controlla se ha appreso i dati,
fa un test e lo promuove all'uso
4. fornisco input (5, 6) e ML risponde SI

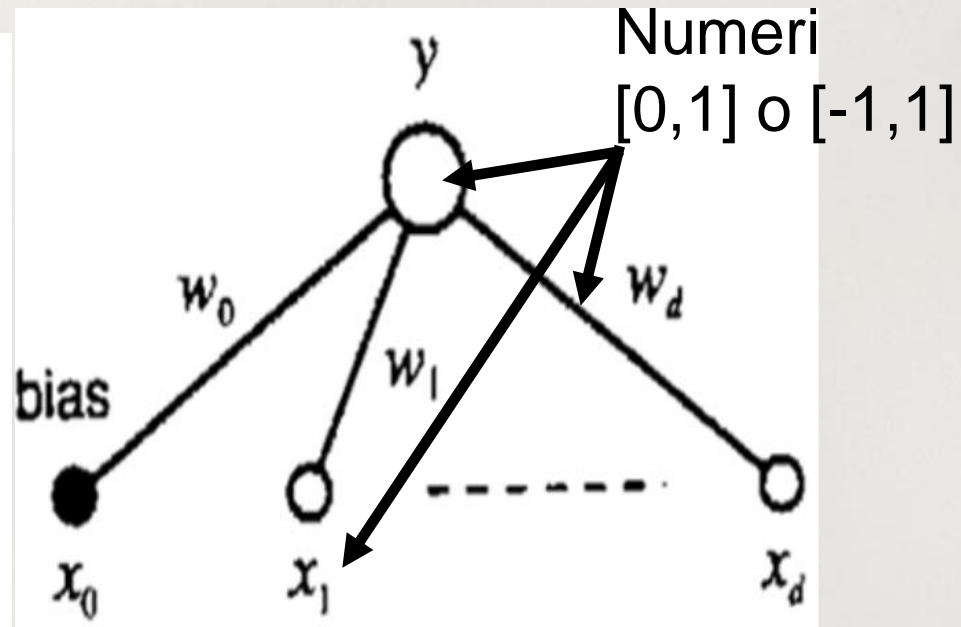
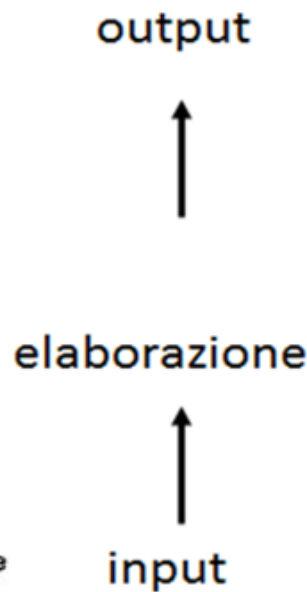


2. Machine Learning

Neural Network ispirato al cervello umano



Neurone biologico



$$y(x) = g\left(\sum_{i=1}^d w_i x_i + w_0 x_0\right)$$

Neurone matematico

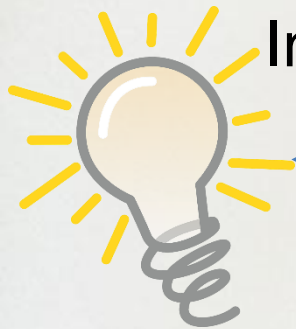
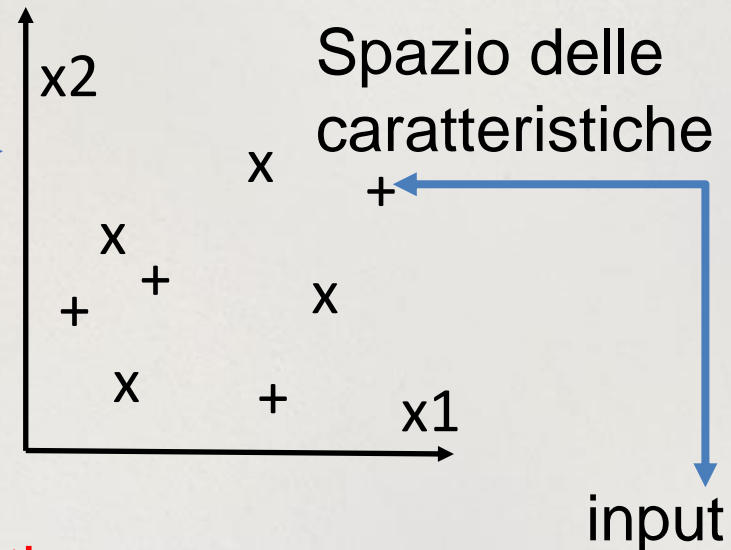
2. Machine Learning



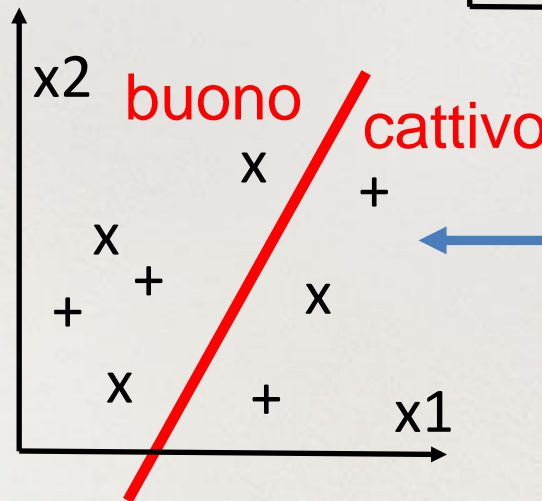
Problema

123456
2354667
32223332
25555667
2222223.4

Numeri descrivono
caratteristiche di oggetto



Interpretare
soluzione



Modello ML
 $\text{out} = f(\text{in}, w)$

Iper-parametri w

XAI Explainable Artificial Intelligence

3.Explainable Artificial Intelligence

Molte decisioni sono critiche

Non c'è 100% di correttezza

Chi risponde in caso di errore?



Serve spiegazione di:

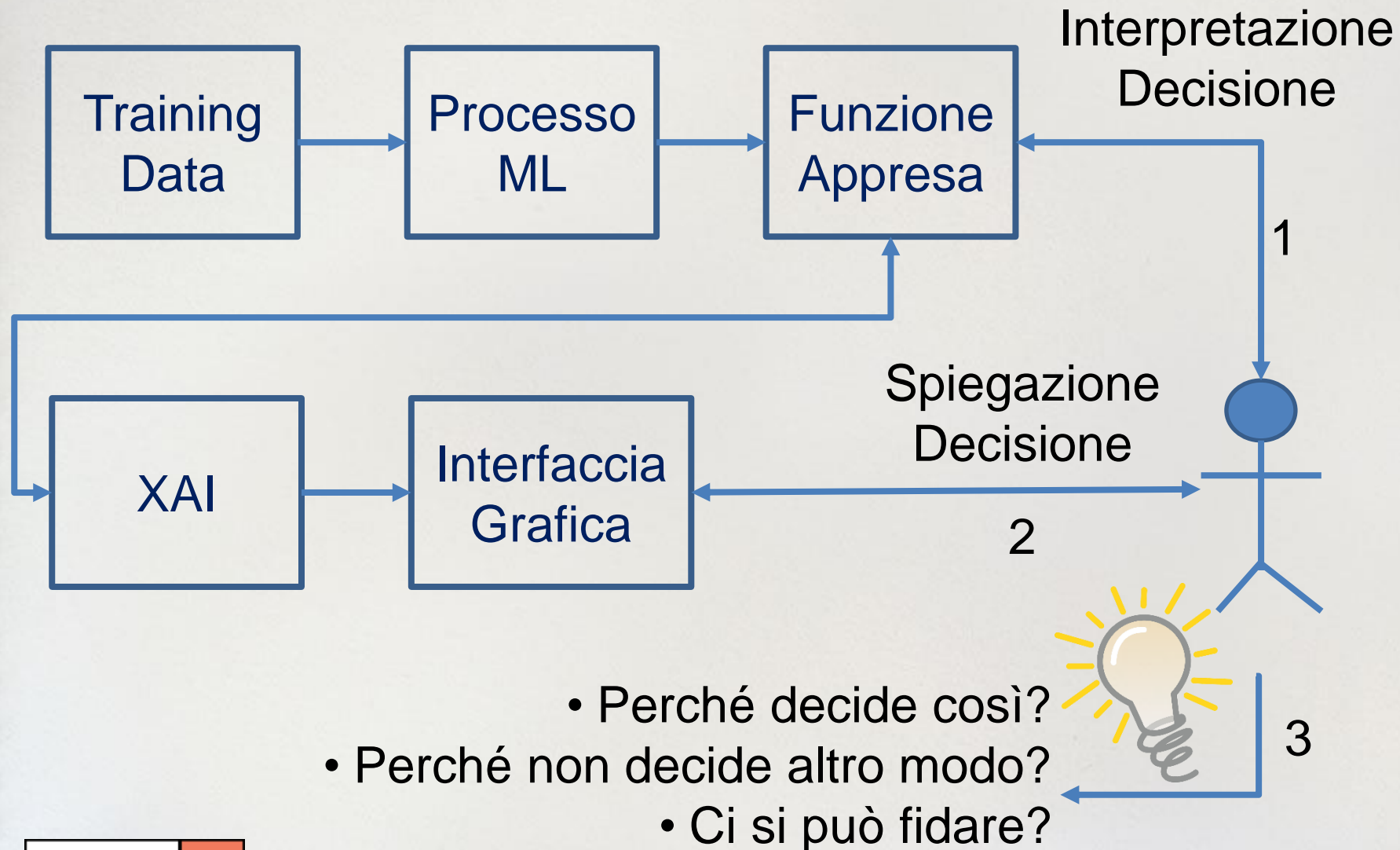
- importanza di ogni input
- importanza dei parametri ottimizzati
- ogni calcolo eseguito
- interpretazione di output



XAI Explainable Artificial Intelligence



3.Explainable Artificial Intelligence



4. Librerie Python

Aggiunta librerie XAI al codice AI di ML



1. nome libreria
2. sito web
3. modalità computazionale
4. quale modello ai e apprendimento può spiegare
5. spiegazione locale o globale
6. limiti di uso
7. esempio di codice

4. Librerie Python, data set

 UCI MACHINE LEARNING · UPDATED 5 YEARS AGO

▲ 2100

New Notebook

Download (26 kB)

Red Wine Quality

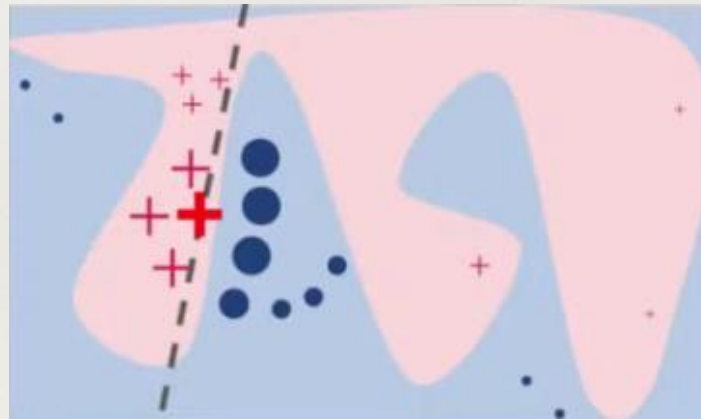
Simple and clean practice dataset for regression or classification modelling



- <https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009>
- Caratteristiche per descrivere un vino rosso:
 1. fixed acidity
 2. volatile acidity
 3. citric acid
 4. residual sugar
 5. chlorides
 6. free sulfur dioxide
 7. total sulfur dioxide
 8. density
 9. ph
 10. sulphates
 11. alcohol
 12. quality, classificare buono/cattivo

4a. LIME

- ✓ Local Interpretable Model-agnostic Explanations
- ✓ <https://github.com/marcotcr/lime>
- ✓ per spiegare una singola predizione fatta da modello ML apprendimento supervisionato



1. con perturbazioni crea punti simili a cosa predetto
2. addestra classificatore lineare su questi punti
3. i pesi del classificatore indicano importanza della caratteristica di input

4a. LIME

```
from sklearn.model_selection import train_test_split

X = wine.drop('quality', axis=1) #dati in input
y = wine['quality'] #valore output da predire come qualità del vino

#creazione training e test set per apprendimento supervisionato
X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.2, random_state=42)

from sklearn.ensemble import RandomForestClassifier
#addestramento di una foresta di alberi di classificazione
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
score = model.score(X_test, y_test)
```

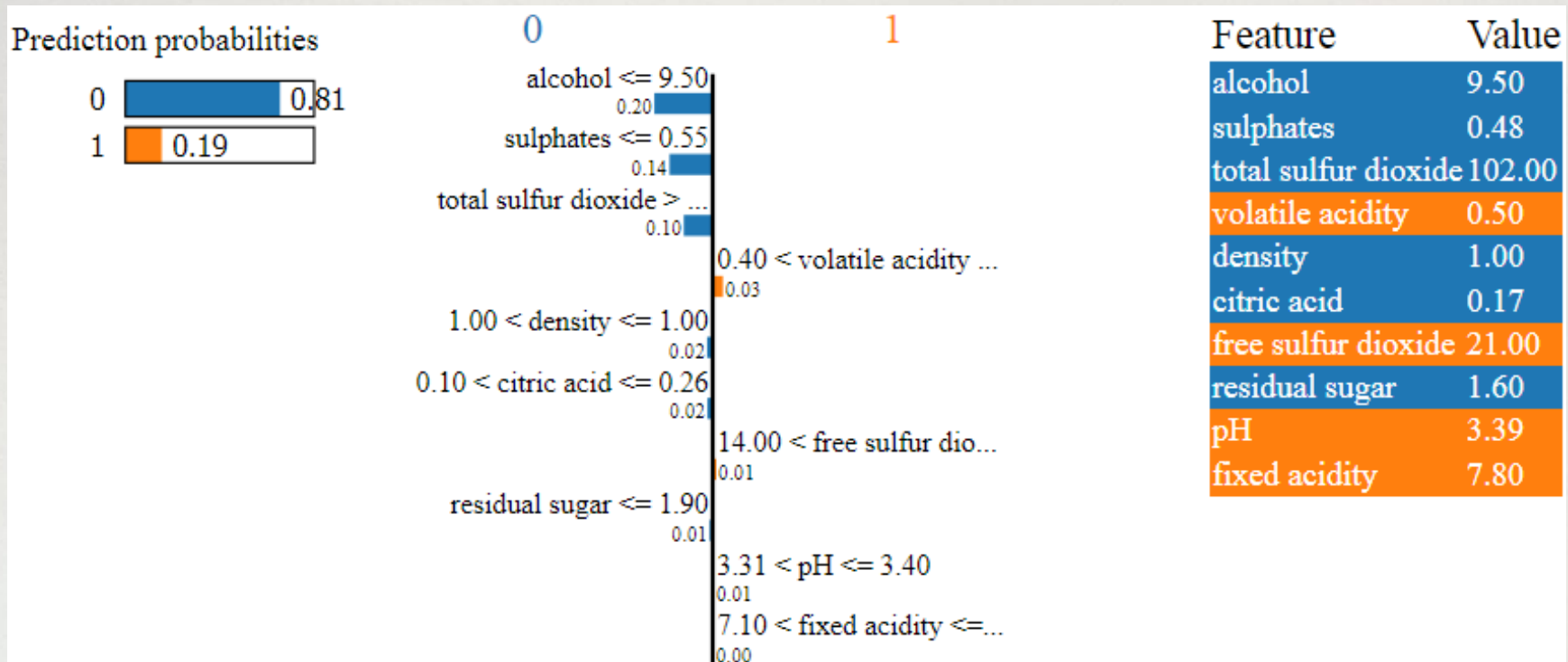
```
#analisi del modello con libreria LIME
import lime
from lime import lime_tabular

#preparazione della spiegazione
explainer = lime_tabular.LimeTabularExplainer(
    training_data=np.array(X_train),
    feature_names=X_train.columns,
    class_names=['0', '1'],
    mode='classification'
)
```

```
#estrazione della spiegazione
exp = explainer.explain_instance(
    data_row=X_test.iloc[1],
    predict_fn=model.predict_proba
)

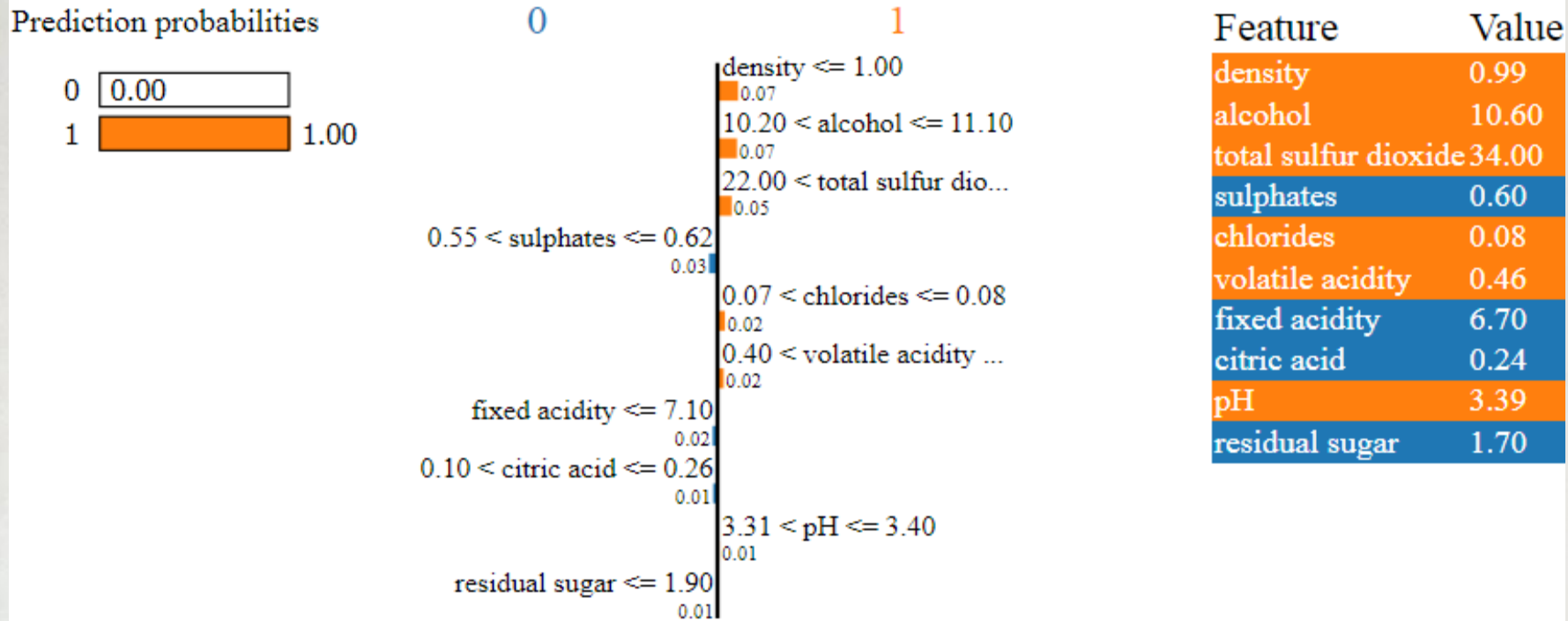
#spiegazione mostrata con interfaccia grafica
exp.show_in_notebook(show_table=True)
```

4a. LIME



alcohol, sulphates, total sulfur dioxide aumentano le possibilità del vino di essere classificato 0, mentre *volatile acidity* fa decrescere

4a. LIME



Vino classificato perfettamente come 1

4b. SHAP

- ✓ SHAP (SHapley Additive exPlanations)
- ✓ <https://github.com/slundberg/shap>
- ✓ spiegazione locale di una predizione o di tutte svolte con ML ad apprendimento supervisionato
- ✓ teoria dei giochi con valore di Shap: contributo di ogni caratteristica nell'ottenere una certa predizione, per ogni caratteristica si calcola la differenza di risultato tra quando c'è o non c'è per ogni combinazione di caratteristiche

4b. SHAP

```
from sklearn.model_selection import train_test_split
wine = pd.read_csv('winequality-red.csv')

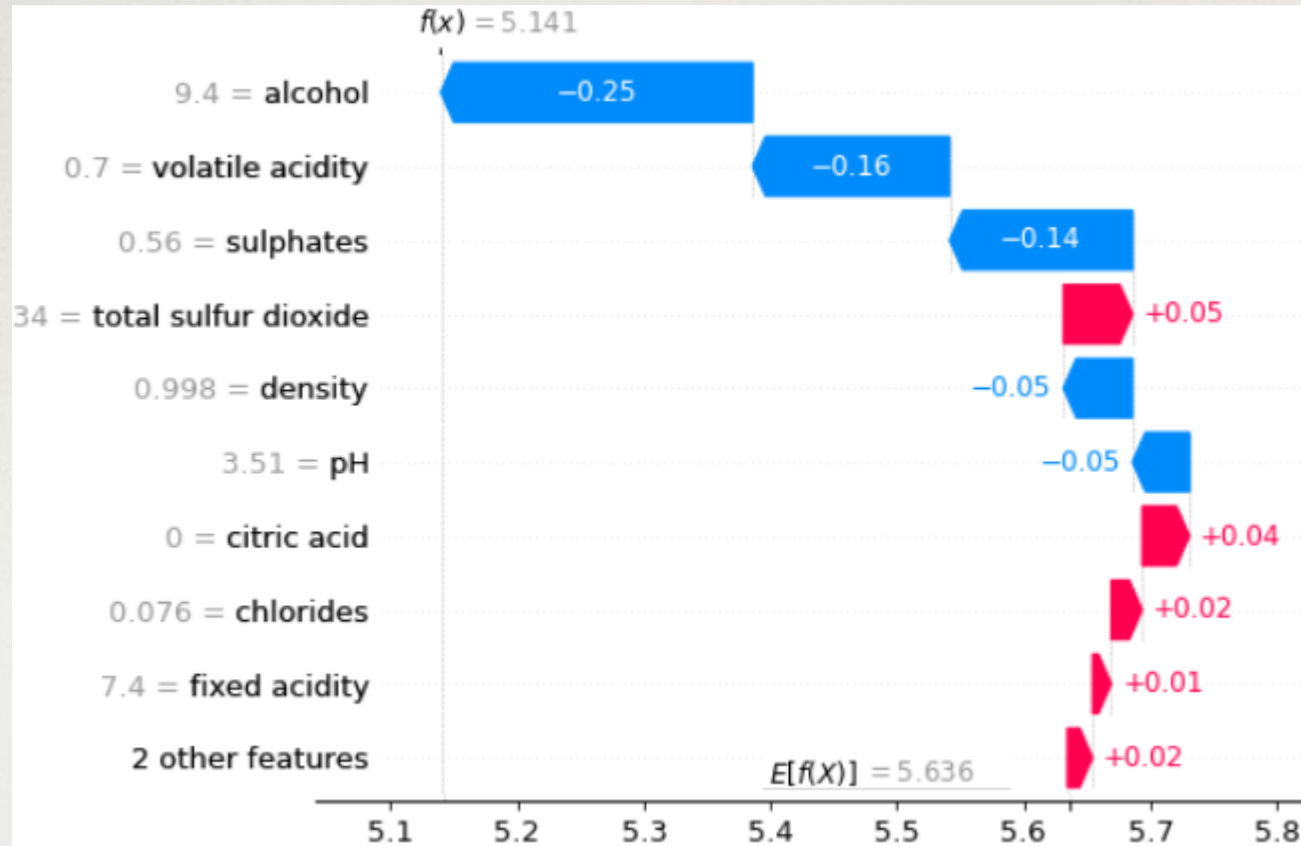
X = wine.drop('quality', axis=1)
y = wine['quality']

import xgboost
# train an XGBoost model
model = xgboost.XGBRegressor().fit(X, y)

import shap
# spiegazione della predizione del modello con SHAP
# sintassi uguale per LightGBM, CatBoost, scikit-learn, transformers, ecc.
explainer = shap.Explainer(model)
shap_values = explainer(X)

# visualizza la spiegazione della predizione per valore input riga 0
shap.plots.waterfall(shap_values[0])
```

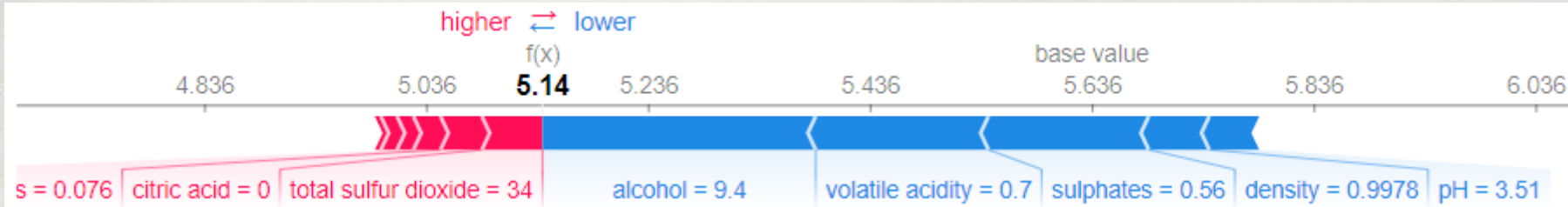
4b. SHAP



Caratteristiche che contribuiscono a spingere output del modello dal valore di base (output medio del modello sui dati di addestramento) all'output del modello. Le rosse spingono la predizione in alto, le blu spingono in basso.

4b. SHAP

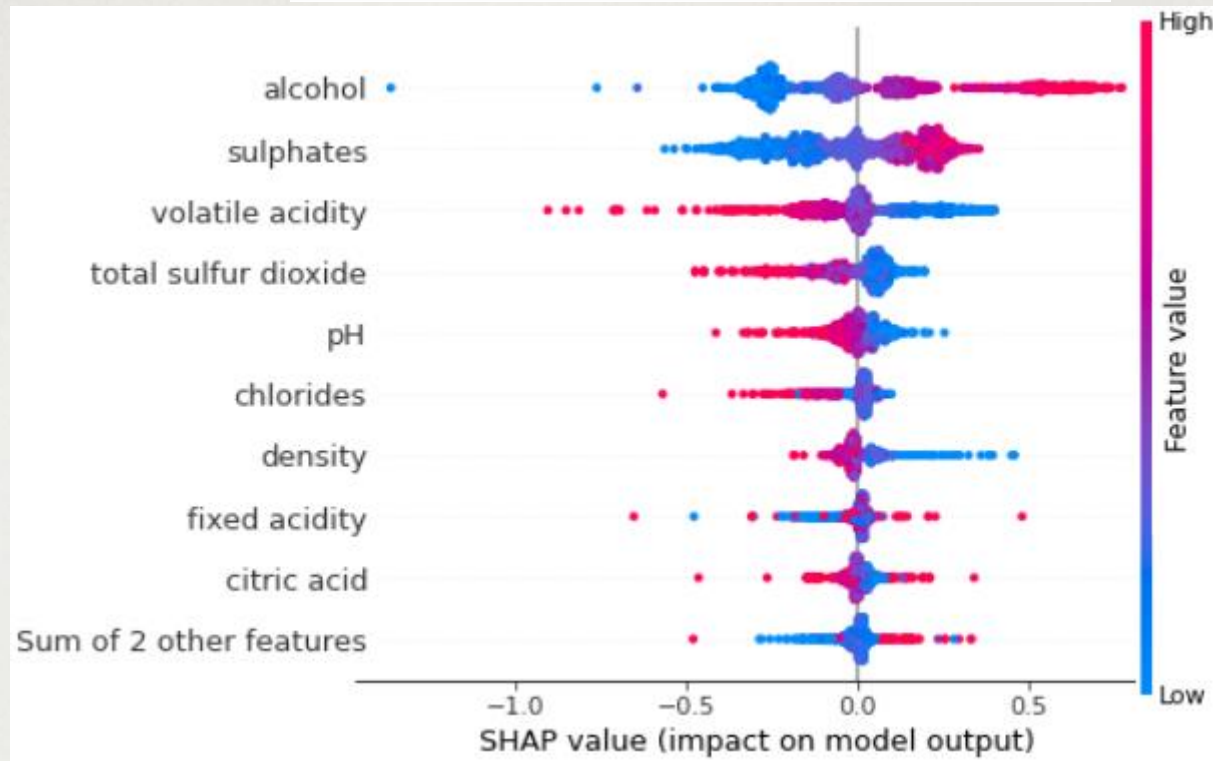
```
# visualizza la prima predizione con force plot  
shap.initjs()  
shap.plots.force(shap_values[0])
```



Altra spiegazione per singola predizione con force plot. Le caratteristiche che contribuiscono a spingere output dal valore di base (output medio del modello) al valore effettivo previsto, le caratteristiche rosse influenzano più in alto, le blu più in basso

4b. SHAP

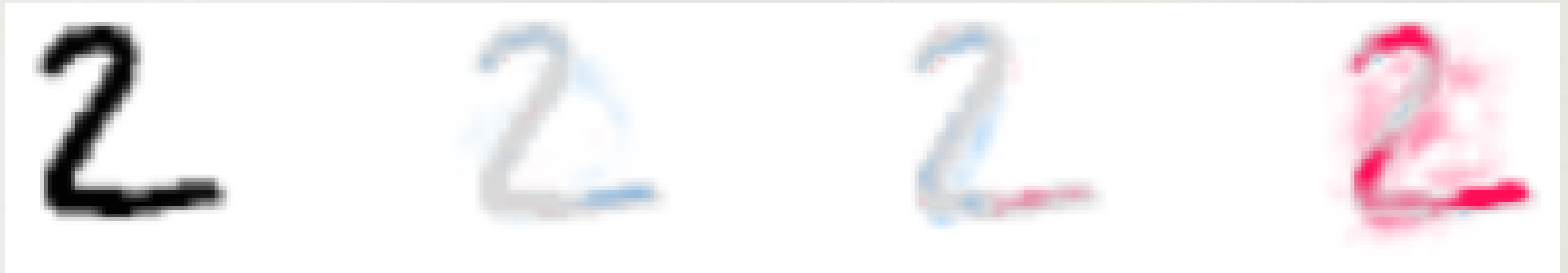
```
# visualizza la prima predizione con force plot  
shap.initjs()  
shap.plots.force(shap_values[0])
```



Impatto delle caratteristiche più importanti per la predizione.
Esempio: un valore alto di alcohol incide molto sul risultato.
Si possono creare altri grafici di analisi delle caratteristiche.

4b. SHAP

```
#DeepShap con DeepExplainer
import shap
# scelta dati di apprendimento per creare lo sfondo dell'immagine
background = x_train[np.random.choice(x_train.shape[0], 100, replace=False)]
# spiega la predizione del modello per 4 immagini
e = shap.DeepExplainer(model, background)
# oppure con passaggio diretto del tensore
# e = shap.DeepExplainer((model.layers[0].input, model.layers[-1].output), background)
shap_values = e.shap_values(x_test[1:5])
# plot delle attribuzioni di caratteristiche
shap.image_plot(shap_values, -x_test[1:5])
```



Colore nero: oggetto da classificare


Colore grigio: proposte di riconoscimento

Pixel blu: quali pixel mancano per classificare 2 in 0

Pixel rossi: quali pixel confermano la classificazione 2

4c. ELI5

- ✓ <https://pypi.org/project/eli5/>
- ✓ permutando (modificando la posizione) i valori di ciascuna caratteristica, una per una, si può verificare come cambiano le predizioni del modello



182	155	...	20
175	147	...	10
...
156	142	...	8
153	130	...	24

4c. ELI5

```
from eli5 import show_weights
from eli5.sklearn import PermutationImportance
#Permutation Importance
perm = PermutationImportance(xgb_clf, scoring = 'accuracy', random_state=101).fit(X_test, y_test)
show_weights(perm, feature names = list(X_test.columns))
show_weights(xgb_clf, importance_type = 'gain')
```

Weight	Feature	Weight	Feature
0.1000 ± 0.0515	alcohol	0.2404	alcohol
0.0400 ± 0.0266	sulphates	0.1154	total sulfur dioxide
0.0256 ± 0.0343	volatile acidity	0.1084	sulphates
0.0175 ± 0.0161	total sulfur dioxide	0.0886	volatile acidity
0.0131 ± 0.0047	residual sugar	0.0754	fixed acidity
0.0075 ± 0.0287	fixed acidity	0.0693	free sulfur dioxide
0.0069 ± 0.0139	citric acid	0.0646	chlorides
0.0050 ± 0.0129	density	0.0632	citric acid
-0.0006 ± 0.0248	pH	0.0621	pH
-0.0075 ± 0.0094	chlorides	0.0580	density
-0.0088 ± 0.0222	free sulfur dioxide	0.0547	residual sugar

mostra contributo di ciascuna caratteristica nella predizione dell'output, quando si permuta alcohol la accuratezza della predizione cambia del valore indicato con certa imprecisione

4d. ExplainerDashboard

- ✓ <https://explainerdashboard.readthedocs.io/en/latest/>
- ✓ <https://github.com/Devashree21/ExplainerDashboard>
- ✓ Dashboard composto da vari grafici per esplorare importanza delle caratteristiche con vari approcci grafici e statistici
- ✓ Posso fornire i grafici a un decisore non esperto di AI per aiutarlo a scegliere

4d. ExplainerDashboard

```
#addestramento del modello
model = RandomForestRegressor(n_estimators=50, max_depth=5)
model.fit(X_train, y_train.values.ravel())
explainer = RegressionExplainer(model, X_test, y_test)
#avvio di creazione Dashboard
db = ExplainerDashboard(explainer, title="Wine Prediction", whatif=False)
#esecuzione di app nel browser su porta 3050
db.run(port=3050)
```

Wine Prediction

[Download ▾](#)[Feature Importances](#)[Regression Stats](#)[Individual Predictions](#)[Feature Dependence](#)[Feature Interactions](#)[Decision Trees](#)

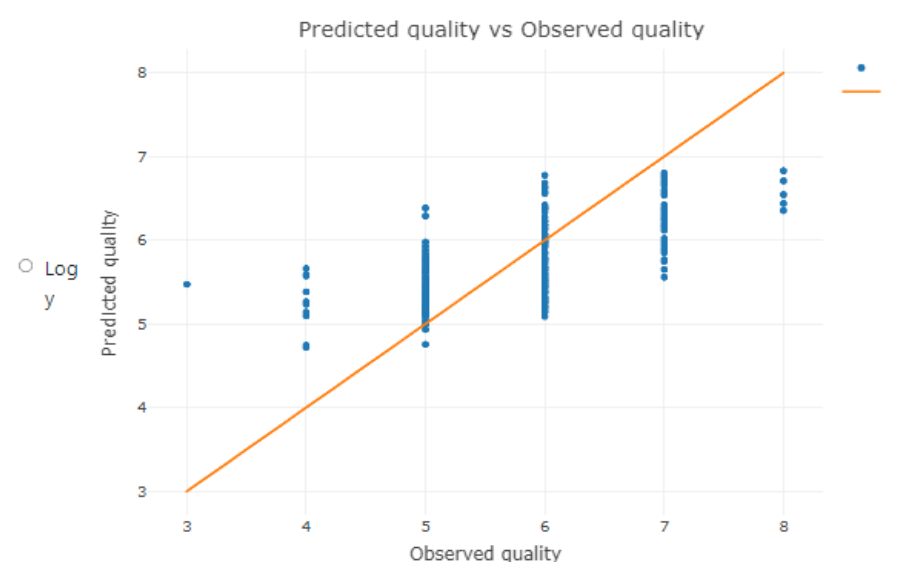
Model Summary

Quantitative metrics for model performance

metric	Score
mean-squared-error	0.366
root-mean-squared-error	0.605
mean-absolute-error	0.489
mean-absolute-percentage-error	0.087
R-squared	0.44

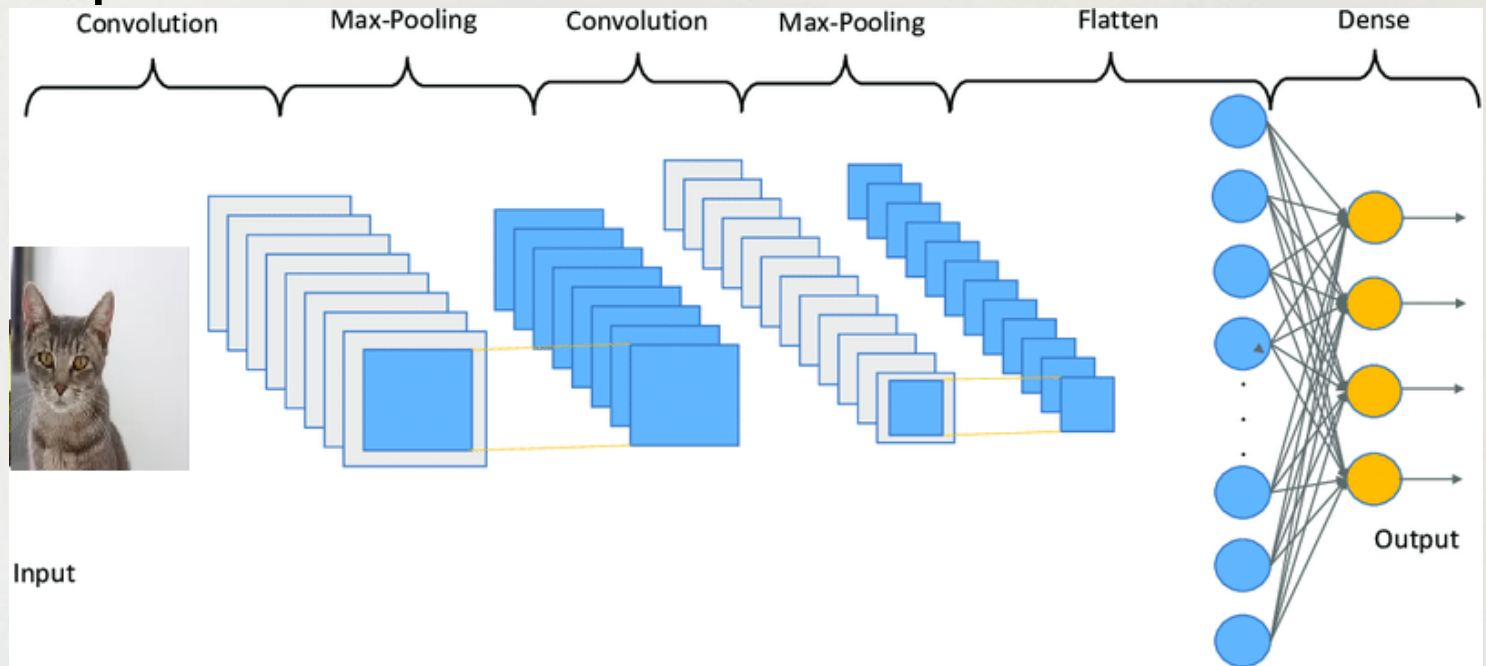
Predicted vs Actual

How close is the predicted value to the observed?



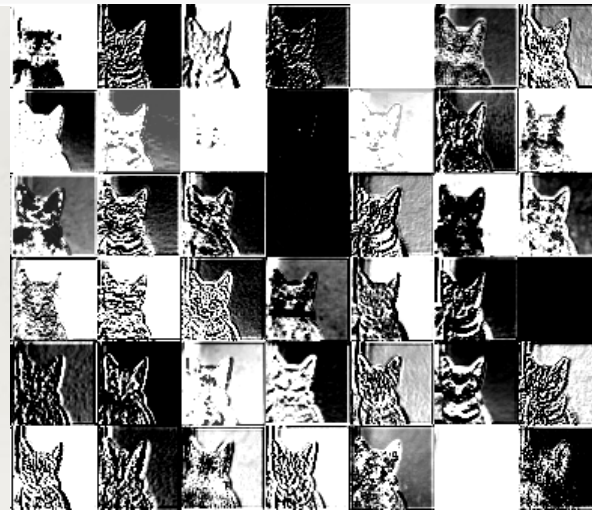
4e. tf-explain

- ✓ <https://github.com/sicara/tf-explain> <https://tf-explain.readthedocs.io/en/latest/>
- ✓ Dashboard composto da vari grafici per esplorare uso di TensorFlow nel Deep Learning
- ✓ Output nei vari livelli di attivazione del network



4e. tf-explain

```
import tensorflow as tf
model = tf.keras.applications.vgg16.VGG16(weights="imagenet", include_top=True)
img = tf.keras.preprocessing.image.load_img("cat.jpg", target_size=(224, 224))
img = tf.keras.preprocessing.image.img_to_array(img)
data = ([img], None)
# avvio di explainer
explainer = GradCAM()
grid = explainer.explain(data, model, class_index=281)
explainer.save(grid, ".", "grad_cam.png")
```



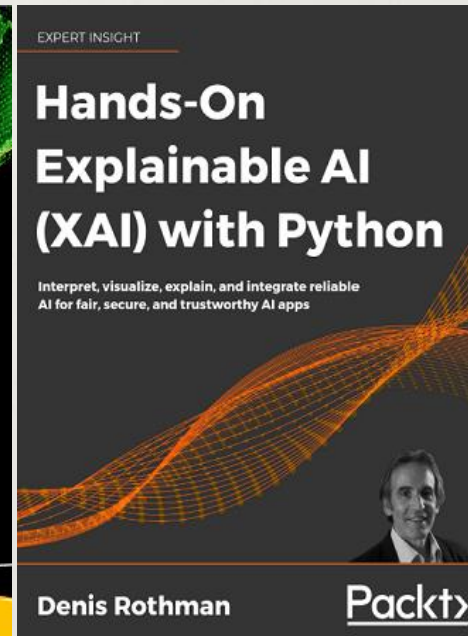
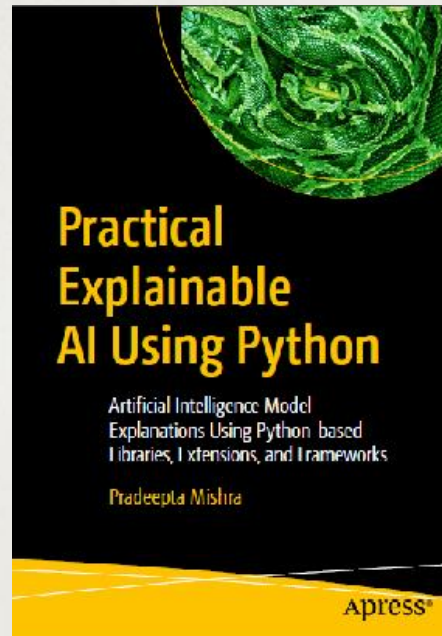
Da sinistra: immagine originale, immagini su livelli intermedi, parti di immagine influenti su output esaminando le mappe di attivazione dei neuroni

5. Conclusioni e bibliografia

- ✓ Intelligenza Artificiale ampiamente diffusa per calcoli avanzati
- ✓ Machine Learning molto usato per imparare dai dati, ma difficile capire come «ragiona»
- ✓ Si può capire qualcosa con XAI
- ✓ Provare diversi metodi di spiegazione
- ✓ Saper interpretare la spiegazione
- ✓ Molte domande ancora da risolvere



5. Conclusioni e bibliografia



<https://www.kdnuggets.com/2020/01/explaining-black-box-models-ensemble-deep-learning-lime-shap.html>

<https://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf>

<https://www.kaggle.com/learn/machine-learning-explainability>

<https://xai-project.eu/>

<http://www.di.uniba.it/~swap/xai-it/> Italian Workshop XAI

Fine!

info@robertomarmo.net



<https://bit.ly/3NPGuT7>

colab

PW: MarmoFirenze22