Interpretable Active Learning

Michał Sokół and Arkadiusz Pajor

Agenda

1. What is Active Learning?
2. What is Interpretable Machine Learning?
3. What is Interpretable Active Learning?
4. LIME demo

Active Learning

Definition

Active learning is the name used for the process of **prioritising** the data which needs to be labelled in order to have the **highest impact to training** a supervised model.

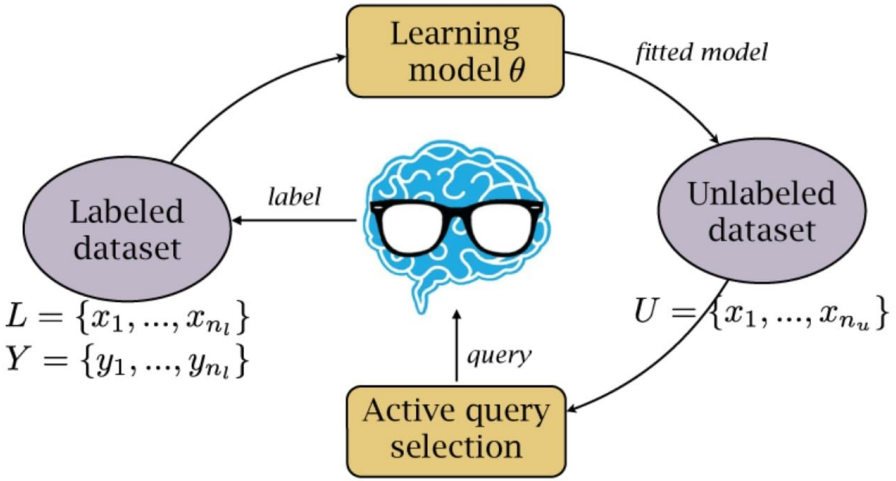
Active learning can be used in situations where the **amount of data is too large** to be labelled and some priority needs to be made to label the data in a smart way [1].

Motivation

1. Most of the data is **unlabeled**, but very often we are interested in **supervised** tasks.
2. Labeling process very often requires **domain-expert knowledge** and it takes long time, so it is **expensive**.
3. What if we **automate** this labeling process?

Popular solution

1. Label **manually** small sample of a data.
2. Train an **imperfect** model on it.
3. Once the model is trained, use it to predict the class of selected remaining unlabelled data points.
4. Use some score to determine a quality of the prediction.
5. Once the best approach has been chosen this process can be **iteratively** repeated (get back to point 2, but this time use manually labelled data and data already labelled by the model).



Selection process

It is done by **prioritising** some data points over the others.

There are several approaches to assign a priority score to each data point:

- **least confidence** - it takes the highest probability for each data point's prediction and sorts them from smaller to larger

$$s_{LC} = \underset{x}{\operatorname{argmax}} \left(1 - P(\hat{y}|x)\right) \qquad \hat{y} = \underset{y}{\operatorname{argmax}} P(y|x)$$

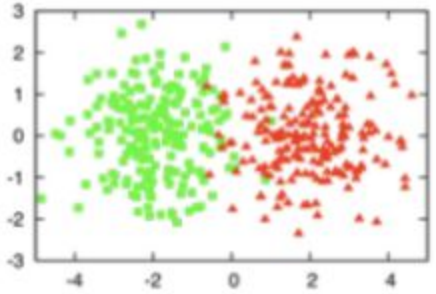
- **margin sampling** - takes into account the difference between two highest probabilities

$$s_{MS} = \underset{x}{\operatorname{argmin}} \left(P(\hat{y}_{max}|x) - P(\hat{y}_{max-1}|x)\right)$$

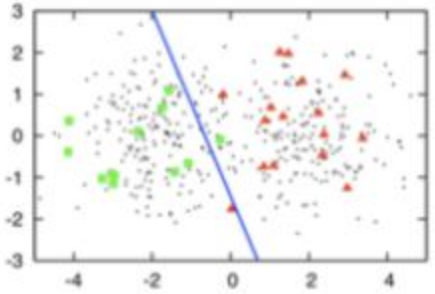
- **entropy** - it also takes the probabilities of the classes, but calculates entropy, prioritising the data points from the highest to the lowest one

$$s_E = \underset{x}{\operatorname{argmax}} \left(-\sum_i P(\hat{y}_i|x) \log P(\hat{y}_i|x) \right)$$

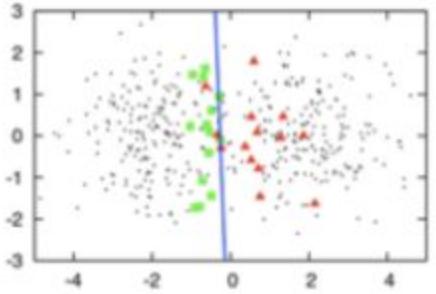
Example



400 instances sampled
from 2 class Gaussians



random sampling
30 labeled instances
(accuracy=0.7)



active learning
30 labeled instances
(accuracy=0.9)

Interpretable Machine Learning

Definition of interpretability

Interpretability is the degree to which a human can **understand the cause of a decision**. [2]

Interpretability is the degree to which a human can **consistently predict the model's result**. [3]

So interpretability is hard to define and measure precisely.

Definition of Interpretable Machine Learning

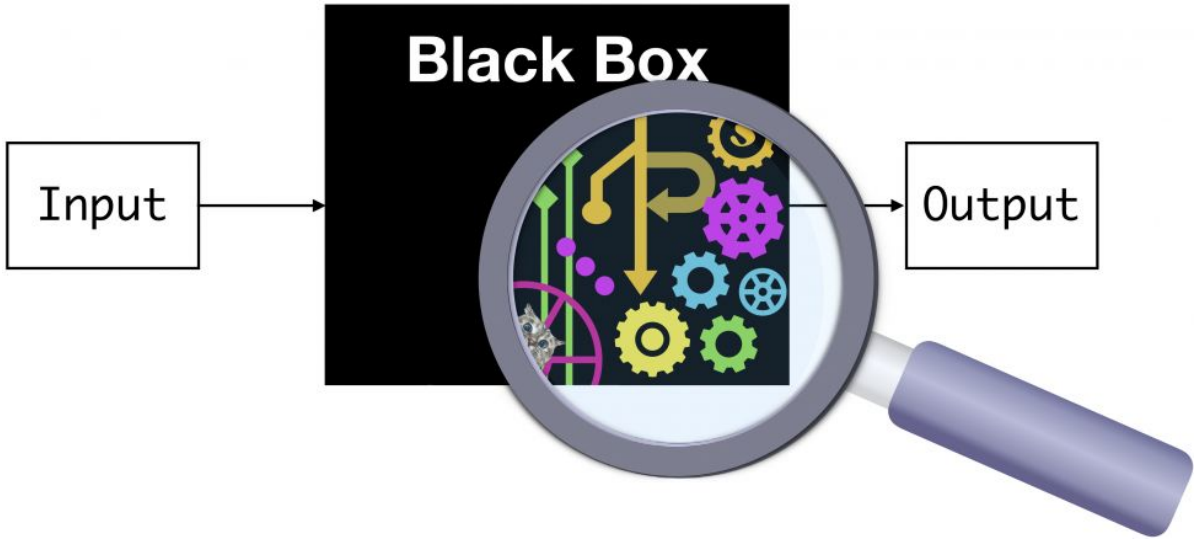
Interpretable machine learning is **extraction** of relevant knowledge from a machine-learning model concerning **relationships** either contained in data or learned by the model. [4]

Motivation

1. More and more systems and applications base on the decisions made by ML models, but many of those are so called **black-boxes**, which predictions are hard to explain and/or interpret.
2. However **interpretability is a crux**, especially when making high stakes decisions (e.g. medicine, jurisdiction).
3. The higher explainability / interpretability the more advantages:
    a. easier debugging and bias detection (e.g. we can discover that our model is racist)
    b. increased social acceptance
    c. safety measures (e.g. we know how autonomous car “sees” bicycle)
4. But there are also disadvantages:
    a. systems may be easier to be manipulated
    b. it is hard to commercialize interpretable models

Solution

1. Use interpretable models (e.g. Linear/Logistic Regression, GLM, GAM, Decision Tree, Decision Rules, Rule Fit, NB, KNN) or...
2. ... try to explain black-box models.



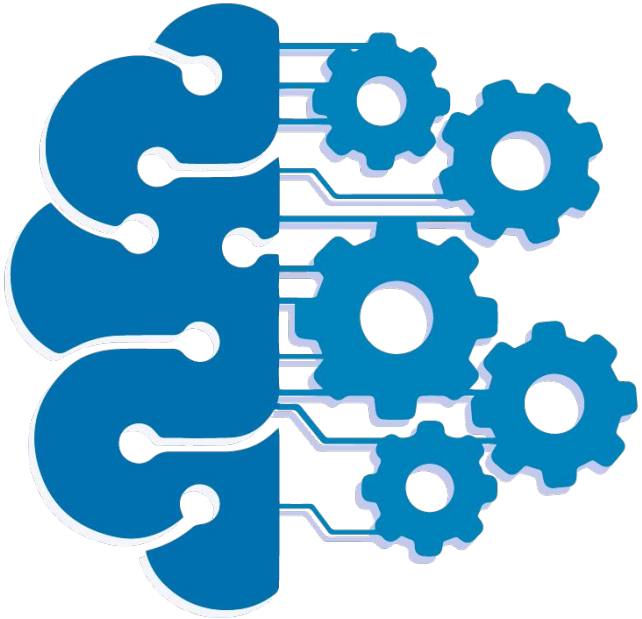
Examples of explanation methods [5]

1. Global Model-Agnostic methods:
    a. **Partial Dependence Plot** - shows the marginal effect one or two features have on the predicted outcome
    b. **Accumulated Local Effects Plot** - shows how features influence the prediction on average
    c. **Feature Interaction**
    d. **Functional Decomposition** - high-dimensional function is expressed as a sum of individual feature effects and interaction effects
    e. **Permutation Feature Importance** - measures the increase in the prediction error of the model after we permuted the feature's values
    f. **Global Surrogate Model** - it is another interpretable model trained to approximate the predictions of a black box model
2. Local Model-Agnostic Methods:
    a. **Individual Conditional Expectation (ICE)** - measures how the instance's prediction changes when a feature changes
    b. **Local Surrogate (LIME)** - interpretable model that is used to explain individual prediction of black-box model
    c. **Counterfactual Explanations**
    d. **Shapley Values** - from game theory, tries to fairly distribute a prediction (payout) among the predictors (players)

Interpretable Active Learning

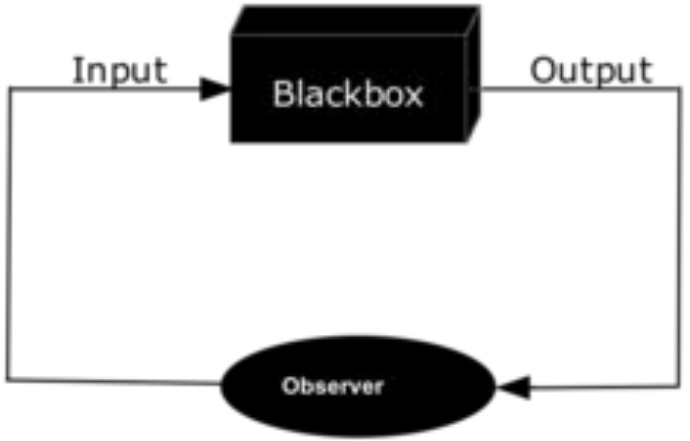
Motivation

We want to know **why** a specific sample or group of samples were queried by active learning strategy.



Local Surrogate (LIME, Local Intepretable Model-Agnostic Explanations)

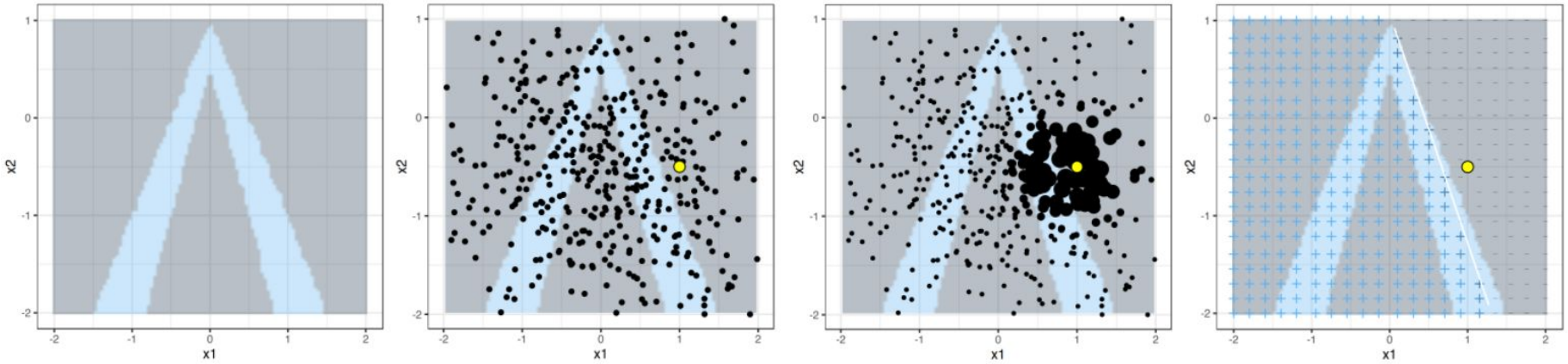
Local surrogate models are interpretable models that are used to **approximate and explain** individual predictions of black box machine learning models only basing on input and output to it.



$$\text{explanation}(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

We choose an interpretable model **g** that minimizes loss **L** (e.g. mean squared error), which measures how close the explanation is to the prediction of the original model **f**, while the model complexity **Ω(g)** is kept low (e.g. prefer fewer features). π_x is a proximity measure (sometimes called as similarity kernel) that defines the neighborhood around instance **x** that we consider for the explanation.

LIME for Tabular Data

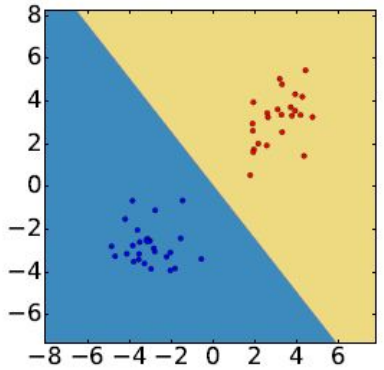


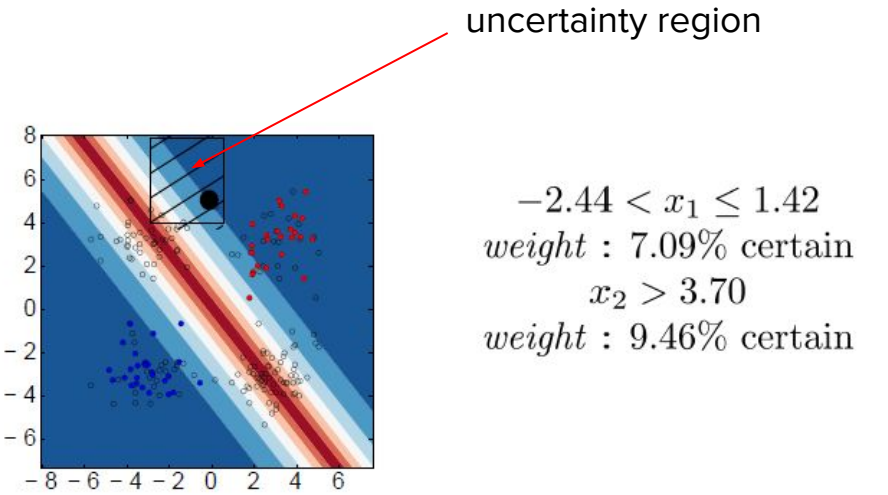
LIME for Images

Example of image classification. The middle picture shows regions which are important to classify the image as a bagel. The rightmost picture shows regions which are important to classify the image as a strawberry. Green colour increases the probability, red color decreases it.

LIME and Active Learning

In [6] it is proposed to use LIME in Active Learning process to **explain query uncertainty**.





Uncertainty Bias

It is used to discover **disproportions** in data or **skewed distribution**, which are crucial for active learner when querying.

In order to understand both what and how an active learning method is learning and whether there is skew in the uncertainty regions targeted to be labeled, **uncertainty bias** of a subgroup is quantified. The goal is to identify subgroups that differ from the majority in terms of their uncertainty in the classifier.

$$1 - \frac{Pr(U = +|x \in r)}{Pr(U = +|x \in R \setminus r)}$$

- **x** - instance vector
- **r** - considered uncertainty region
- **R** - set of all uncertainty regions
- **U** - uncertainty labels (+ = certain; - = uncertain) - any point with certainty greater than or equal to some metric (e.g. median) over the pool is considered as certain (U = +)

Batch of queries

In practice, performing queries one-by-one can be highly inefficient:

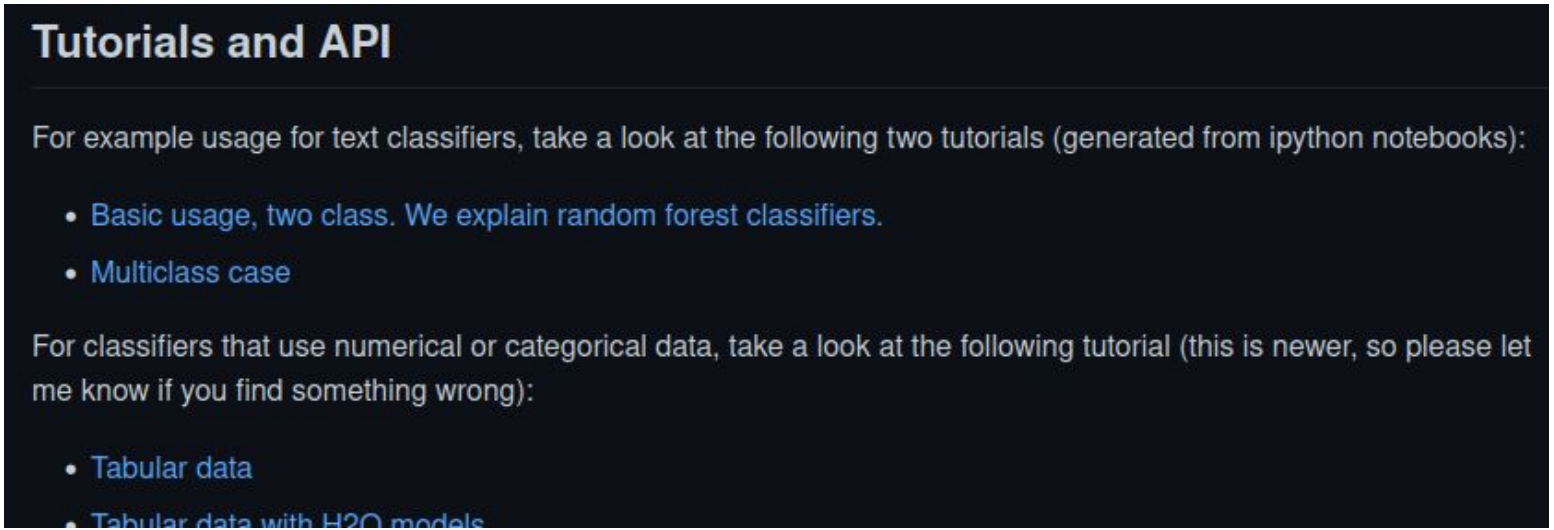
- single recommendations provide little indication of what long term patterns will be explored,
- it may be harder to form hypotheses before exploring problem sub-spaces...
- ...which may lead to retrospective justifications or inefficiencies in experimental design.

In order to create interpretable batches of queries, we consider each uncertainty region as a candidate for batch selection within that region instead of from the entire pool. Every instance inside the batch has the same uncertainty and explanation.

LIME demo

Code

Official repo with tutorials: https://github.com/marcotcr/lime



Tutorials and API

For example usage for text classifiers, take a look at the following two tutorials (generated from ipython notebooks):

- Basic usage, two class. We explain random forest classifiers.
- Multiclass case

For classifiers that use numerical or categorical data, take a look at the following tutorial (this is newer, so please let me know if you find something wrong):

- Tabular data
- Tabular data with H2O models

Our repo with working tutorials: https://github.com/Mlokos/active-learning-lime

lime-tutorial-mnist

Playing around

In [61]:
```
def explain_detailed(X_test, y_test, simple_rf_pipeline, index):
    # predicted number
    pipe_pred_test = simple_rf_pipeline.predict(X_test)
    print('Predicted number: {}'.format(pipe_pred_test[index]))

    # prepare explain
    explainer = lime_image.LimeImageExplainer(verbose = False)
    segmenter = SegmentationAlgorithm('quickshift', kernel_size=1, max_dist=200, ratio=0.2)

    # explain
    explanation = explainer.explain_instance(X_test[0],
                                             classifier_fn = simple_rf_pipeline.predict_proba,
                                             top_labels=10, hide_color=0, num_samples=1000, segmentation_fn=segmenter
    # show graph
    fig, m_axs = plt.subplots(2,5, figsize = (12,6))
    for i, c_ax in enumerate(m_axs.flatten()):
        temp, mask = explanation.get_image_and_mask(i, positive_only=True, num_features=1000, hide_rest=False, min_we
        c_ax.imshow(label2rgb(mask,X_test[index], bg_label = 0), interpolation = 'nearest')
        c_ax.set_title('Positive for {}\nActual {}'.format(i, y_test.iloc[index]))
        c_ax.axis('off')
```

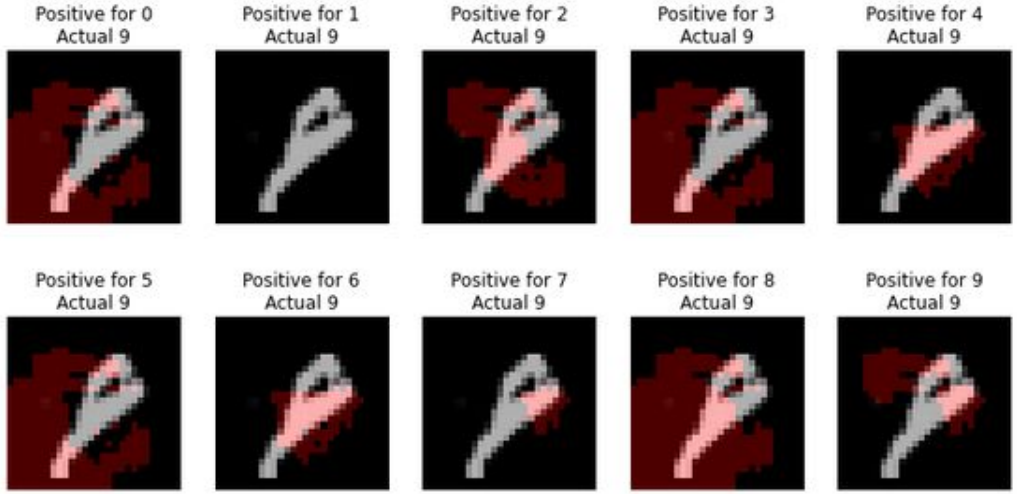
In [62]:
```
def find_wrongly_assumed_numbers(X_test, y_test, simple_rf_pipeline):
    pipe_pred_test = simple_rf_pipeline.predict(X_test)
    wrong_idx = np.random.choice(np.where(pipe_pred_test!=y_test)[0])
    print('Using #{} where the label was {} and the pipeline predicted {}'.format(wrong_idx, y_test.iloc[wrong_idx],
```

lime-tutorial-mnist

In [68]:
```
explain_detailed(X_test, y_test, simple_rf_pipeline, 4455)
```

```
Predicted number: 8
```

100% 1000/1000 [00:01<00:00, 736.75it/s]



In [66]:
```
find_wrongly_assumed_numbers(X_test, y_test, simple_rf_pipeline)
```

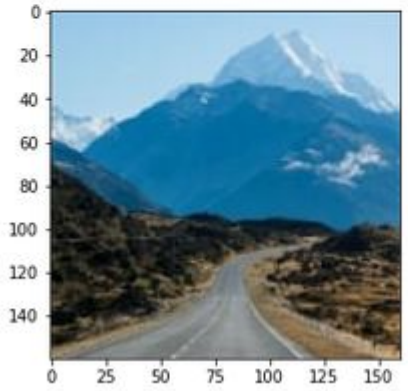
```
Using #4455 where the label was 9 and the pipeline predicted 8
```

lime-tutorial-image-basic

In [38]:
```
def get_image(path):
    with open(os.path.abspath(path), 'rb') as f:
        with Image.open(f) as img:
            return img.convert('RGB')

# img = get_image('./data/dogs.png')
img = get_image('./data/mountain.jpeg')
plt.imshow(img)
```

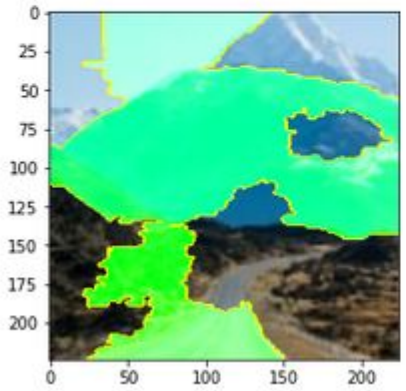
Out[38]:
```
<matplotlib.image.AxesImage at 0x7f713a6ad160>
```



lime-tutorial-image-basic

In [50]:
```
temp, mask = explanation.get_image_and_mask(explanation.top_labels[0], positive_only=False, num_features=10, hide_res
img_boundry2 = mark_boundaries(temp/255.0, mask)
plt.imshow(img_boundry2)
```

Out[50]: <matplotlib.image.AxesImage at 0x7f713a711b20>



References

1. https://towardsdatascience.com/active-learning-in-machine-learning-525e61be16e5
2. Miller, Tim. “Explanation in artificial intelligence: Insights from the social sciences.” arXiv Preprint arXiv:1706.07269. (2017)
3. Kim, Been, Rajiv Khanna, and Oluwasanmi O. Koyejo. “Examples are not enough, learn to criticize! Criticism for interpretability.” Advances in Neural Information Processing Systems (2016)
4. Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. Definitions, methods, and applications in interpretable machine learning. Proceedings of the National Academy of Sciences, 116(44), 22071-22080. (2019).
5. https://christophm.github.io/interpretable-ml-book/
6. Richard L. Phillips, Kyu Hyun Chang, Sorelle A. Fiedler. “Interpretable Active Learning”, In Proceedings of Machine Learning Research 81:1-13 (2018).