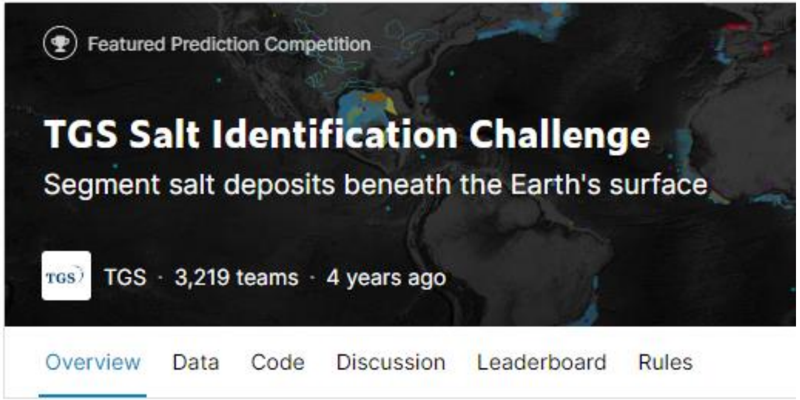
Explainable Salt Segmentation

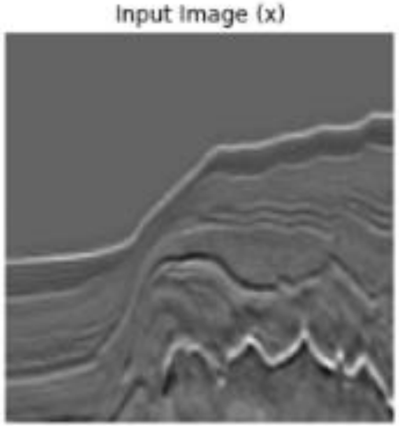
Edwin Brown
Marcos Jacinto
Team: Explainable Christmas

Introduction

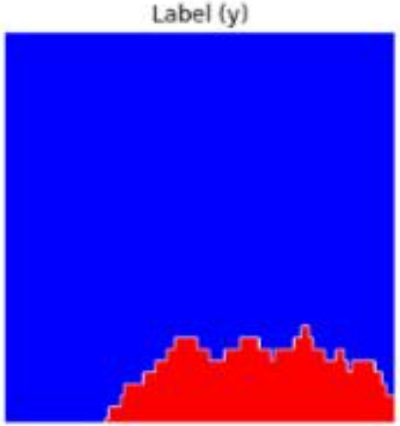
Understand how each feature in a multi-input segmentation task contributes to the predicted mask.



https://www.kaggle.com/c/tgs-salt-identification-challenge



Seismic

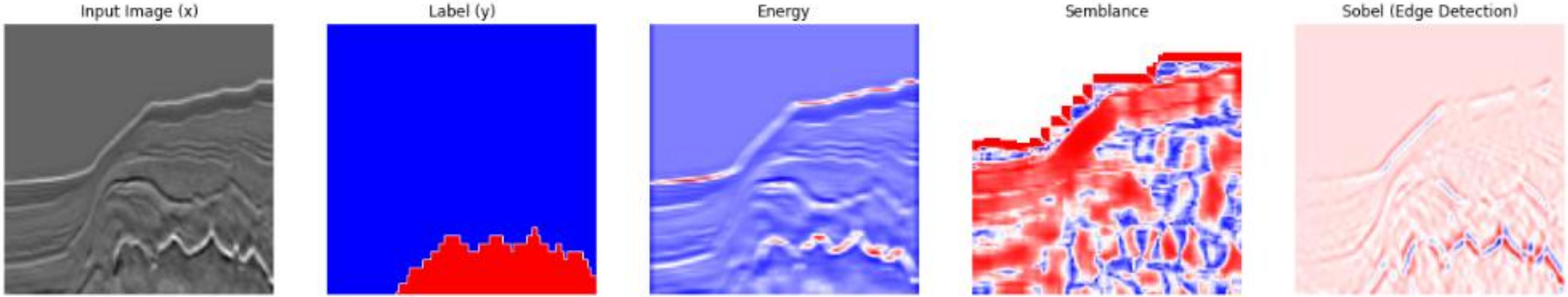


Salt Mask

Attribute Generation

Original X and Y Pair

Generated Features

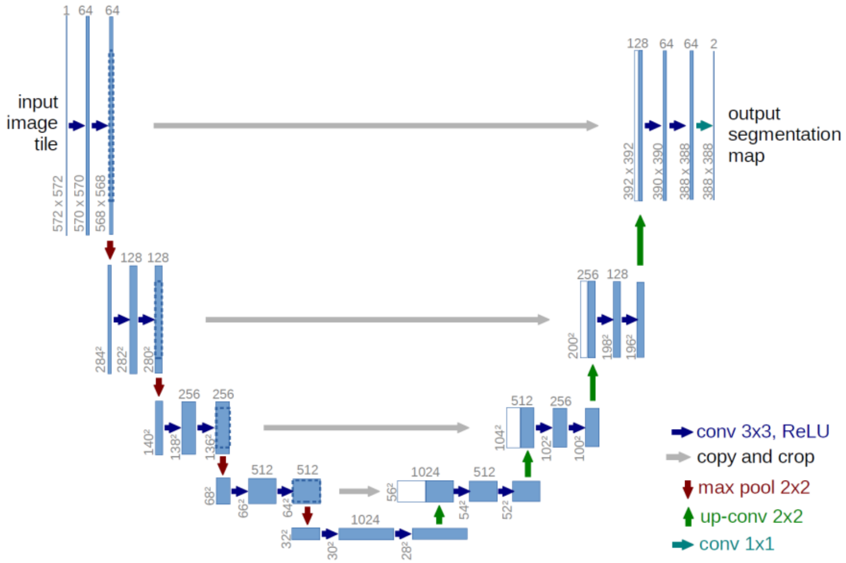


Low Wavelength Amplitude.

Red indicates areas of high spatial coherency within an image

Edge Detection Filter. Extreme values indicate a rapid contrast in values

Model Architecture



U-Net Architecture

Model: "model"

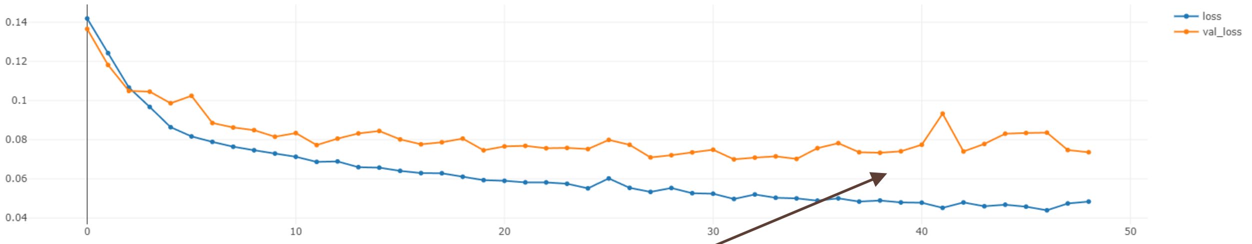
Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 24, 24, 4)]	0	[]
conv2d (Conv2D)	(None, 24, 24, 8)	808	['input_1[0][0]']
conv2d_1 (Conv2D)	(None, 24, 24, 8)	1608	['conv2d[0][0]']
max_pooling2d (MaxPooling2D)	(None, 12, 12, 8)	0	['conv2d_1[0][0]']
dropout (Dropout)	(None, 12, 12, 8)	0	['max_pooling2d[0][0]']
conv2d_2 (Conv2D)	(None, 12, 12, 16)	3216	['dropout[0][0]']
conv2d_3 (Conv2D)	(None, 12, 12, 16)	6416	['conv2d_2[0][0]']
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 16)	0	['conv2d_3[0][0]']
dropout_1 (Dropout)	(None, 6, 6, 16)	0	['max_pooling2d_1[0][0]']
conv2d_4 (Conv2D)	(None, 6, 6, 128)	51328	['dropout_1[0][0]']
conv2d_5 (Conv2D)	(None, 6, 6, 128)	409728	['conv2d_4[0][0]']
conv2d_transpose (Conv2DTranspose)	(None, 12, 12, 16)	51216	['conv2d_5[0][0]']
concatenate (Concatenate)	(None, 12, 12, 32)	0	['conv2d_transpose[0][0]', 'conv2d_3[0][0]']
dropout_2 (Dropout)	(None, 12, 12, 32)	0	['concatenate[0][0]']
conv2d_6 (Conv2D)	(None, 12, 12, 16)	12816	['dropout_2[0][0]']
conv2d_7 (Conv2D)	(None, 12, 12, 16)	6416	['conv2d_6[0][0]']
conv2d_transpose_1 (Conv2DTranspose)	(None, 24, 24, 8)	3208	['conv2d_7[0][0]']
concatenate_1 (Concatenate)	(None, 24, 24, 16)	0	['conv2d_transpose_1[0][0]', 'conv2d_1[0][0]']
dropout_3 (Dropout)	(None, 24, 24, 16)	0	['concatenate_1[0][0]']
conv2d_8 (Conv2D)	(None, 24, 24, 8)	3208	['dropout_3[0][0]']
conv2d_9 (Conv2D)	(None, 24, 24, 8)	1608	['conv2d_8[0][0]']
conv2d_10 (Conv2D)	(None, 24, 24, 1)	9	['conv2d_9[0][0]']

Total params: 551,585
Trainable params: 551,585
Non-trainable params: 0

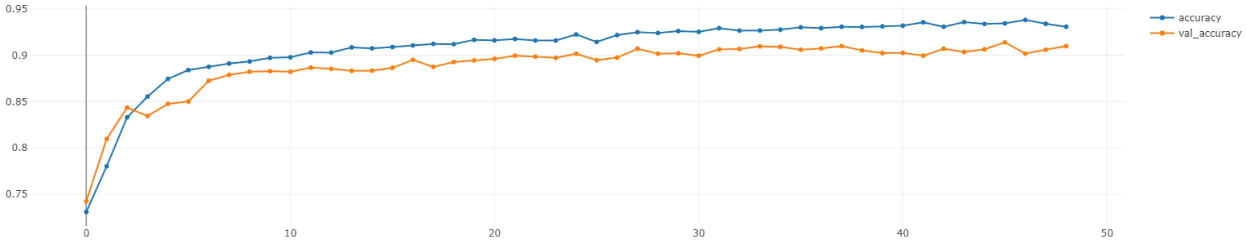
Built in Tensorflow/Keras

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

Learning Curves and Training Details

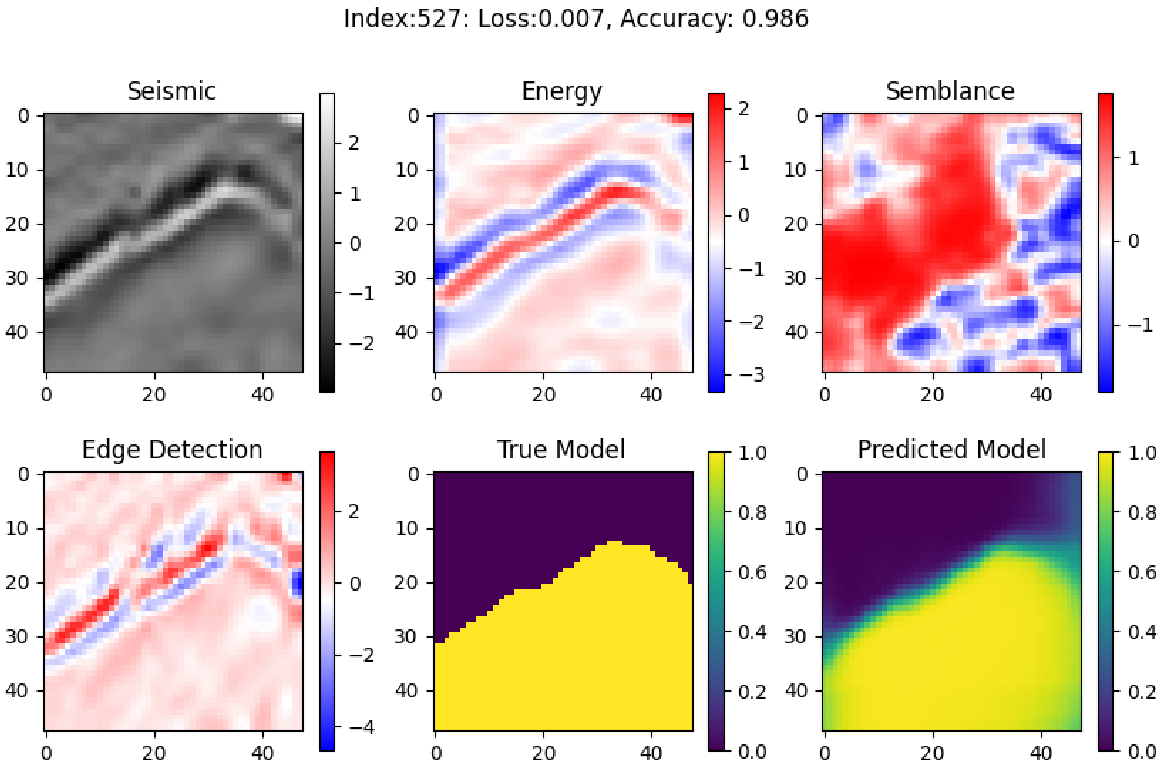


Widening of gap between training and validation loss
suggests overfitting

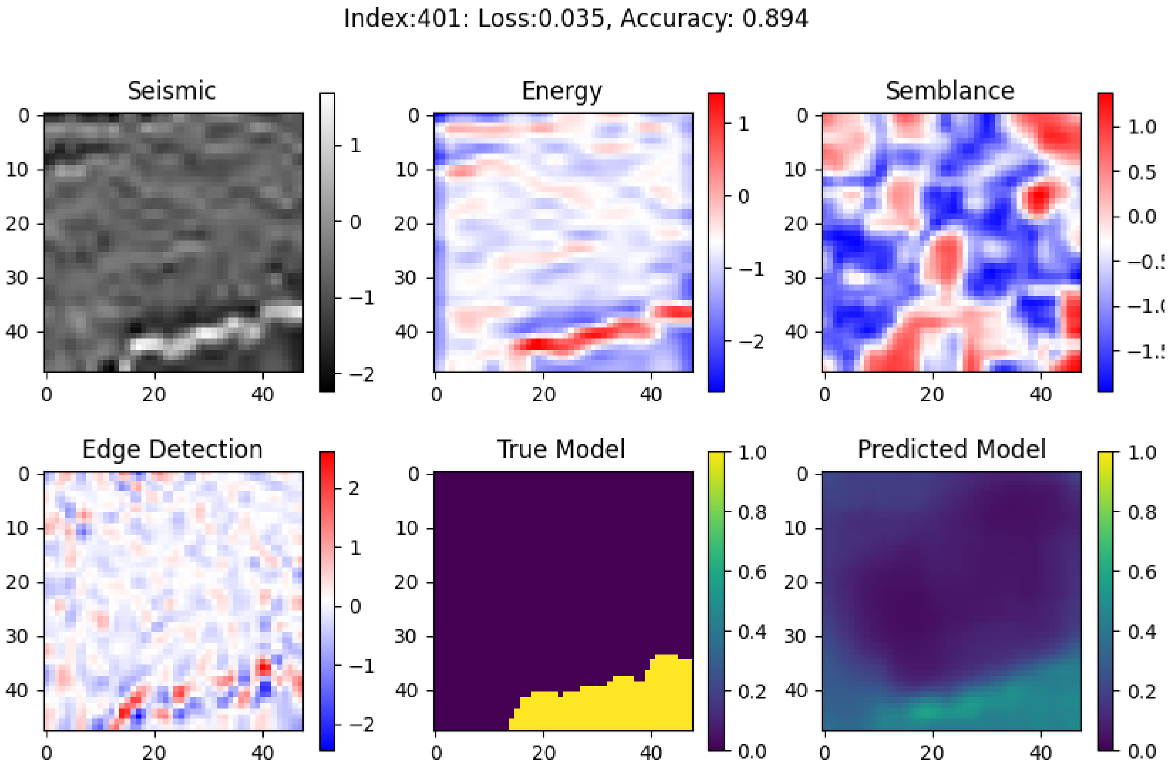


- Data split into training, validation and test datasets.
- Standardize input data
- Scale masks to 0-1
- ~2500 training pairs
- Batch Size – 32
- Learning Rate – 0.001

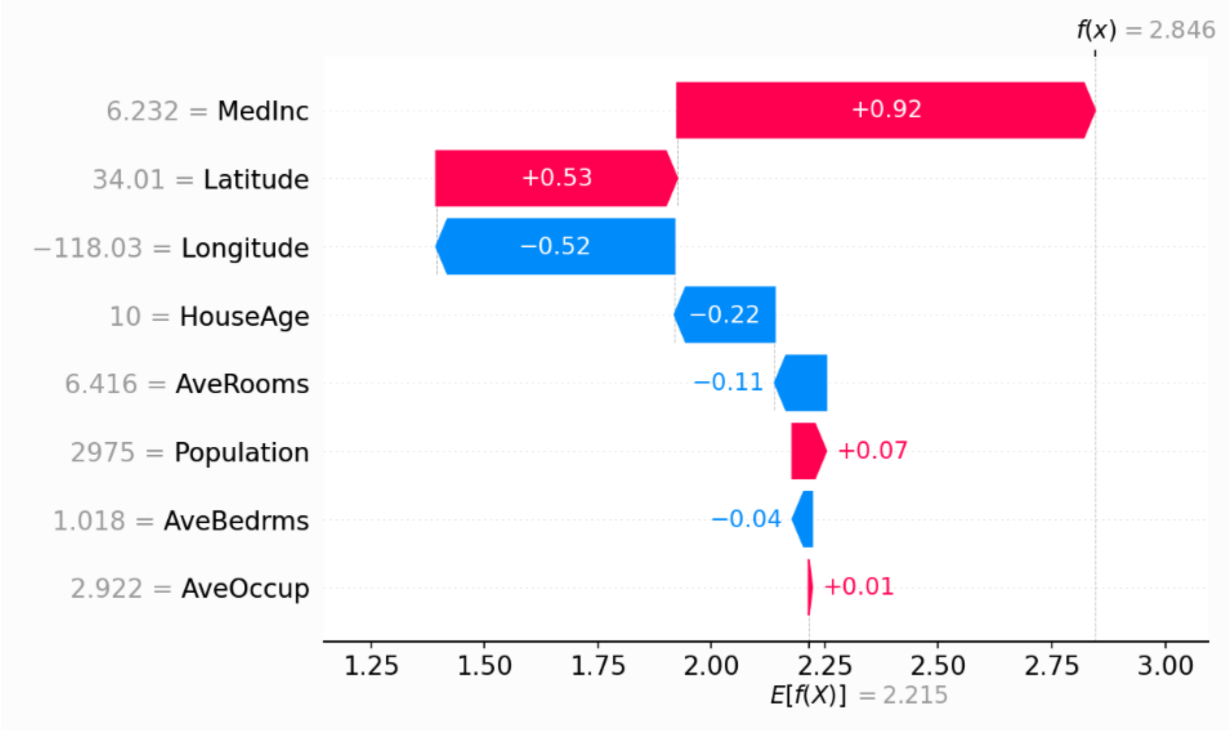
Segmentation Results



Segmentation Results



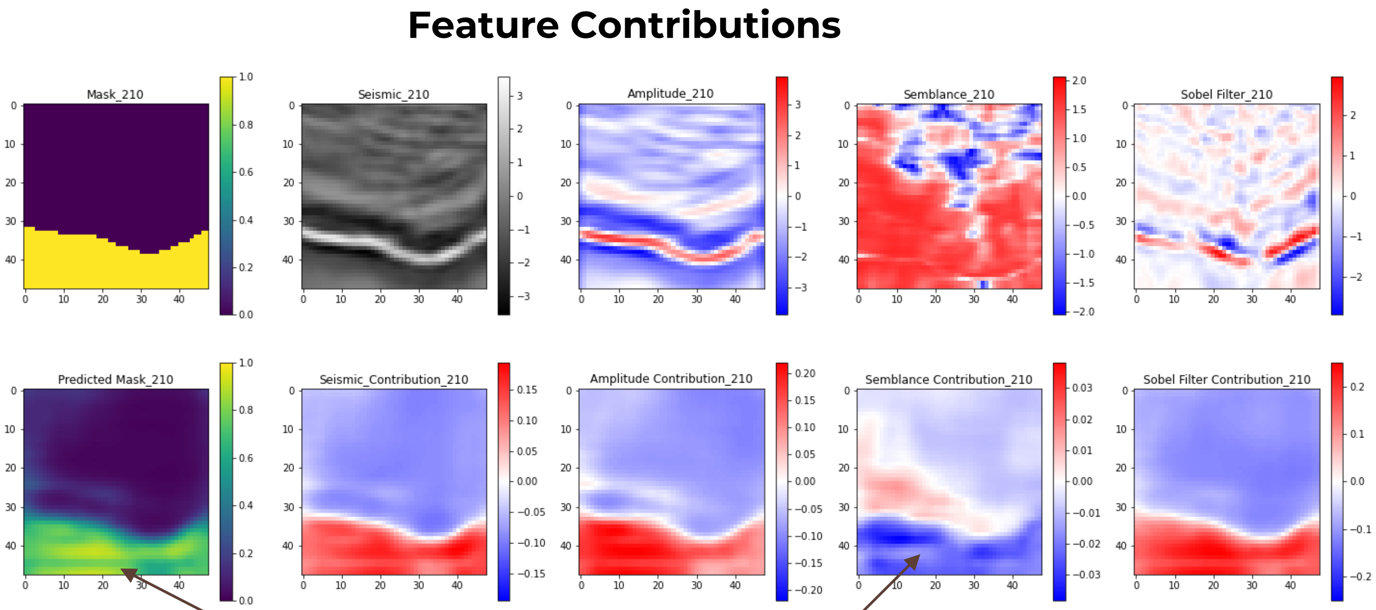
SHAP Values – Boston Housing Dataset - Example



https://shap.readthedocs.io

House Price ($100K)

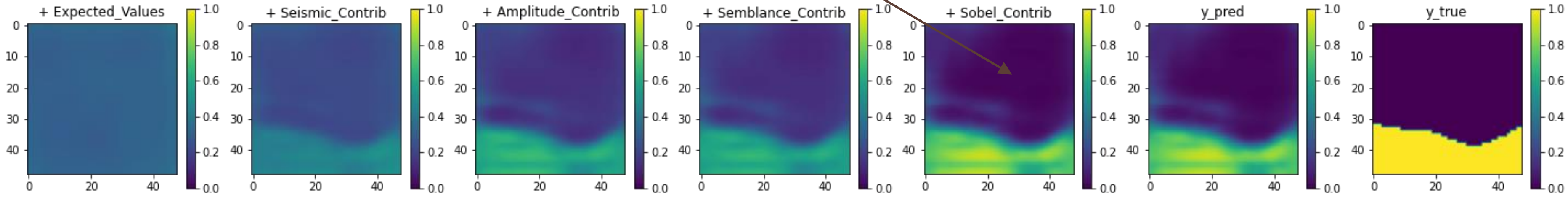
SHAP Values - Example



Semblance contribution is reducing the probability of salt in the predicted mask!

SHAP Values - Example

Sobel feature clearly contributes the most to reduce the background to 0 and increase the salt probability to 1.



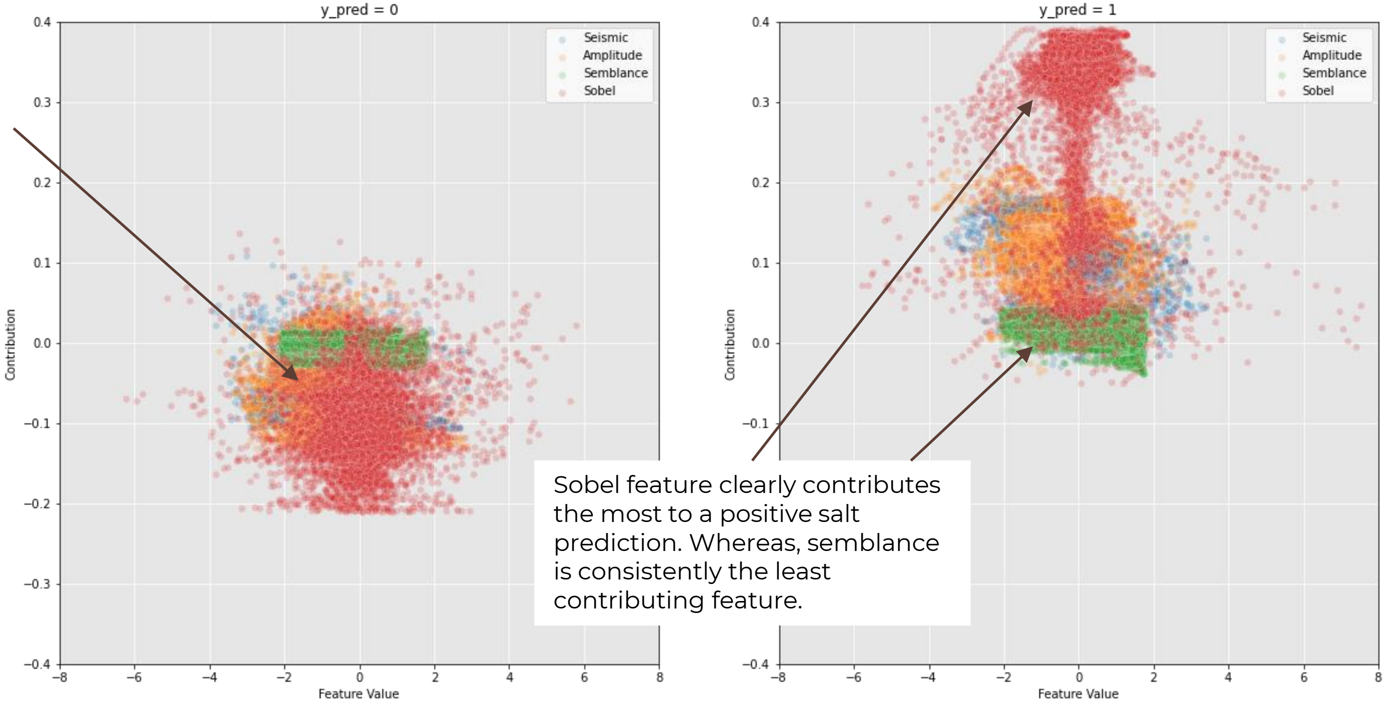
Very little change from Amplitude to Semblance suggests the semblance feature is having much impact on the prediction.

SHAP Values – Global Assessment

Feature value vs Feature Contribution. Different features are coloured.
Left Plot = No Salt, Right Plot = Salt

All features appear to have similar contributions in areas of no salt.

Note there appears to be no correlation between feature value and contribution. **This is expected because spatial relationships are vital for image data. This is why CNNs are often superior to other ML models with image data.**



Sobel feature clearly contributes the most to a positive salt prediction. Whereas, semblance is consistently the least contributing feature.

Repository

EdwinB12 update README.md a275ca1 3 hours ago 58 commits

Jobs	Delete utils.py	14 hours ago
Notebooks	Add image for readme	10 hours ago
Presentations	Repository Restructure	yesterday
outputs	Small changes to allow smoothing run with mlflow	15 hours ago
.gitignore	dockerfiles mlproject requirements	3 months ago
MLproject	Added environment.yml and edited requirements.txt and MLproject	3 hours ago
README.md	update README.md	3 hours ago
environment.yml	Added environment.yml and edited requirements.txt and MLproject	3 hours ago
requirements.txt	Added environment.yml and edited requirements.txt and MLproject	3 hours ago

Positives

- Clear conclusions. It can be seen that the Sobel feature is the most important, and semblance is the least important on the prediction of salt.
- Explainable AI is satisfying. It is important to understand why models make the predictions they do.
- Data is open source and the code is easy to run/edit.
- Entire project is reproducible and although a little slow, doesn’t require specialised hardware.
- Repository provides experiment tracking via MLFlow

Challenges and Further Work

- Computing the SHAP values **isn't optimized yet to GPUs. Very slow to generate SHAP values**

- Baseline model performance could be improved with some regularisation such as augmentation.

- Parameter tuning or transfer learning would likely improve performance too.

- No in-depth evaluation into segmentation performance. Where does our current model struggle the most?

- A refactor is probably required. Module names like 'process data.py' are not very helpful.

Food for thought. .

“Deep learning also makes problem-solving much easier, because it completely automates what used to be the most crucial step in a machine learning workflow: feature engineering. “

Francois Chollet, Deep Learning with Python, 2021