SHAP - SUMMARY

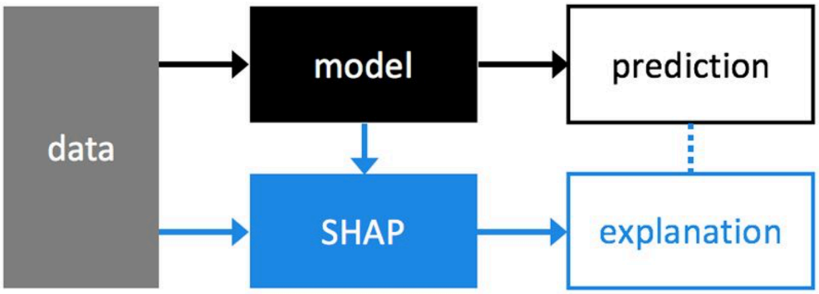
Project Part2 - Explainability

Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy in many applications. However, the highest accuracy for large modern datasets is often achieved by complex models that even experts struggle to interpret, such as ensemble or deep learning models, creating a tension between accuracy and interpretability. In response, various methods have recently been proposed to help users interpret the predictions of complex models, but it is often unclear how these methods are related and when one method is preferable over another.

A unified framework for interpreting predictions, **SHAP (SHapley Additive exPlanations)** - addresses this problem. *SHAP assigns each feature an importance value for a particular prediction. It is an additive feature attribution method to explain the output of any ML model*.Its novel components include:

- the identification of a new class of additive feature importance measures, and
- theoretical results showing there is a unique solution in this class with a set of desirable properties. The new class unifies three existing methods, notable because several recent methods in the class lack the proposed desirable properties.

FIG - Shows SHAP Block Diagram



STEPS Involved to Plugin COVID19-PNEUMONIA to SHAP

- Load the **Pre-trained persisted model** (model.h5)
- Plugin the pre-trained model in to SHAP for explanation
- Use **Gradient Explainer** for interpretation of last but layer in RNN
- Get the image to explain and visualize the results

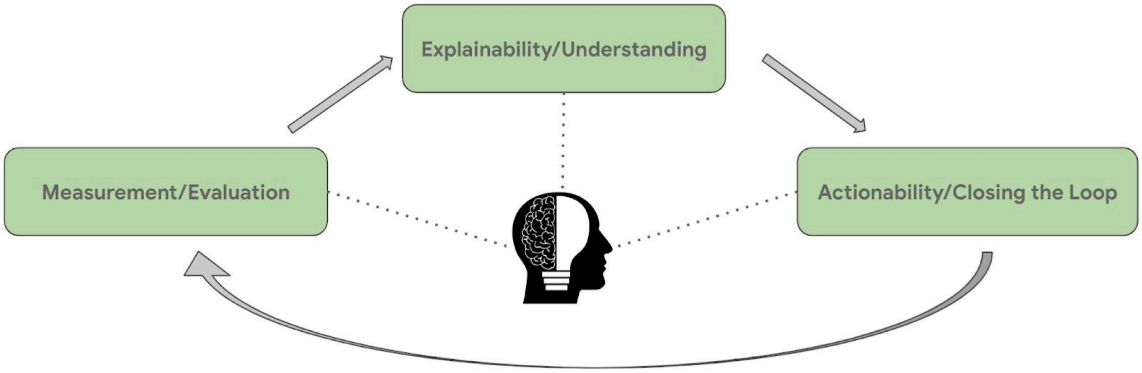
Note - Programmatically above steps will be covered in Part #3, however, below concepts explanation of SHAP will use some other examples for better explanation

Goals

- Determine if a machine learning classifier can be trained to distinguish cases of pneumonias vs covid from CXRs.
- Determine if a machine learning classifier can be trained to distinguish CXRs of cases of pneumonias from other normal cases.
- Apply a well-known explainability algorithm (SHAP - Gradient Explainer) to get explanations of the model's predictions. Use the explanations to inform feature engineering and data selection.

Model Development Workflow

During model development, explainability insights can be a powerful complement to model evaluation and a key enabler in closing the loop and achieving better models.



SHAP (SHapley Additive exPlanations) - *is a game theoretic approach to explain the output of any machine learning model*. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions.The three frequently used attribution includes,

- **Integrated Gradients** is recommended for neural networks and differentiable models in general. It offers computational advantages especially for large input feature spaces (e.g. images with thousands of input pixels).
- **XRAI** is recommended for image models where it's desirable to localize attributions at the region vs. pixel level.
- **Sampled Shapley** is recommended for tabular and non-differentiable models, which is the case in AutoML Tables models consisting of meta-ensembles of trees and neural networks. This method scales to the number of input features and is therefore slower for models with many input features.

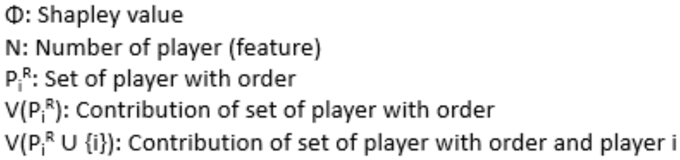
The complete question we're asking to explain for an AI model is "*why did the model make this prediction instead of that other prediction?*". The first part of the question is informed by the instance in question X and the second part is informed by the baseline .

What are Shapley values?

The Shapley value (proposed by Lloyd Shapley in 1953) is a classic method to distribute the total gains of a collaborative game to a coalition of cooperating players. It is provably the only distribution with certain desirable properties.In our case, for understanding, we formulate a game for the prediction at each instance. We consider the “total gains” to be the prediction value for that instance, and the “players” to be the model features of that instance. The collaborative game is all of the model features cooperating to form a prediction value. *The Shapley value efficiency property says the feature attributions should sum to the prediction value*. The attributions can be negative or positive, since a feature can lower or raise a predicted value.

Below is the equation of Shapley value which is a solution concept in cooperative game theory.

$$\phi_i(v) = \frac{1}{|N|!} \sum_{R} \left[v(P_i^R \cup \{i\}) - v(P_i^R) \right]$$



Φ: Shapley value

N: Number of player (feature)

P_i^R: Set of player with order

$V(P_i^R)$: Contribution of set of player with order

$V(P_i^R \cup \{i\}$): Contribution of set of player with order and player i

What is a Shapley-value-based explanation method?

A Shapley-value-based explanation method tries to approximate Shapley values of a given prediction by examining the effect of removing a feature under all possible combinations of presence or absence of the other features.

In other words, this method looks at function values over subsets of features like ***F(x1, <absent>, x3, x4, …, <absent>, …, xn)***. *How to evaluate a function F with one or more absent features is subtle.*

For example, SHAP (SHapely Additive exPlanations) estimates the model's behavior on an input with certain features absent by averaging over samples from those features drawn from the training set.
In other words, F(x1, <absent>, x3, …, xn) is estimated by the expected prediction when the missing feature x2 is sampled from the dataset.

Seeking Explanations

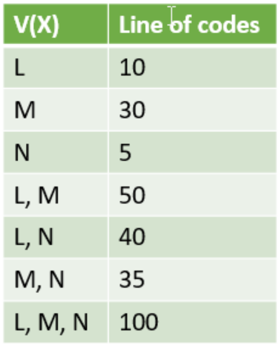
In the context of AI, we summarize the following explanations purposes:

- Informing and supporting human decision making when AI is used as a decision aid.
- Improving transparency by creating a shared understanding between the AI and the human.
- Enabling debugging when the system behaves unexpectedly.
- Enabling auditing in the context of regulatory requirements.

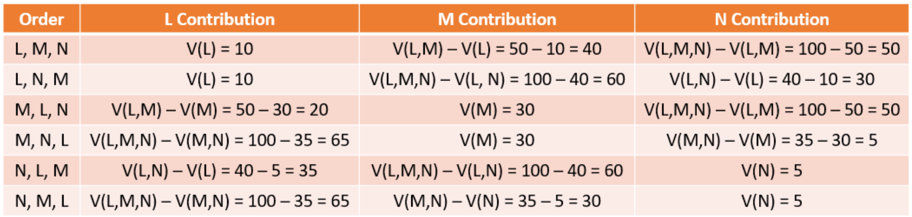
- Verifying generalization ability and moderating trust to appropriate levels.
- Through these functions, a model user can make better decisions and a model builder can improve the model itself.

Development Team - Example to demonstrate Shapley

Let’s take a development team as an example. Our target is going to *deliver a deep learning model which needs to finish 100 line of codes while we have 3 data scientists (L, M, N)*. 3 of them must work together in order to deliver the project. Given that:

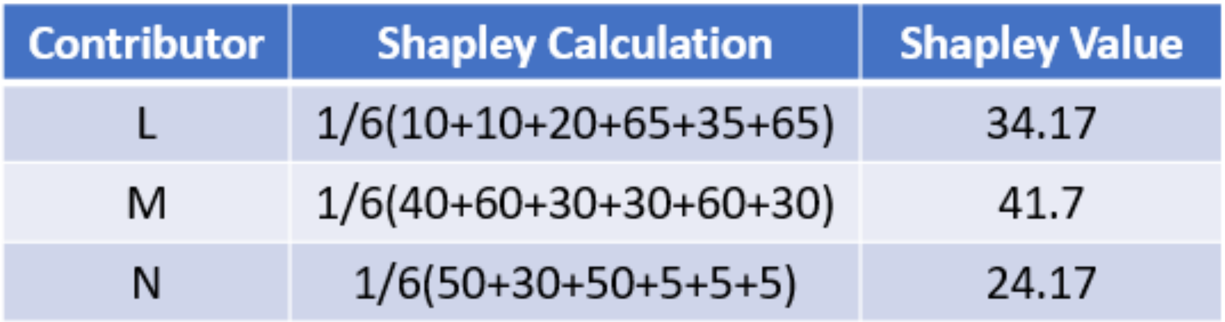


V(X)	Line of codes
L	10
M	30
N	5
L, M	50
L, N	40
M, N	35
L, M, N	100



Order	L Contribution	M Contribution	N Contribution
L, M, N	V(L) = 10	V(L,M) – V(L) = 50 – 10 = 40	V(L,M,N) – V(L,M) = 100 – 50 = 50
L, N, M	V(L) = 10	V(L,M,N) – V(L, N) = 100 – 40 = 60	V(L,N) – V(L) = 40 – 10 = 30
M, L, N	V(L,M) – V(M) = 50 – 30 = 20	V(M) = 30	V(L,M,N) – V(L,M) = 100 – 50 = 50
M, N, L	V(L,M,N) – V(M,N) = 100 – 35 = 65	V(M) = 30	V(M,N) – V(M) = 35 – 30 = 5
N, L, M	V(L,N) – V(L) = 40 – 5 = 35	V(L,M,N) – V(L,N) = 100 – 40 = 60	V(N) = 5
N, M, L	V(L,M,N) – V(M,N) = 100 – 35 = 65	V(M,N) – V(N) = 35 – 5 = 30	V(N) = 5

We have 3 player so the total combination is 3! which is 6. The above tables show the contribution according to different order of coalition.



Contributor	Shapley Calculation	Shapley Value
L	1/6(10+10+20+65+35+65)	34.17
M	1/6(40+60+30+30+60+30)	41.7
N	1/6(50+30+50+5+5+5)	24.17

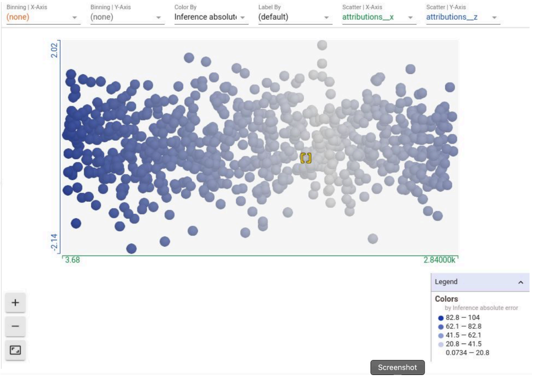
According to the Sherley value formula, we have the above tables. **Although the capability of M is 6 times greater than N (30 vs 5), M should get 41.7% of reward while N should get 24.17% reward.**

By using the the Shapley formula, SHAP will compute all above scenario and returning the average contribution and . In other word, it is **not talking about the difference when the particular feature missed**.

The Shapley value is the unique method that satisfies the following desirable axioms, which
motivates its use and are the main reason we chose it to power our feature attributions offering:
1. Completeness (also known as efficiency):
2. Symmetry: For two participants i, j and S, where S is any subset of participants not including i, j, when {S, i} = {S, j} → i and j should have the same attribution value.
3. Dummy: For one participant i and S, where S is any subset of participants not including i, when {S, i} = {S} → i should get zero attribution.
4. Additivity: For two outcome functions, the Shapley values of the sum of the two outcome functions should be the sum of the Shapley values of the two outcome functions.

Visualizations - for Shap

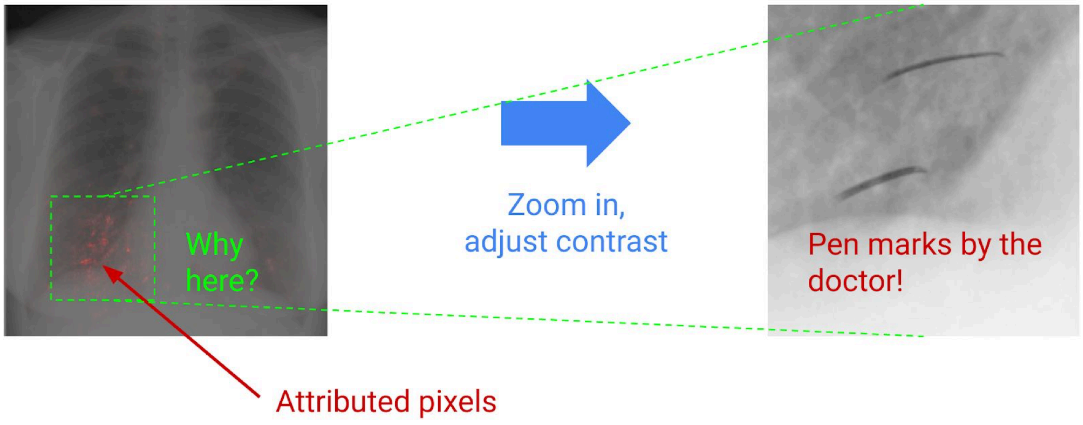
Provide very good Visualization of explainable logs, results.



Binning | X-Axis (none)
Binning | Y-Axis (none)
Color By Inference absolut
Label By (default)
Scatter | X-Axis attributions__x
Scatter | Y-Axis attributions__z

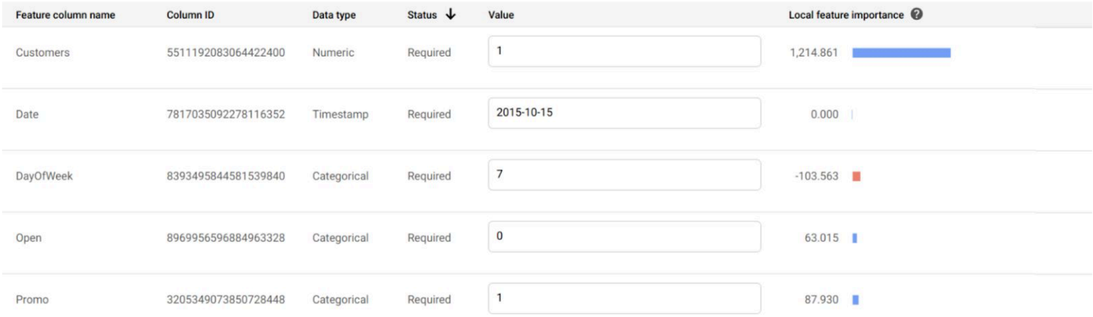
Detecting Data Issues (COVID-19 - Pneumonia use case)

As can be seen above, the attributions were clustering around a seemingly odd region in the X-Ray. Upon closer examination, this area is where



radiologist left pen marks. The model learned to rely on these pen marks, which is clearly not desirable from the perspective of being able to generalize to new/unseen instances. Should this model have been deployed, it would be running on images without these pen marks, and its performance would've been markedly worse than the holdout set results.

Raising Flag on a single feature dependency



Feature column name	Column ID	Data type	Status ↓	Value	Local feature importance
Customers	5511192083064422400	Numeric	Required	1	1,214.861
Date	7817035092278116352	Timestamp	Required	2015-10-15	0.000
DayOfWeek	8393495844581539840	Categorical	Required	7	-103.563
Open	8969956596884963328	Categorical	Required	0	63.015
Promo	3205349073850728448	Categorical	Required	1	87.930

Prediction Auditing and Model Monitoring

Once a model is deployed and is serving live traffic, there are a few ways explainability can help:

- Auditing predictions, especially for the rare/unlikely class. For example, say we've deployed a model to monitor for fraudulent transactions, we can leverage attributions on the percentage of traffic that the model tags as fraud to record into logs which can be analyzed by a human reviewer if/when needed.

- Monitoring the model to ensure it's operating correctly. Traditional techniques monitor for training/serving skew on the model's feature values and predictions.

Conceptual Limitations

1. Attributions depend on the model:
2. Attributions depend on baselines
3. Attributions are communicated at the level of input features, this entails some loss of information
4. Attributions do not summarize the entire model behavior
5. Attributions are efficient approximations of Shapley values