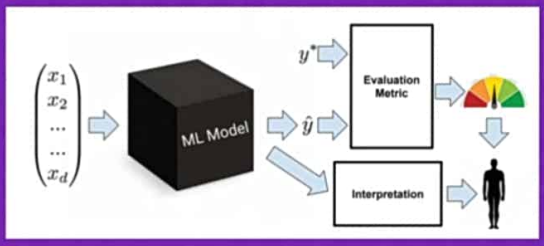
Model Interpretability

- By model interpretation, one can be able to understand the algorithmic decisions of a machine learning model
- It is the ability to approve and interpret the decisions of a predictive model in order to enable transparency in the decision-making process



Python Libraries for Building Interpretable Machine Learning

- ✓ ELI5
- ✓ LIME
- ✓ SHAP
- ✓ Yellowbrick
- ✓ Alibi
- ✓ Lucid

To interpret the models using the predictions and parameters to understand why the cl chose a particular class for example

WHY DO WE NEED INTERPRETABLE ML?

Fairness

- Employees' performance, data from the last 10 years,
- what if that company tends to promote more men than women? (and this bias has unfortunately happened in certain real-world scenarios).
- Now, if there is no way to interpret our model at this stage, the model might end up providing false insights at the cost of compromising on fairness

WHY DO WE NEED INTERPRETABLE ML?

Checking causality of features & Debugging models

- Consider that we are building a model for classifying wolves vs dogs.
- Now, what if we have wolves and dogs in entirely different backgrounds?
- But on using one of the interpretability methods, we see that our model is actually ignoring the dog and wolf while using just the background pixels for doing the classification.
- This model might give a good performance on the validation set as they contain different backgrounds for the wolf and dog respectively
- Having an interpretable model, in this case, enables us to test the causality of the features, test its reliability and ultimately can help us to debug the model appropriately.

WHY DO WE NEED INTERPRETABLE ML?

Regulations

- In the banking and finance industry, questions such as the following can come up and have to be answered by these banks:

- Why was my loan rejected?
- Why did I get a low Credit Limit on my credit card? etc.

When we no Need Interpretable ML?

- When interpretability does not impact the end customer
- If the problem is well studied, we are confident about the results

Classification of Interpretability Techniques

- **Scope**

Whether we are looking to interpret globally for all data points, the importance of each variable, or are we looking to explain a particular prediction which is local?

- **Model**

Is whether we are talking about a technique that works across all types of models (**model agnostic**) or is tailor-made for a particular class of algorithms (**model specific**).

YELLOWBRICK

- Yellowbrick is used to evaluate the model performance and visualize the model behavior
- It is based on the scikit-learn and matplotlib libraries
- Yellowbrick uses the concept of 'Visualisers'. Visualizers are a set of tools that help us visualize the features in our data considering individual datapoints
- Visualizers are a set of tools that help us visualize the features in our data considering individual datapoints

List of Visualizers

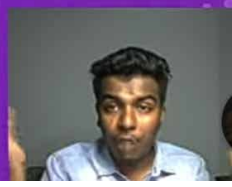
- ✓ **Rank Features**
- ✓ **RadViz Visualizer**
- ✓ **Parallel Coordinates**
- ✓ **PCA Projection**
- ✓ **Manifold Visualization**
- ✓ **Direct Data Visualization/Joint Plot Visualiser**

ELI5

- Inspect model parameters and try to figure out how the model works globally
- Inspect an individual prediction of a model and figure out why the model makes the decision.
- If Yellowbrick focuses on the features and model performance interpretation, ELI5 focus on the model parameters and the prediction result

SHAP-SHAPLEY ADDITIVE EXPLANATIONS

- Uses Shapley values at its core and is aimed at explaining individual predictions
- Shapley values are derived from Game Theory, where each feature in our data is a player, and the final reward is the prediction. Depending on the reward, Shapley values tell us how to distribute this reward among the players fairly
- SHAP using the SHAP values to explain the importance of each feature
- SHAP is not limited to global interpretability; it also gives you the function to interpret individual dataset
- The best part about SHAP is that it offers a special module for tree-based models



LIME -LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS

- Would you just pull up a review of a movie from some random person and watch it if the person recommends it? No, right? You would read what a well-known movie-critic has to say about the movie and then make a decision. This is because you trust the movie critic's opinion
- Similarly the keyword for building a machine learning model is trust.
- It is to provide the reasons why a prediction was made
- If a machine learning model predicts that a movie is going to be a blockbuster, LIME highlights the characteristics of the movie that would make it a super hit

1. **Interpretable**: The explanation must be easy to understand depending on the target demographic
2. **Local fidelity**: The explanation should be able to explain how the model behaves individual predictions
3. **Model-agnostic**: The method should be able to explain any model
4. **Global perspective**: The model, as a whole, should be considered while explaining

12_DL_ModelInterpretability.ipyn

colab.research.google.com/drive/1Yd_YzKya1D4K2rCVUm5Jx-GIFHVJET23#scrollTo=sXV2dObbvxbL

12_DL_ModelInterpretability.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM
Disk

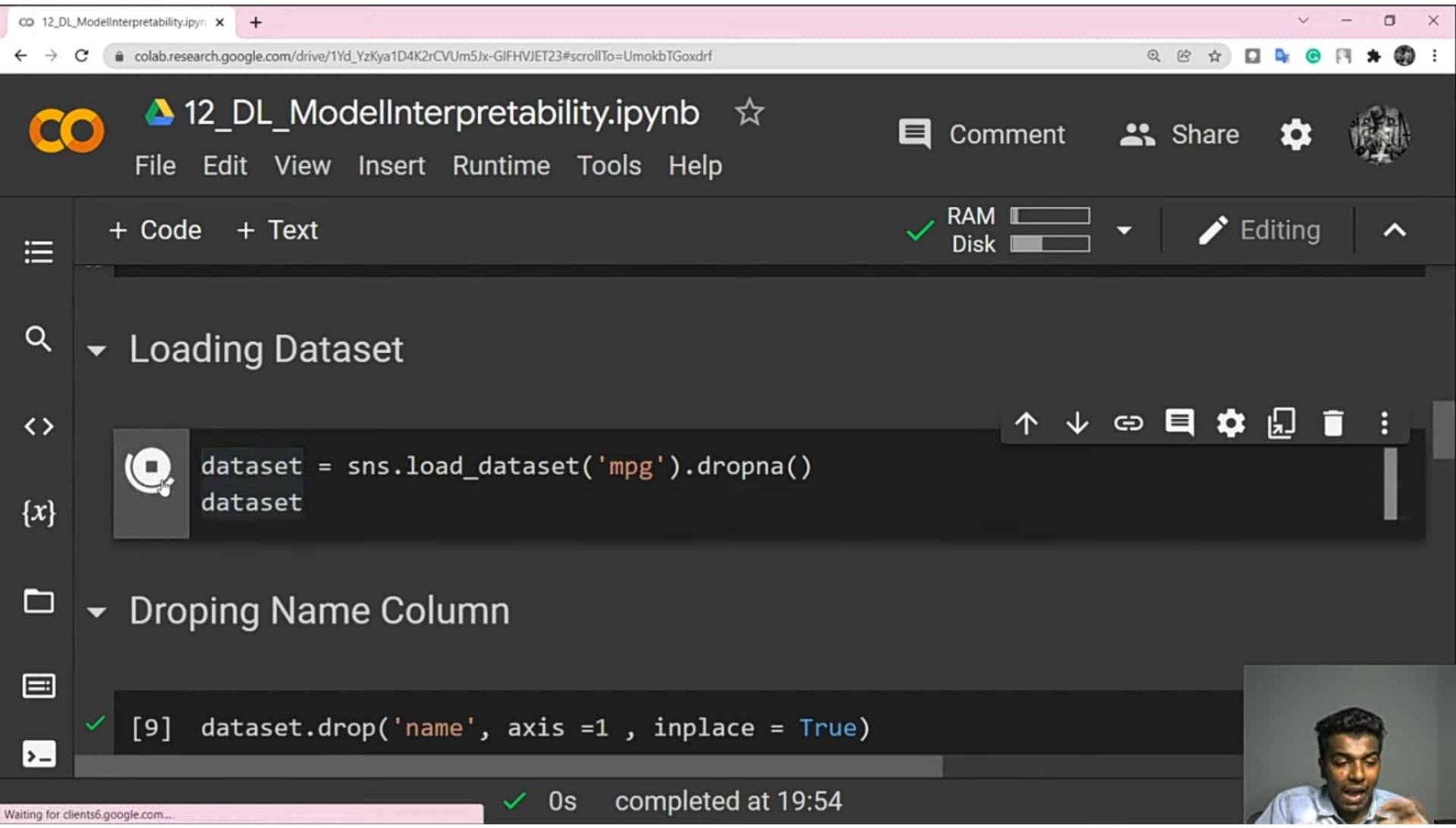
Editing

Installing Eli5

```
!pip install eli5
```

Importing Libraries

```
[7]  from sklearn.ensemble import RandomForestClassifier
     from sklearn.model_selection import train_test_split
     import seaborn as sns
```


12_DL_ModelInterpretability.ipynb
colab.research.google.com/drive/1Yd_YzKya1D4K2rCVUm5Jx-GIFHVJET23#scrollTo=UmokbTGoxdrf
12_DL_ModelInterpretability.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
Loading Dataset
dataset = sns.load_dataset('mpg').dropna()
dataset
Droping Name Column
[9] dataset.drop('name', axis =1 , inplace = True)
0s completed at 19:54
Waiting for clients6.google.com...

12_DL_ModelInterpretability.ipyn

colab.research.google.com/drive/1Yd_YzKya1D4K2rCVUm5Jx-GIFHVJET23#scrollTo=UJMqbFEazPJw

12_DL_ModelInterpretability.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

392 rows × 8 columns

Splitting Dataset

```
[30] X_train, X_test, y_train, y_test = train_test_split(dataset.drop('origin', axis = 1), mpg['origin'], test_size = 0.2, random_state = 121)
```

Model Training

```
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

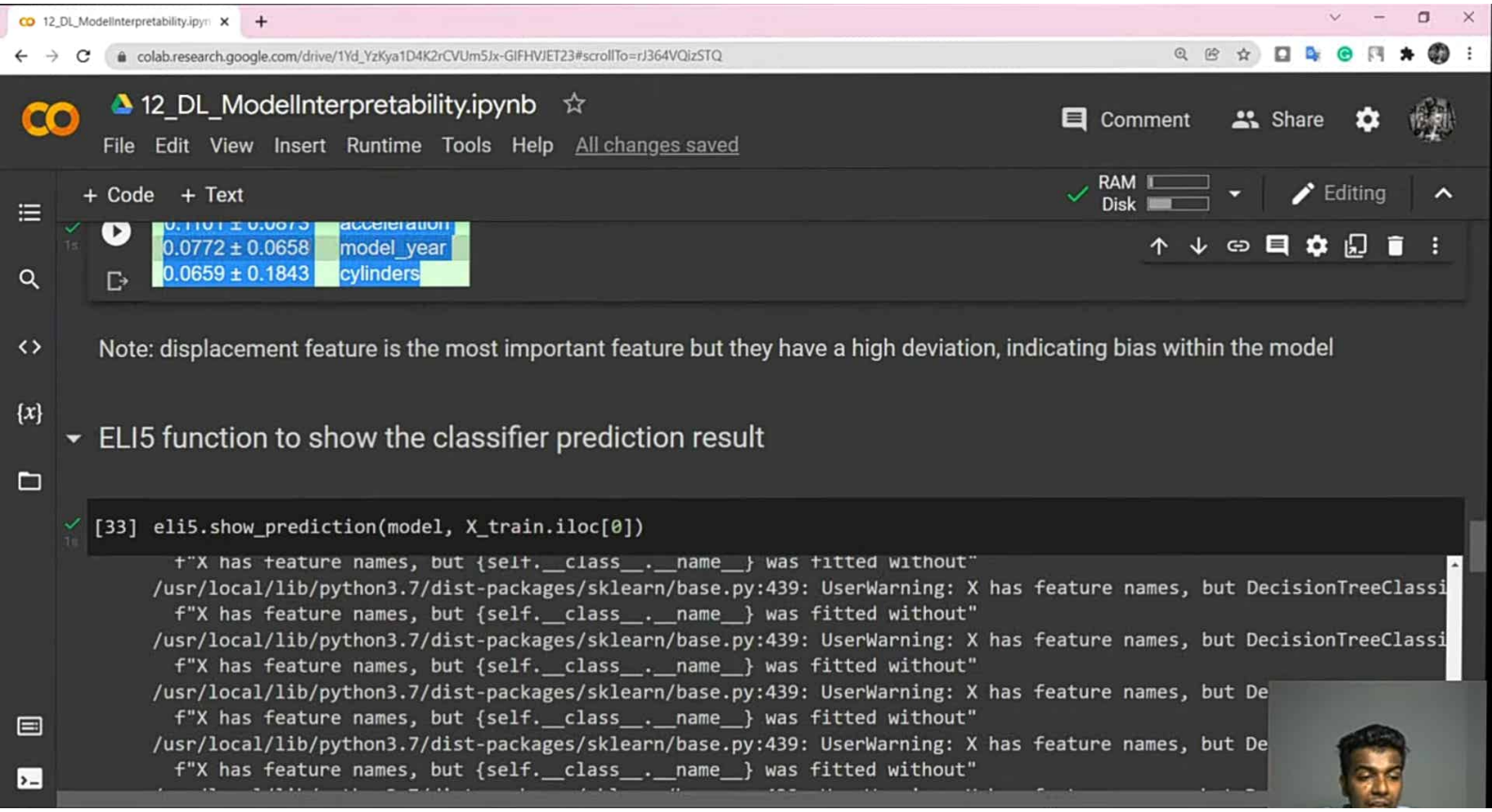
```
RandomForestClassifier()
```

ELI5 function to show the classifier weight

```
[16] import eli5
     eli5.show_weights(model, feature_names = list(X_test.columns))
```

Note: displacement feature is the most important feature but they have a high deviation, indicating bias within the model

0s completed at 19:59


12_DL_ModelInterpretability.ipynb
colab.research.google.com/drive/1Yd_YzKya1D4K2rCVUm5Jx-GIFHVJET23#scrollTo=rJ364VQizSTQ
12_DL_ModelInterpretability.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0.0772 ± 0.0658 model_year
0.0659 ± 0.1843 cylinders
Note: displacement feature is the most important feature but they have a high deviation, indicating bias within the model
ELI5 function to show the classifier prediction result
[33] eli5.show_prediction(model, X_train.iloc[0])
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:439: UserWarning: X has feature names, but DecisionTreeClassi
f"X has feature names, but {self.__class__.__name__} was fitted without"

12_DL_ModelInterpretability.ipyn

colab.research.google.com/drive/1Yd_YzKya1D4K2rCVUm5Jx-GIFHVJET23#scrollTo=zcOHswBu0_Nh

12_DL_ModelInterpretability.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

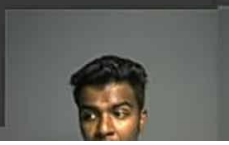
Editing

Note: Using the show prediction function by ELI5, we could get the feature contribution information. What features contribute to certain prediction results, and how much is the probability shifting by these features.

Black-box model based on the model metric — It is called Permutation Importance

```
from eli5.sklearn import PermutationImportance
perm = PermutationImportance(model, scoring = 'accuracy', random_state=101).fit(X_test, y_test)
eli5.show_weights(perm, feature_names = list(X_test.columns))
```

Weight	Feature
0.2962 ± 0.0612	displacement
0.0405 ± 0.0372	horsepower
0.0253 ± 0.0160	cylinders
0.0228 ± 0.0436	weight
0.0127 ± 0.0320	model_year



Deep Learning Terminology - 1

Embeddings

- An embedding is a mapping of a discrete — categorical — variable to a vector of continuous numbers
- Typically, an embedding is a translation of a high-dimensional vector into a low-dimensional space
- In TensorFlow, embeddings are trained by backpropagating loss just like any other parameter in a neural network.

1. Finding nearest neighbors in the embedding space. These can be used to make recommendations based on user interests or cluster categories.
2. As input to a machine learning model for a supervised task.
3. For visualization of concepts and relations between categories

Deep Learning Terminology - 2

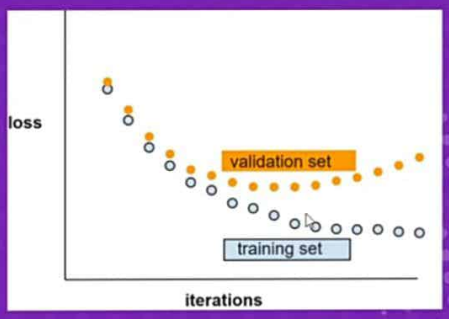
Ensemble

- A merger of the predictions of multiple models.
- Different initializations
- Different hyperparameters
- Different overall structure

Deep Learning Terminology - 3

Generalization

- Refers to your model's ability to make correct predictions on new, previously unseen data as opposed to the data used to train the model.



Generalization curve

A loss curve showing both the training set and the validation set. A generalization curve can help you detect possible overfitting

For example, the following generalization curve suggests overfitting because loss for the validation set ultimately becomes significantly higher than for the training set