HAROKOPIO UNIVERSITY
DEPARTMENT OF INFORMATION & TELEMATICS

2nd Assignment: Human-Machine Communication

Club:

Kazakos Christos, it22033

Konstantinos Katsaras, it22045

Manousos Linardakis, it22064

Observation: To make the predictions, we first normalize the data us (better results).

Description 1

First, the required libraries are imported. The dataset is then copied from Colab. The data set is then loaded by printing statistics for nan, numeric and categorical values. The 'type' column is removed, as it is not included in the reported pronunciation columns (description 1). Next, the nan values are removed because they are an extremely small percentage of the data set. Finally, we proceed to normalize the data in order to achieve optimal results.

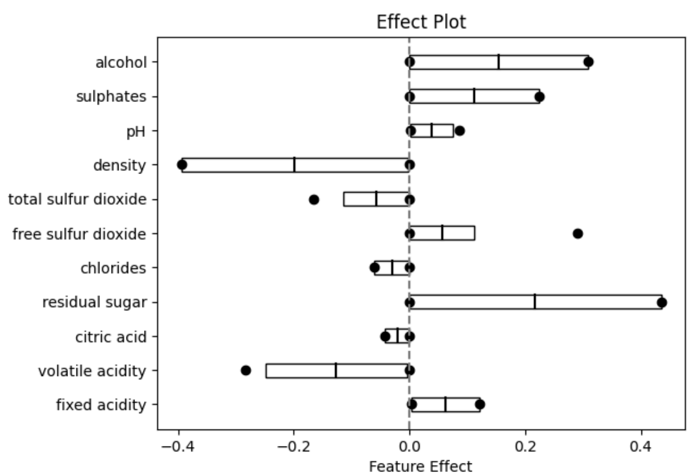
Wanted

Required A:

R-Squared outputs**0.35**, which does not bode well for the model's overall ability to explain the data. The usual range of R-squared values is between '0' (model does not explain the data at all) and '1' (models explain all the variation in the data).

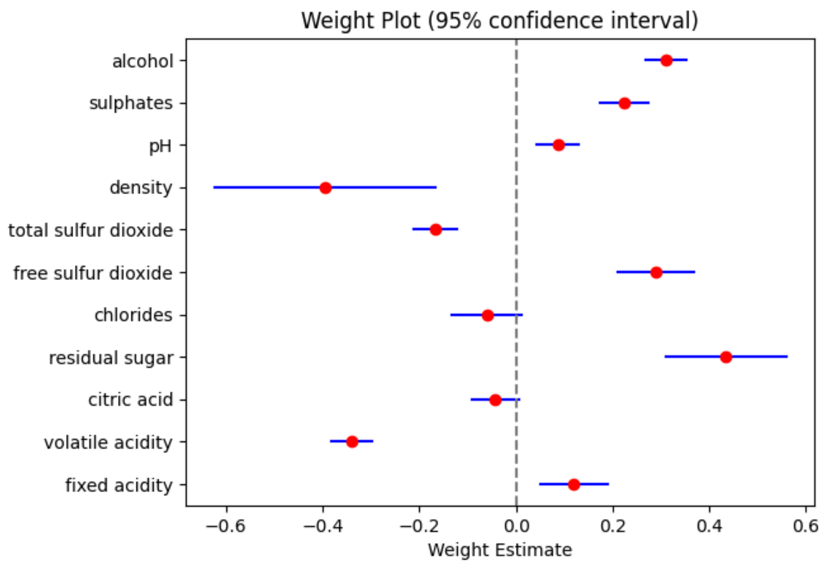
Required B:

<u>Effects Plot:</u>



According to the diagram above, the features with the biggest positive contribution are "density" as we can see from the size of the boxes (followed by the "alcohol" feature). "residual sugar" has the biggest negative contribution, followed by "volatile acidity".

Weight Plot:

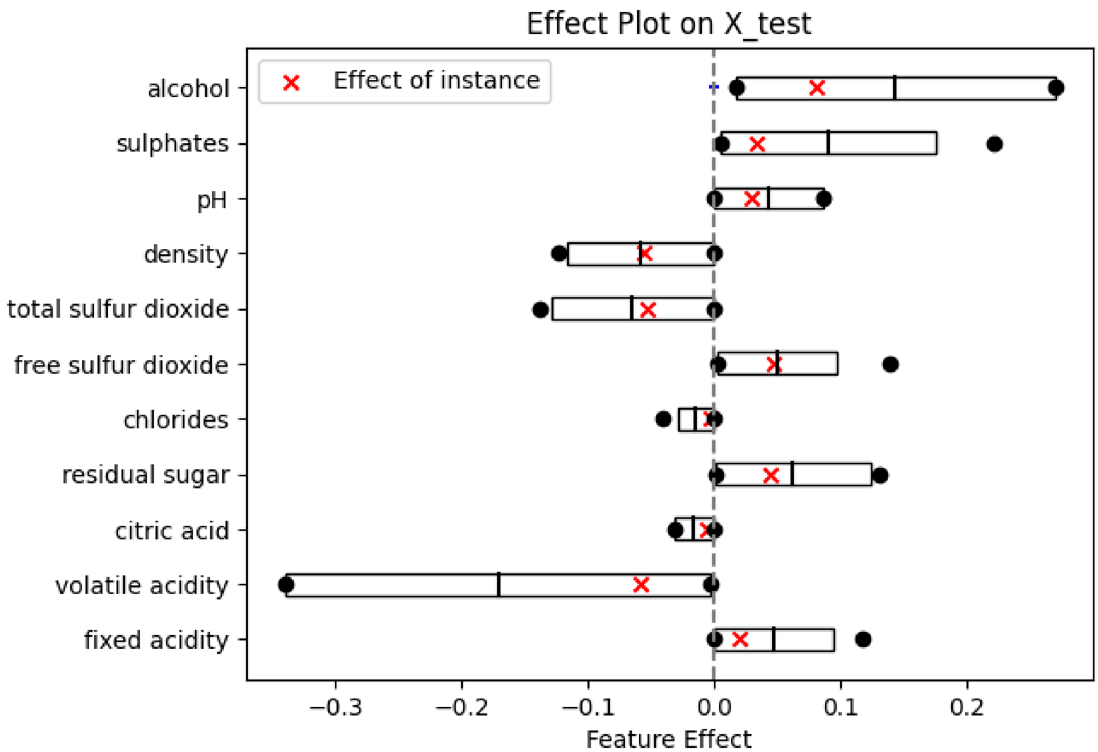


We observe that the effect of the trait "residual sugar" has the greatest positive effect (compared to the rest of the traits), followed by free sulfur dioxide. The "density" feature has the biggest negative effect.

On the other hand, features such as pH, chlorides, citric acid, fixed acidity do not have a statistically significant effect on the prediction.

Observation: Forecasts have a confidence interval95%

Required C:



We notice that the predicted value of the specific sample is**bigger** from the mean predicted value.

From the figure, we can see that the instance effect for each feature is within the expected range. Specifically, sample effects are shown by the red "X's" in the diagram.

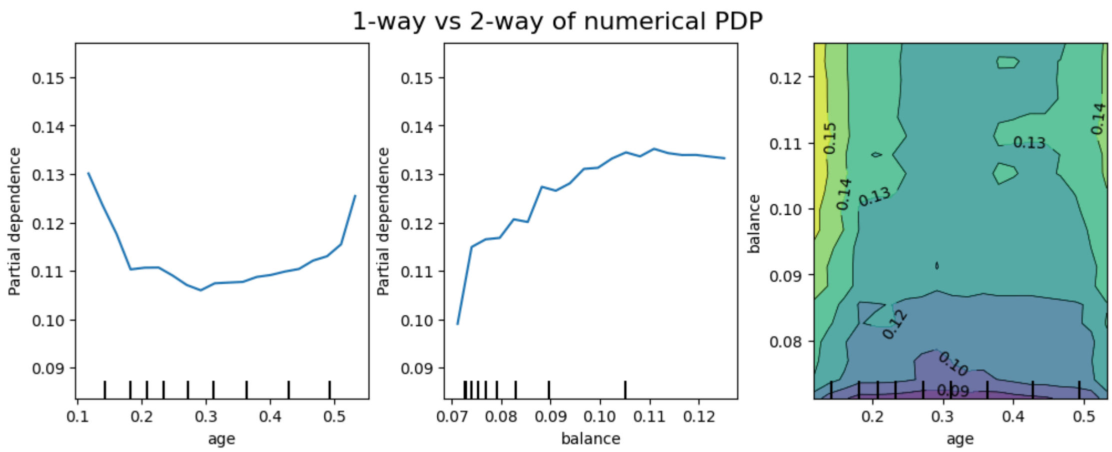
Description 2

Code

First, we import the required libraries. Next, we load the dataset, as described in the dataset manual (import in python). In the preprocessing step, we convert the categorical variables into numeric ones via one-hot encoding. In addition, we normalize the values   so that they range from 0 to 1. Finally, we execute the code mentioned in the second description.

Wanted

Required A

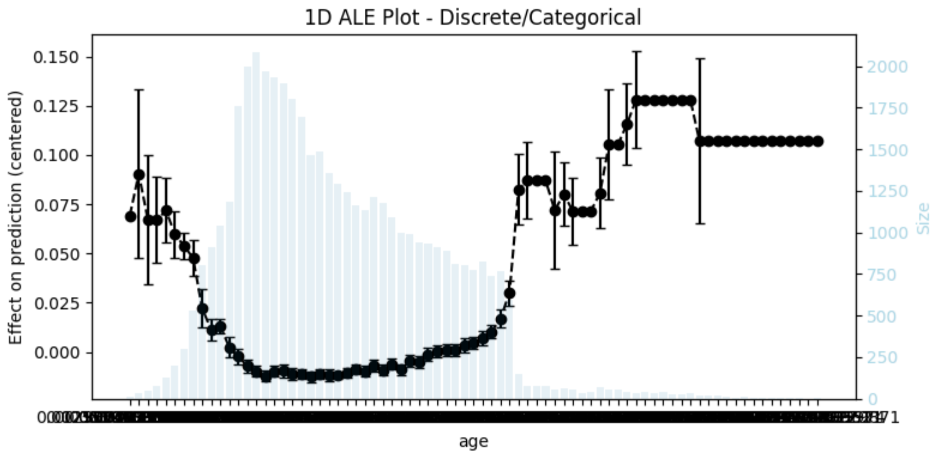


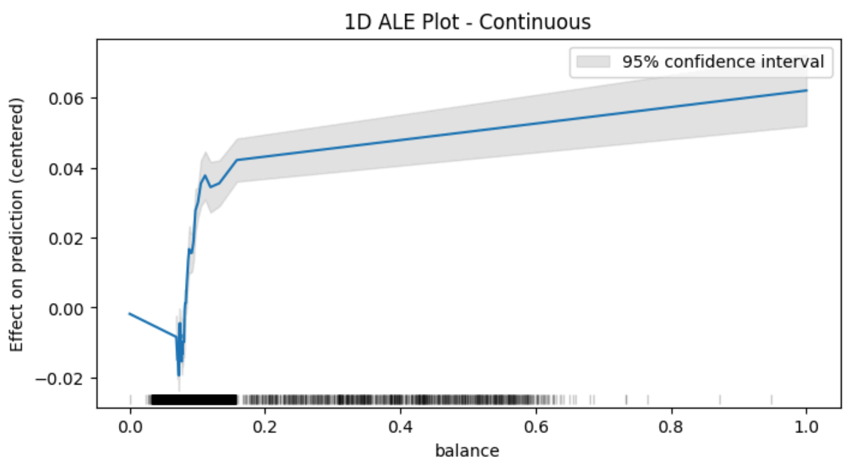
- The first (left) graph shows the partial dependence against 'Age'. A blue line in this first plot indicates an initial decrease in partial dependence as age increases, followed by a U-shaped increase.

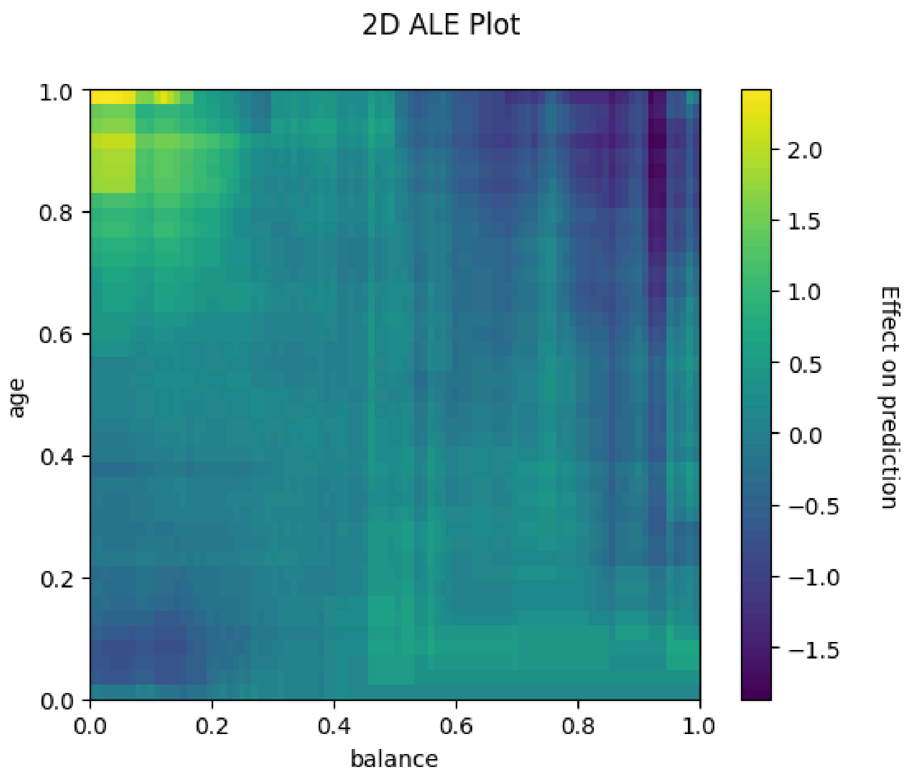
- The middle graph shows partial dependence against 'Average Balance', with values ranging from about 0.07 to about 0.12 for equilibrium. We see that as the balance increases, so does the partial dependence, until it "stabilizes" at 0.1.

- The last graph on the right shows how the average residual with age affects the partial dependence. We notice that if age < 0.15 (approximately) and balance is in the interval [0.1, 0.12] the partial dependence is higher (thus the effect on the model's prediction is greater).

Required B


1D ALE Plot - Discrete/Categorical
Effect on prediction (centered)
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Size
2000
1750
1500
1250
1000
750
500
250
0
age


1D ALE Plot - Continuous
95% confidence interval
Effect on prediction (centered)
0.06
0.04
0.02
0.00
−0.02
0.0
0.2
0.4
0.6
0.8
1.0
balance

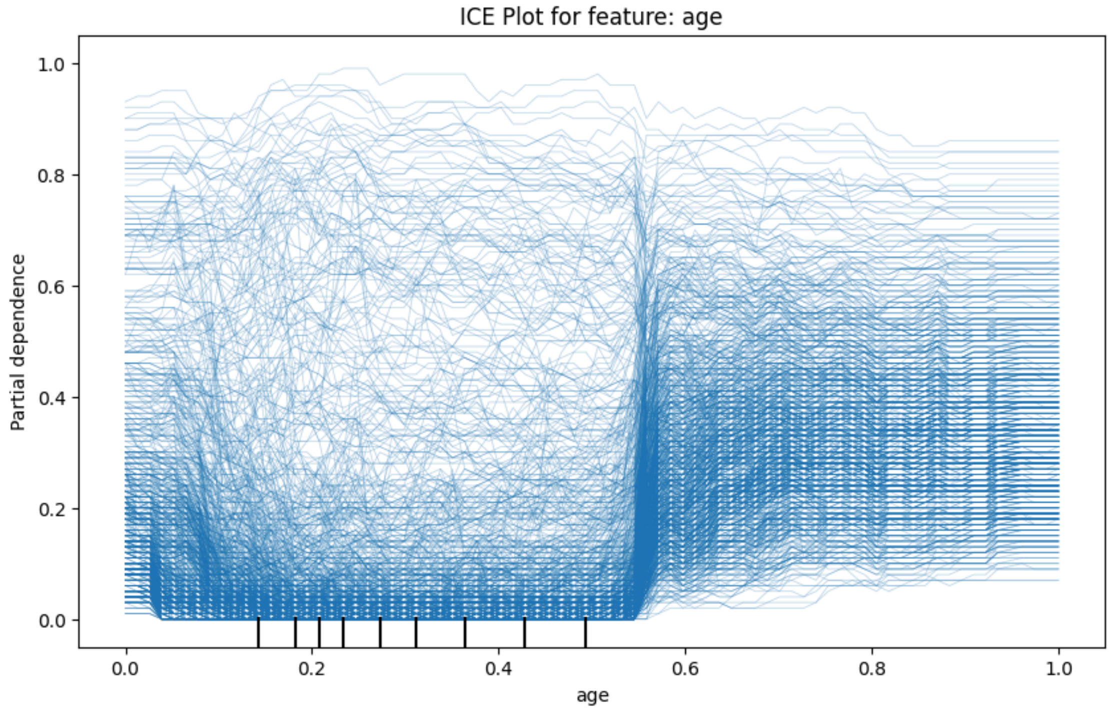


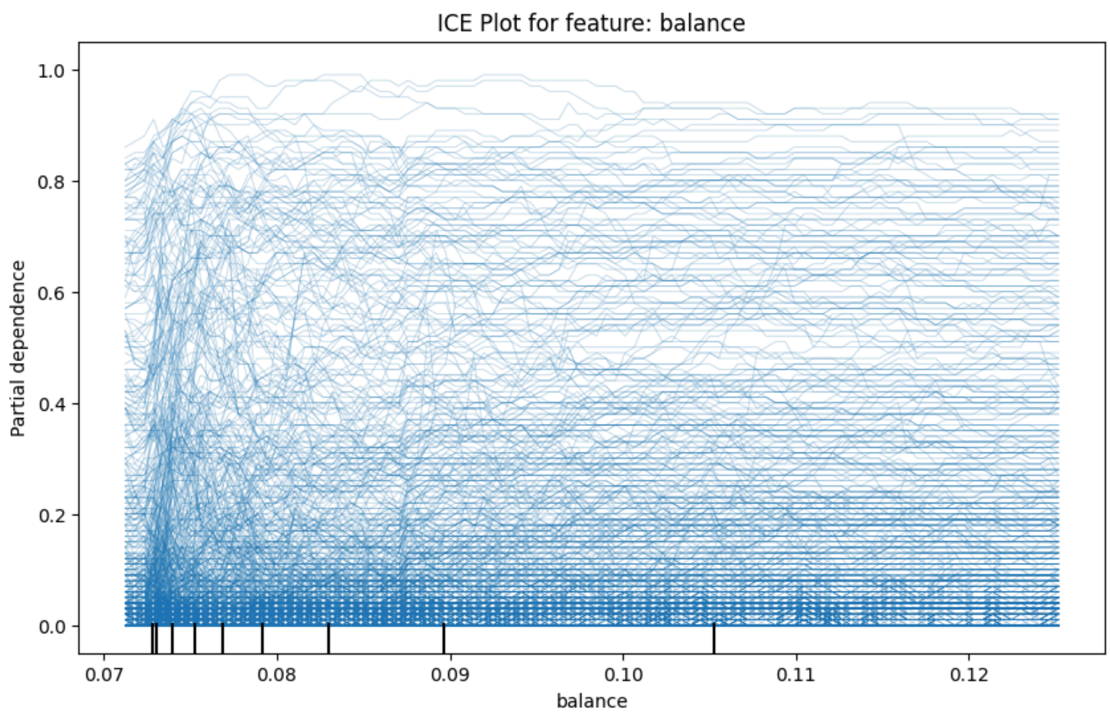
- In the first graph, we notice that if age is (roughly) in [0.8, 1] it has a high effect on predictions.

- In the second graph, we see that the higher the balance, the greater the effect it has on predictions (but generally it has a lower effect than 'age').

- Combining balance, prediction in the last graph we notice that the biggest effect on prediction is at the values [0, 0.2] for balance and [0.8, 1] for age.

Required C



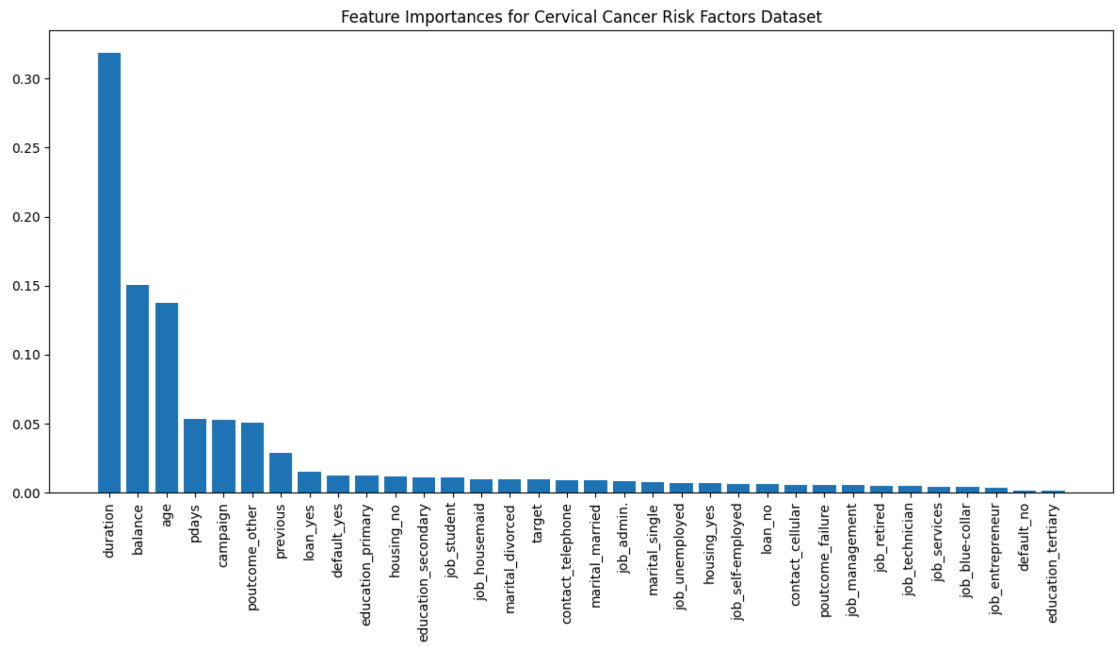


- ICE charts represent one line per display showing how the prediction changes when an attribute changes.

- In the first diagram we observe a general increase in the correlation of age when it exceeds 0.55, while when it is less than 0.55 it is comparatively low.

- In the second diagram we notice that the balance has a consistently positive and slightly increasing correlation with the predicted value as we do not encounter any sharp change in Dependence during the change in the value of the balance.

Required D



We notice that duration is the most important feature (according to the importance metric), followed by average balance and age. The rest of the features are of relatively low importance.

Question Q:

From the questions A, B, C we see overall the correlations between age, balance as well as the effect they have on the predictions of the model.

From the diagrams we can draw the following conclusions / observations:

1. Balance and age have an "inverse" relationship in the diagrams of A and B. Specifically, in A the partial dependence is greater when balance is large and age is small. On the contrary, in B there is a greater effect on the prediction when the age is large and the balance is small.
2. Diagram C "combines" the results of A and B, as in it as age "grows", the partial dependence increases (like B) and as the balance increases, the partial dependence also increases (like A) .

Description 3

Code

First, we proceed to normalize the values in the range from 0 to 1 for optimal results. Next, we copy the dataset from Colab, load the dataset, and normalize the values to the range 0 to 1 to optimize the results. Next, we run the code mentioned in the third description.

Wanted

Required A

```
Surrogate Decision Tree Accuracy: 69.51% |--- rm
<= 0.63
|   |--- lstat <= 0.35
|   | |--- lstat <= 0.09
|   |   |   |--- ptratio <= 0.76
|   |   |   | |--- rm <= 0.60
|   |   |   | | |--- value: [0.49]
|   |   |   | |--- rm > 0.60
|   |   |   | | |--- value: [0.58]
|   |   |   |--- ptratio > 0.76
|   |   |   | |--- lstat <= 0.05
|   |   |   | | |--- value: [0.85]
|   |   |   | |--- lstat > 0.05
|   |   |   | | |--- value: [0.91]
|   |   |--- lstat > 0.09
|   |   | |--- lstat <= 0.22
|   |   | | |--- rm <= 0.49
|   |   | | | |--- value: [0.37]
|   |   | | |--- rm > 0.49
|   |   | | | |--- value: [0.45]
|   |   | |--- lstat > 0.22
|   |   | | |--- indus <= 0.09
|   |   | | | |--- value: [0.46]
|   |   | | |--- indus > 0.09
|   |   | | | |--- value: [0.35]
|   |--- lstat > 0.35
|   | |--- crim <= 0.06
|   |   |   |--- nox <= 0.30
```

```
|   |   |   |   |--- nox <= 0.18
|   |   |   |   |  |--- value: [0.28]
|   |   |   |   |--- nox > 0.18
|   |   |   |   |  |--- value: [0.35]
|   |   |   |--- nox > 0.30
|   |   |   |  |--- lstat <= 0.45
|   |   |   |  |  |--- value: [0.28]
|   |   |   |  |--- lstat > 0.45
|   |   |   |  |  |--- value: [0.22]
|   |   |--- crim > 0.06
|   |   |  |--- lstat <= 0.51
|   |   |  |  |--- nox <= 0.41
|   |   |  |  |  |--- value: [0.27]
|   |   |  |  |--- nox > 0.41
|   |   |  |  |  |--- value: [0.20]
|   |   |  |--- lstat > 0.51
|   |   |  |  |--- nox <= 0.60
|   |   |  |  |  |--- value: [0.18]
|   |   |  |  |--- nox > 0.60
|   |   |  |  |  |--- value: [0.10]
|--- rm >    0.63
|   |--- rm <= 0.74
|   |  |--- nox <= 0.56
|   |   |   |--- lstat <= 0.10
|   |   |   |  |--- nox <= 0.48
|   |   |   |  |  |--- value: [0.65]
|   |   |   |  |--- nox > 0.48
|   |   |   |  |  |--- value: [0.89]
|   |   |   |--- lstat > 0.10
|   |   |   |  |--- rm <= 0.67
|   |   |   |  |  |--- value: [0.50]
|   |   |   |  |--- rm > 0.67
|   |   |   |  |  |--- value: [0.63]
|   |   |--- nox > 0.56
|   |   |  |--- rm <= 0.69
|   |   |  |  |--- value: [0.20]
|   |   |  |--- rm > 0.69
|   |   |  |  |--- dis <= 0.07
|   |   |  |  |  |--- value: [0.35]
```

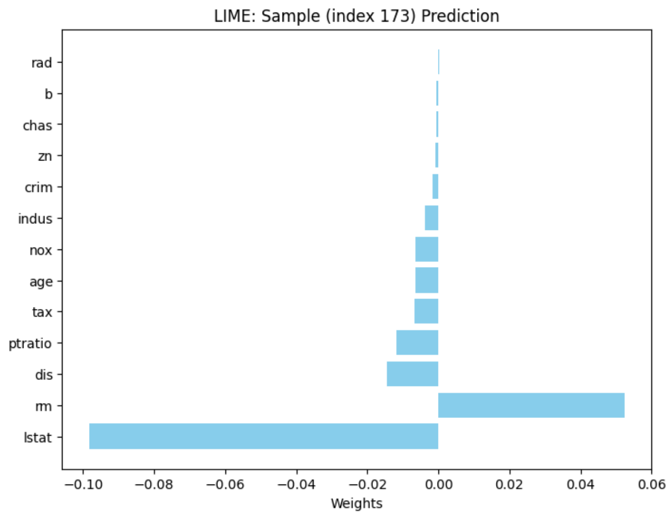
```
|   |   |   |   |--- dis > 0.07
|   |   |   |   |   |--- value: [0.29]
|   |--- rm > 0.74
|   |   |--- ptratio <= 0.61
|   |   |   |--- ptratio <= 0.30
|   |   |   |   |--- rm <= 0.82
|   |   |   |   |   |--- value: [0.89]
|   |   |   |   |--- rm > 0.82
|   |   |   |   |   |--- value: [0.98]
|   |   |   |--- ptratio > 0.30
|   |   |   |   |--- age <= 0.62
|   |   |   |   |   |--- value: [0.88]
|   |   |   |   |--- age > 0.62
|   |   |   |   |   |--- value: [0.82]
|   |   |--- ptratio > 0.61
|   |   |   |--- indus <= 0.44
|   |   |   |   |--- value: [0.74]
|   |   |   |--- indus > 0.44
|   |   |   |   |--- value: [0.51]
```

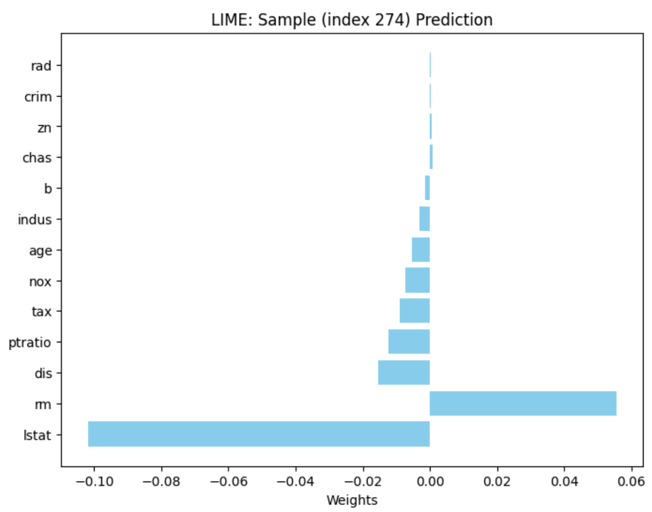
We notice that the characteristic RM (Average number of rooms per dwelling) is one of the basic characteristics and significantly determines the property's value predictions, as it is found in many nodes. Specifically, low values of 'RM' (<0.63) seem to be associated with low values of the target (house price), while higher values of 'RM' (> 0.74) are associated with higher house prices.

The attribute LSTAT' (percentage of low-income residents) is also quite an important factor, and we observe that low percentages are associated with higher predicted house prices, while high percentages (> 0.35) are associated with lower ones.

Finally, the CRIM (crime ratio) trait also seems to have some significance as low values are associated with higher predicted values.

Required B





For the 2 selected samples we observe that the LSTAT characteristic has a negative correlation with the price of the houses. Therefore the higher the value of LSTAT, the lower we expect the price of the house to be and similarly the lower the value of LSTAT we expect the price of the house to be high.

Regarding the RM attribute, we observe that it has a positive correlation with the price of the house. That is, when the RM price is high we expect the house price to be high and when the RM price is low we expect the house price to be equally low.

Finally, we see that the CRIM characteristic has a slight negative correlation but mainly does not significantly affect the price of a house as according to the diagram its correlation is close to 0.

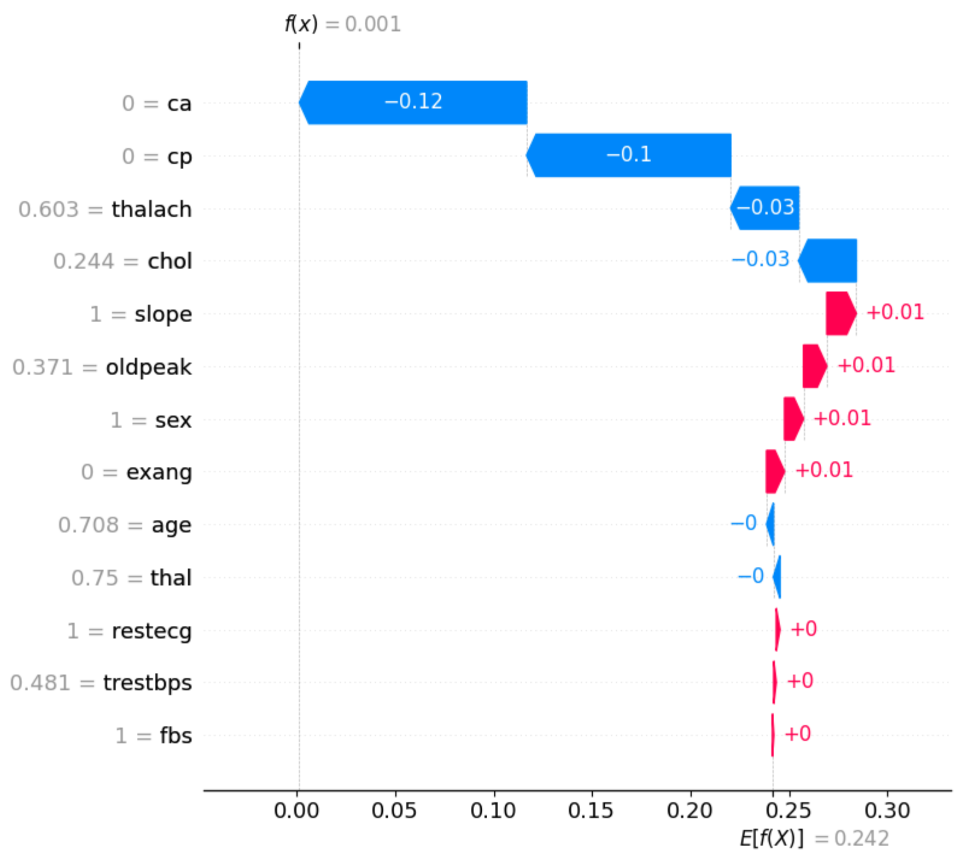
Description 4

Code

First, the required libraries are imported. We then proceed to load the dataset as described in the dataset manual (import in python). At this stage, we remove the nan values   and proceed to normalize the data. Next, we train and initialize SHAP (SHapley Additive exPlanations).

Wanted

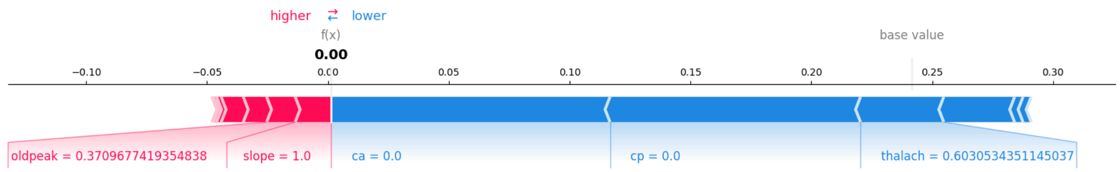
Question A:



We notice that,**for this particular sample**, traits ca, cp, thalach, chol, thal and age (blue bars) have a negative effect on prediction, while traits such as slope, oldpeak, sex, exang, restecg, trestbps and fbs (red bars) have a positive effect.

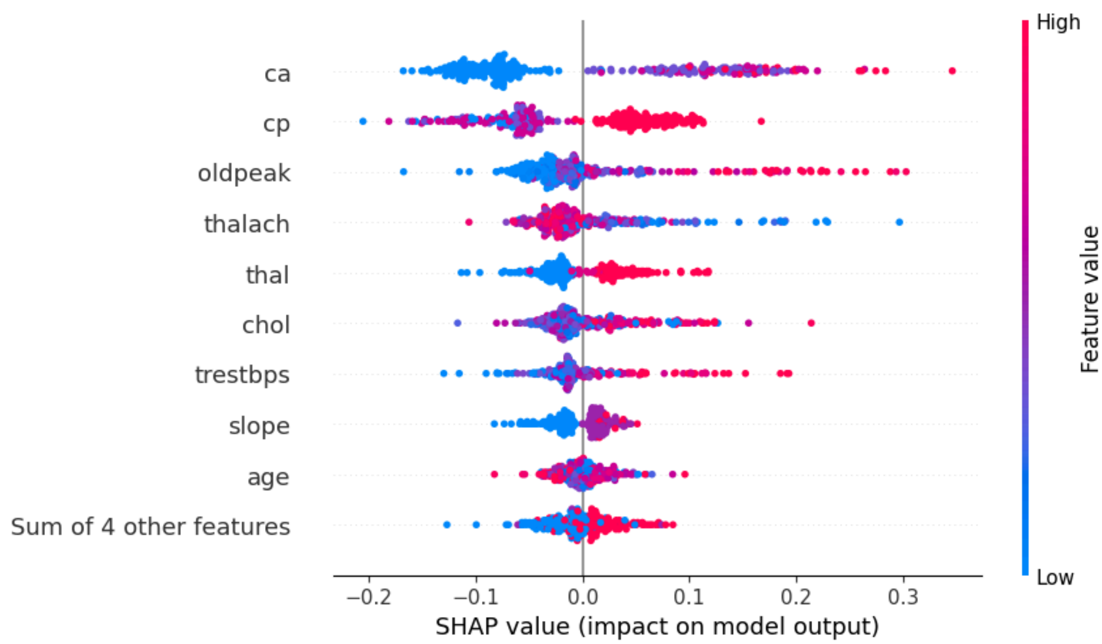
Additionally, the trait ca has the largest negative effect (followed by cp and then thalach), while the trait slope has the largest positive effect (followed by oldpeak).

Question B:



We observe that the features "oldpeak" and "slope" increase the prediction (relative to the average prediction), while features such as "ca", "cp" and "thalach" decrease the prediction**for this particular sample**.

Required C:

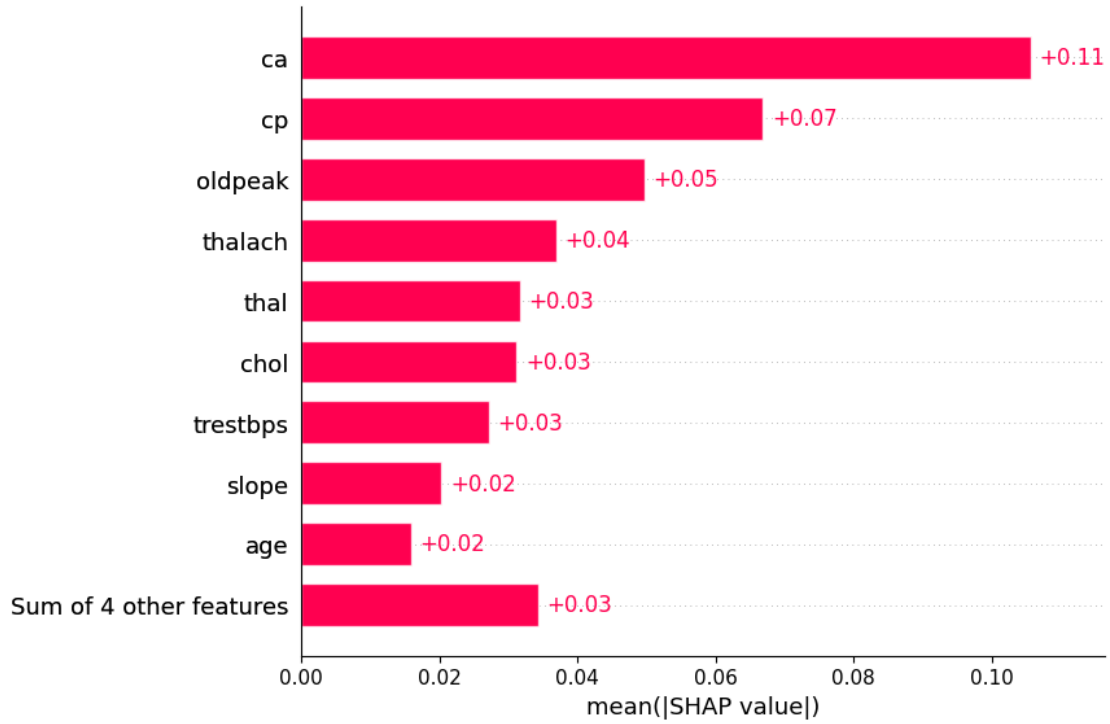


We observe that features such as ca, cp, oldpeak, thalach have a greater impact on prediction.

Also, we see that a low number of ca reduces the prediction result, while as ca increases, the result (of the prediction) also increases. The cp, oldpeak attributes have a similar behavior.

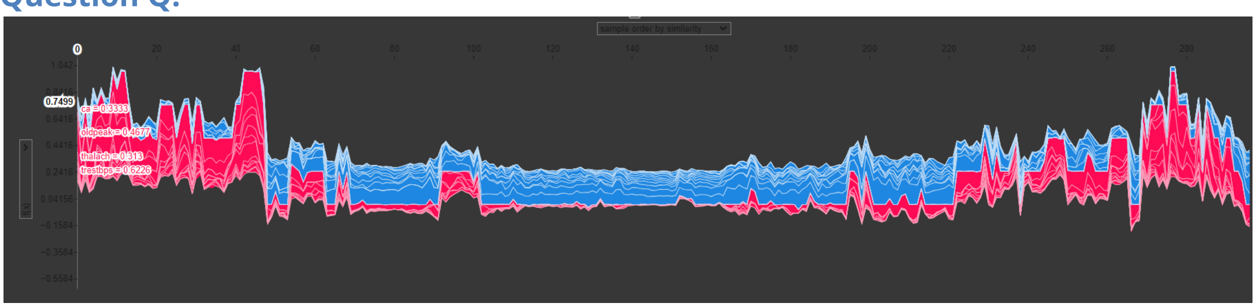
On the other hand, the trait thalach as it decreases increases the prediction effect, while as it increases it decreases it (the effect).

Required D:

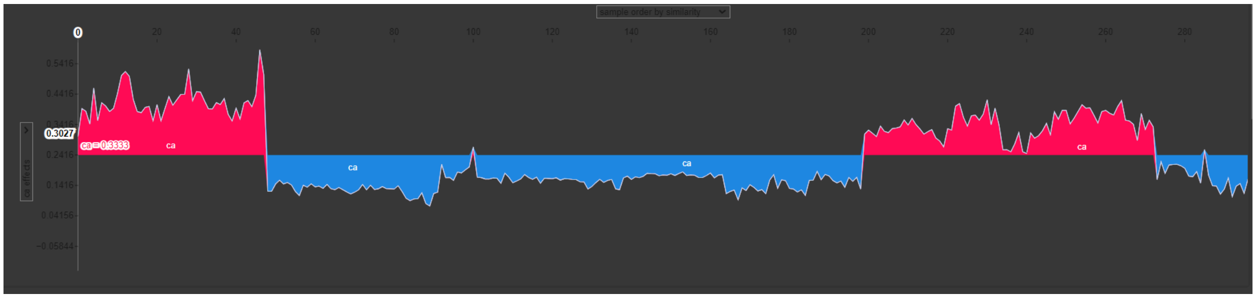


We observe that features such as ca, cp have the largest effects on the mean prediction value (by 11% and 7% respectively). Then, follows the oldpeak with an impact (on the average forecast value) of 5%. The remaining features have an impact of less than or equal to 4%.

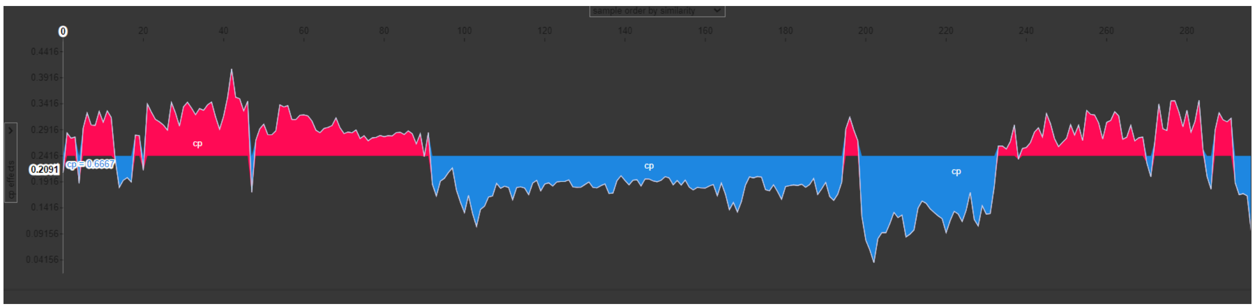
Question Q:



y = f(x)



y = ca effects



y = cp effects

Remarks:

Looking at the graph above (y axis = f(x) and x axis = sample order by similarity) and comparing it with y axis = ca effects, we see that ca has a big impact on the prediction (since when it increases or decreases, many times the forecast changes in a similar way).

Almost the same (with less effect than ca) we can also observe for cp (if y axis = cp effects).

Explanations

These observations are "justified" as from the question D we saw that ca, cp had the highest effect on the average prediction.

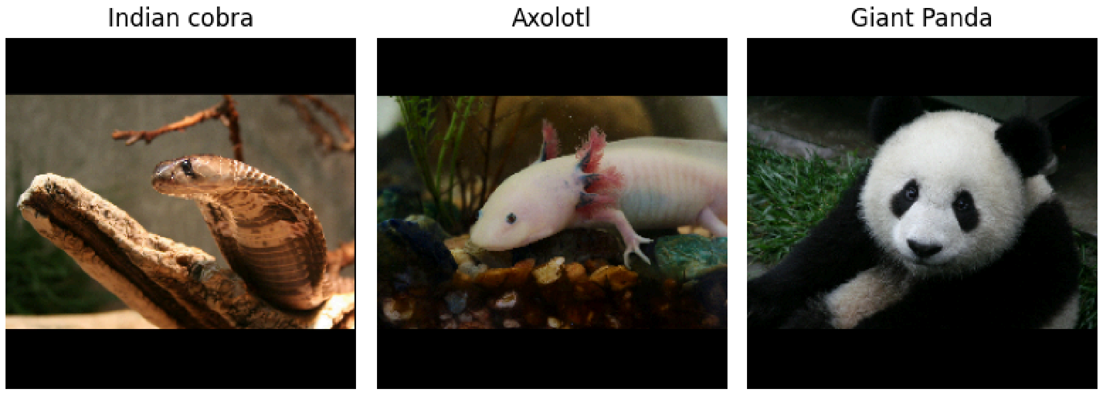
Examples:

Also, if we put x = original sample ordering and go to x = 0 (where this is the sample index of the requested A and B) we see similarities with the requested A and B. Specifically, we notice that the attribute values are very close to those of question B (if we change the y-axis to the values described in B
- oldpeak, slope, ca, cp, thalach).

As for the colors (blue = prediction decrease and red = increase) we see that the corresponding features have the same colors as those of A and B.

Description 5

Images used:



Question A:

```
Model: "sequential_2"
_________________________________________________________________ Layer (type)
                                        Output Shape              Param#
================================================================= ===============
 inception_v1 (KerasLayer) (None, 1001)                           6633209

================================================================= ===============
Total params: 6633209 (25.30 MB)
Trainable params: 0 (0.00 Byte)
Non-trainable params: 6633209 (25.30 MB)
_________________________________________________________________
```

We notice how in Inception_v1 the neural network correctly recognizes the axolotl with 100% accuracy. But for the rest, it doesn't express that much certainty regarding the recognition of the images. The cobra is recognized with a success rate of 68.4%, while the panda, the network is quite confident with a success rate of 89.4%.

```
Model: "sequential_4"
_________________________________________________________________ Layer (type)
                                        Output Shape              Param#
================================================================= ===============
 inception_v2 (KerasLayer) (None, 1001)                           11199137
```

```
========================================================== ===============
Total params: 11199137 (42.72 MB)
Trainable params: 0 (0.00 Byte)
Non-trainable params: 11199137 (42.72 MB)
__________________________________________________________________
```

As far as Inception_v2 is concerned, the neural network expresses a much higher certainty regarding the recognition of images compared to inception_v1. Cobra is recognized with a success rate of 91.7%, which is much better than inception_v1, while for panda, the network is again quite confident with a success rate of 93.6%. Conversely for axolotl, it is not recognized as inception_v1 (100%) and the prediction dropped to 94.2%.

```
Model: "sequential_5"
__________________________________________________________________ Layer (type)
                                        Output Shape                  Param#
========================================================== ===============
 inception_v3 (KerasLayer) (None, 1001)                               23853833

========================================================== ===============
Total params: 23853833 (91.00 MB)
Trainable params: 0 (0.00 Byte)
Non-trainable params: 23853833 (91.00 MB)
__________________________________________________________________
```

In Inception_v3 we observe a significant increase in certainty compared to the other 2 versions. Initially it recognizes the cobra almost perfectly with 99.6% (much better than the previous ones) while the axolotl and the panda with 100% success.

```
Model: "sequential_6"
__________________________________________________________________ Layer (type)
                                        Output Shape                  Param#
========================================================== ===============
 inception_resnet_v2 (Keras (None, 1001)                              55875273
 layers)

========================================================== ===============
Total params: 55875273 (213.15 MB)
Trainable params: 0 (0.00 Byte)
```

```
Non-trainable params: 55875273 (213.15 MB)
_________________________________________________________________
```

In Inception_resnet_v2, we see the highest success rates so far, as all animals are correctly identified with a "confidence" rate of 100%.

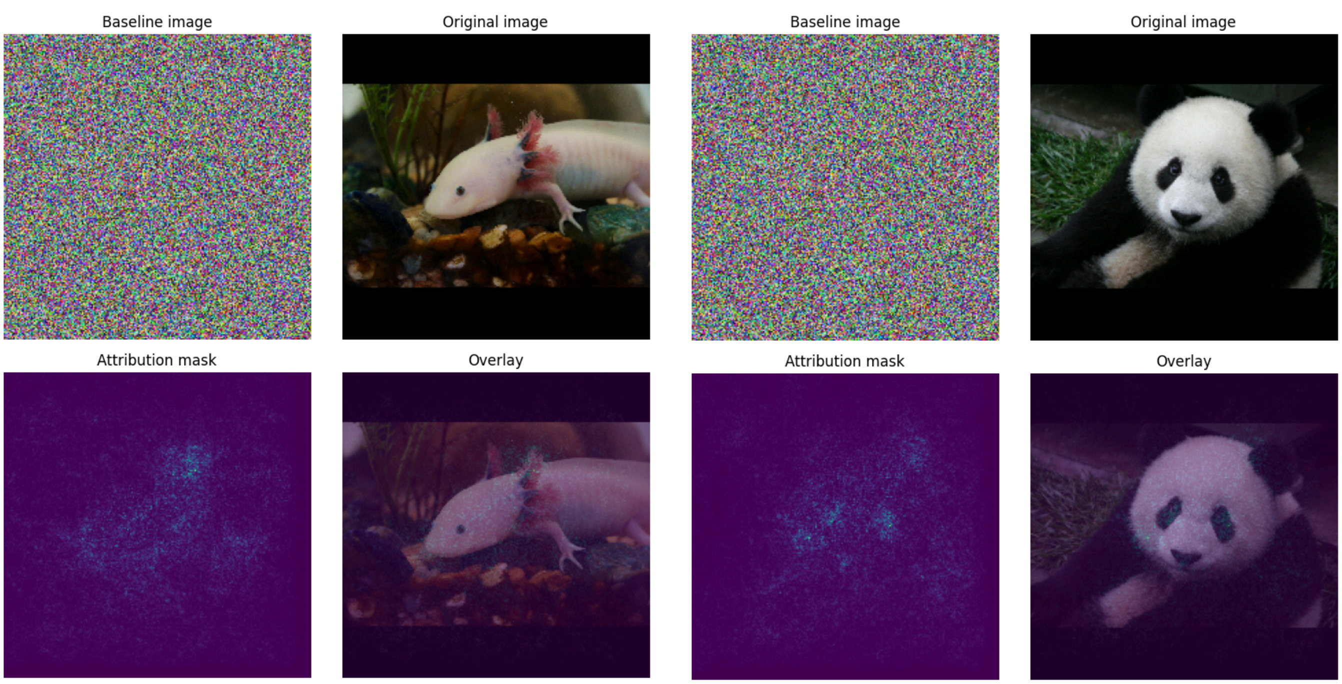
Question B:

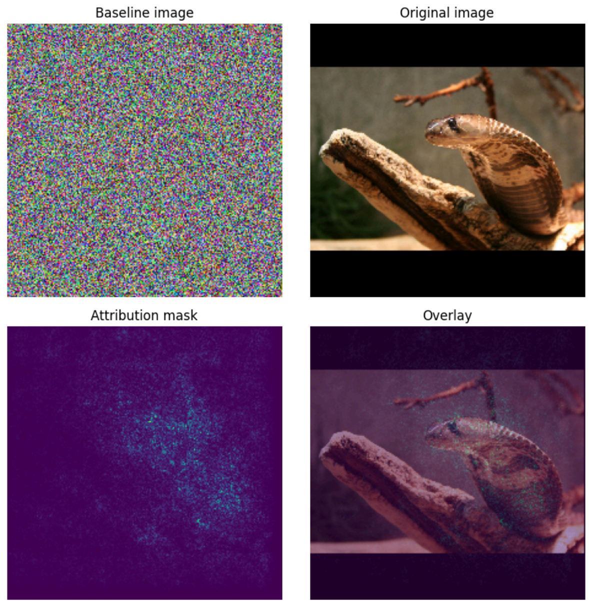
We notice that the best result (classification) for each image is inception_resnet_v2, so we choose it. Here is the test with baseline random:

```
Model: "sequential_9"
_________________________________________________________________ Layer (type)
                                  Output Shape              Param#
================================================================= ===============
  inception_resnet_v2 (Keras (None, 1001)                    55875273
  layers)

================================================================= ===============
Total params: 55875273 (213.15 MB)
Trainable params: 0 (0.00 Byte)
Non-trainable params: 55875273 (213.15 MB)
_________________________________________________________________
```

Results:





Using the random baseline, we notice that the model emphasizes different pixels (highlighted in light color in the attribution mask) with the result that the generated explanations (attribution mask, overlay) have more "noise" than before when we used " black" baseline.

Only in the attribution mask for the giant panda do we see that the model has emphasized pixels that highlight the animal's features (such as the face and eyes).

In the image of the axolotl, on the other hand, looking at the overlay, we notice that the model has given importance in a general way to pixels on the animal's body, but without emphasizing any of its particular characteristics (as when we used the black baseline in inception_resnet_v2, which gave importance in features such as the fins - legs of the axolotl). Similar behavior can be observed for the image of the Indian cobra.

You can see all the colab codehere .