AI CAMP

school of ai Algiers

Hello and Welcome to AI Camp

AI CAMP

school of ai Algiers

XGBOOST

EXTREME GRADIENT BOOSTING WITH XGBOOST

HARIS BEL.
TURKI

AI CAMP

school of ai Algiers

LET'S TALK ABOUT ENSAMBLE TECHNIQUES FIRST

ENSAMBLE TECHNIQUES :

- **BAGGING :**
  - Random forest classifier
  - Random forest regressor
- **BOOSTING :**
  - Adaboost
  - Gradient boosting
  - XGBoost

BOOSTING :

- **WHAT IS BOOSTING :**
  - **Ensemble sequential meta-algorithm used to convert many weak learners into a strong learner**
- **HOW BOOSTING WORKS :**

BOOSTING :

- **WHAT IS BOOSTING :**
  - **Ensemble sequential meta-algorithm used to convert many weak learners into a strong learner**
- **HOW BOOSTING WORKS :**

TRAINING DATA

BOOSTING :

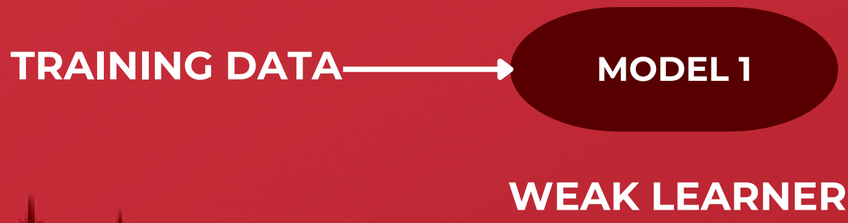
- **WHAT IS BOOSTING :**
  - **Ensemble sequential meta-algorithm used to convert many weak learners into a strong learner**
- **HOW BOOSTING WORKS :**

AI CAMP

school of ai Algiers

BOOSTING :

- **WHAT IS BOOSTING :**
  - **Ensemble sequential meta-algorithm used to convert many weak learners into a strong learner**
- **HOW BOOSTING WORKS :**

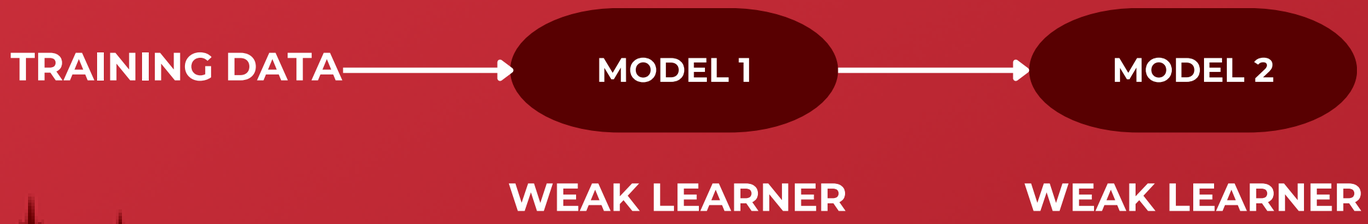


AI CAMP

school of ai Algiers

BOOSTING :

- **WHAT IS BOOSTING :**
  - **Ensemble sequential meta-algorithm used to convert many weak learners into a strong learner**
- **HOW BOOSTING WORKS :**

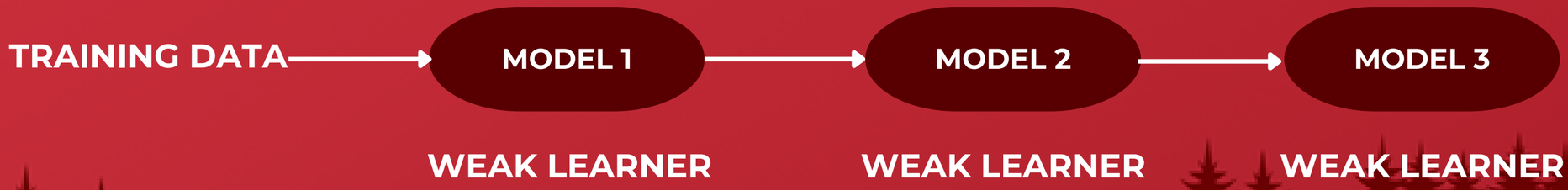


AI CAMP

school of ai Algiers

BOOSTING :

- **WHAT IS BOOSTING :**
  - **Ensemble sequential meta-algorithm used to convert many weak learners into a strong learner**
- **HOW BOOSTING WORKS :**

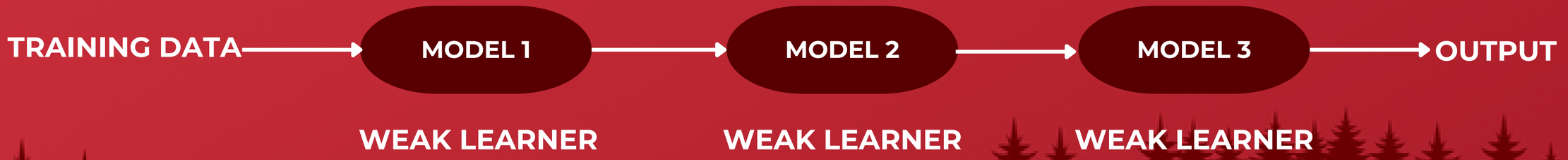


AI CAMP

school of ai Algiers

BOOSTING :

- **WHAT IS BOOSTING :**
  - **Ensemble sequential meta-algorithm used to convert many weak learners into a strong learner**
- **HOW BOOSTING WORKS :**

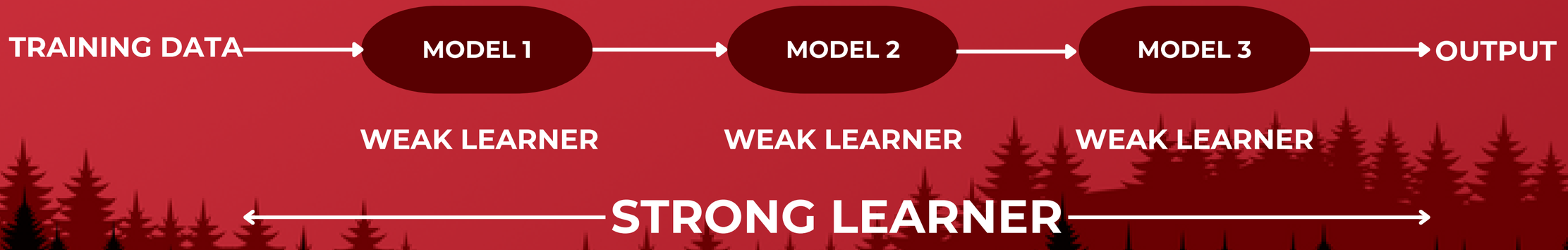


AI CAMP

school of ai Algiers

BOOSTING :

- **WHAT IS BOOSTING :**
  - **Ensemble sequential meta-algorithm used to convert many weak learners into a strong learner**
- **HOW BOOSTING WORKS :**



AI CAMP

school of ai Algiers

WEAK LEARNERS AND STRONG LEARNERS :

- **WEAK LEARNER :**
  - ML algorithm that is slightly better than chance
  - Slightly better than chance means we are always going to have an error rate which is less than 1/2
- **STRONG LEARNER :**
  - Any algorithm that can be tuned to achieve good performance

AI CAMP

school of ai Algiers

LET'S TALK ABOUT XGBOOST!

EXTREME GRADIENT BOOSTING WITH XGBOOST

WHAT IS XGBOOST (eXtreme Gradient Boosting) :

- Open source implementation of boosted trees
- Optimized gradient-boosting machine learning library
- Originally written in C++
- Has API’s in several languages :
  - Python
  - R
  - Java

AI CAMP

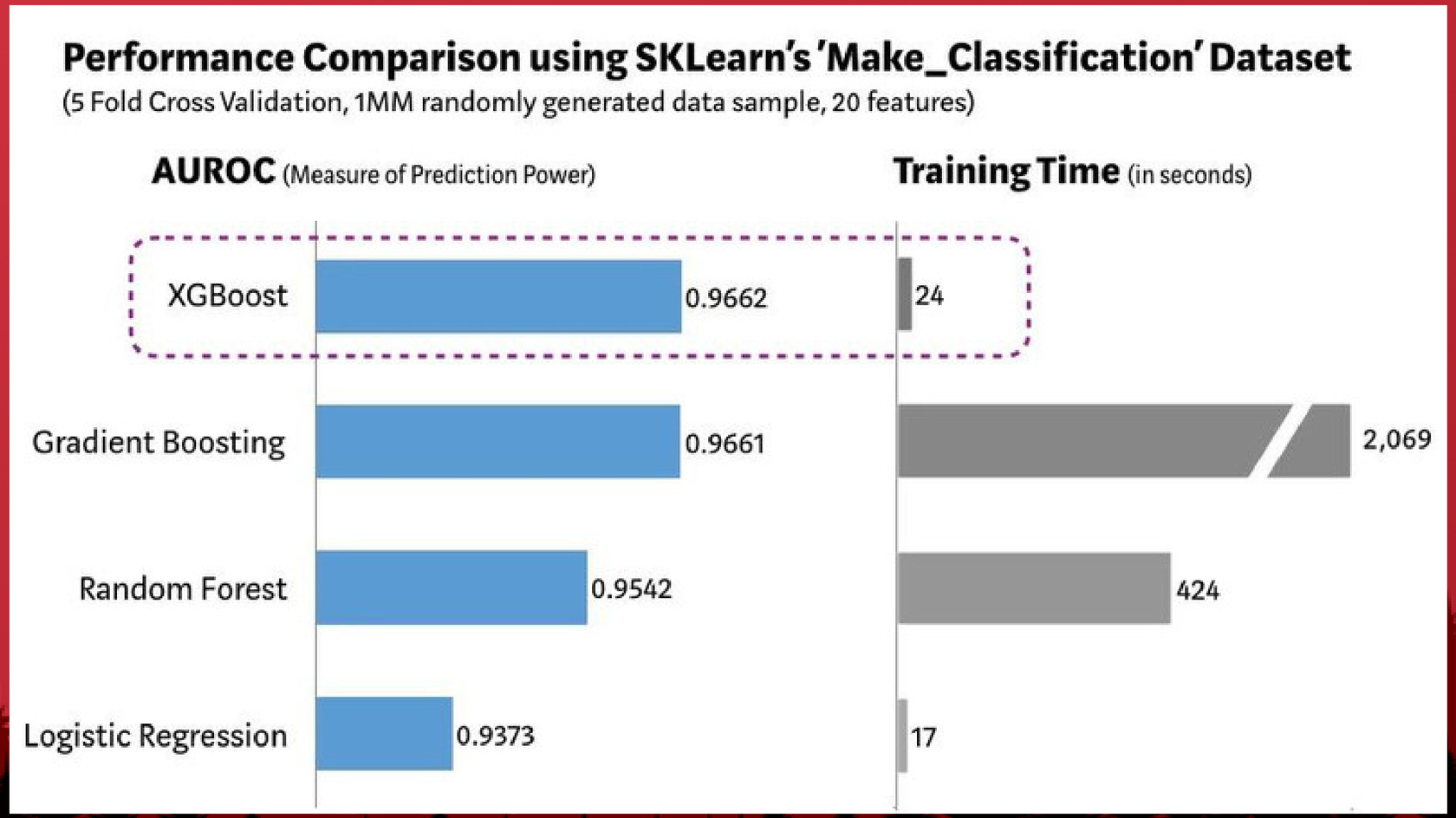
school of ai Algiers

WHAT MAKES XGBOOST SO POPULAR :

- Fast & Efficient Implementation
- Speed and performance
- Core algorithm is parallelizable
- Consistently outperforms single-algorithm methods
- State-of-the-art performance in many ML tasks
- Built in regularization to prevent overfitting
- Highly competitive algorithm for machine learning competitions (eg: kaggle competitions)

WHAT MAKES XGBOOST SO POPULAR :

- **Comparison between XGBoost and other models :**



AI CAMP

school of ai Algiers

WHAT MAKES XGBOOST SO POPULAR :

- **Comparison between XGBoost and Keras :**

Category	XGBoost	Keras Deep Learning	Winner?
Performance on test set			
Accuracy	80.1%	78.1%	XGBoost
recall: true positive / (true positive + false negative)	0.89	0.68	
false negatives	1,200	3,500	
Training time	1 minute 24 seconds	2 minutes – 3 minutes for 50 epoch experiment depending on hw env and call back patience setting	Inconclusive – deep learning training time varies
Code complexity	• Extra steps required to transform data coming out of pipeline • 1 line to build model	• Data from pipeline ready to train model • Complex model build	Inconclusive
Flexibility	Handles continuous & categorical columns	Handles continuous, categorical and text columns	Deep learning

XGBOOST STEPS :

- XGBoost is an implementation of gradient boosting that goes through cycles to iteratively add models into an ensemble

XGBOOST STEPS :

- XGBoost is an implementation of gradient boosting that goes through cycles to iteratively add models into an ensemble
- It begins by initializing the ensemble with a single model, whose predictions can be pretty naive. (even if its predictions are wildly inaccurate, subsequent additions to the ensemble will address those errors.)

BASE / INITIAL MODEL

XGBOOST STEPS :

- Then, we start the cycle:
- First, we use the current ensemble to generate predictions for each observation in the dataset. To make a prediction, we add the predictions from all models in the ensemble.

XGBOOST STEPS :

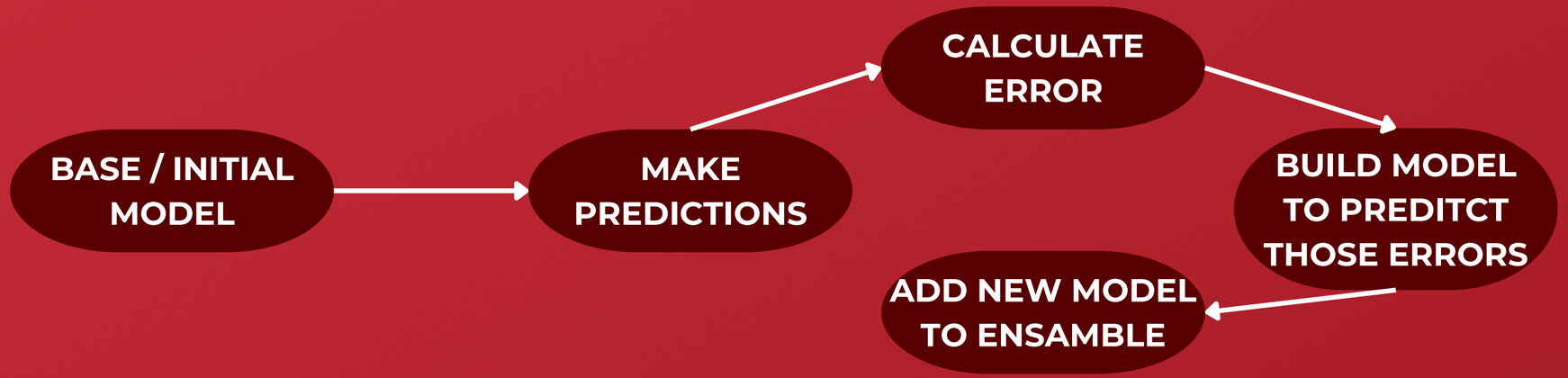
- Then, we start the cycle:
- First, we use the current ensemble to generate predictions for each observation in the dataset. To make a prediction, we add the predictions from all models in the ensemble.
- These predictions are used to calculate the errors(residuals) based on the previous model residuals for each observation in the dataset.

XGBOOST STEPS :

- **Then, we build a new model to predict those errors (residuals)**

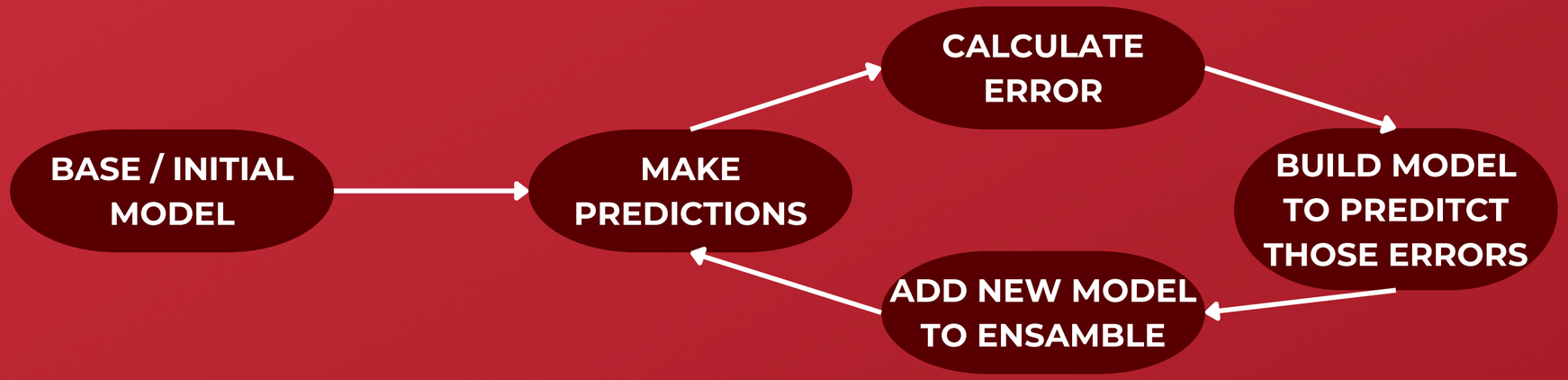
XGBOOST STEPS :

- Then, we build a new model to predict those errors (residuals)
- Finally, we add the new model to ensemble, and ...



XGBOOST STEPS :

- Then, we build a new model to predict those errors (residuals)
- Finally, we add the new model to ensemble, and ...
- ... repeat!



AI CAMP

school of ai Algiers

LET'S BEGIN USING XGBOOST!

EXTREME GRADIENT BOOSTING WITH XGBOOST

AI CAMP

school of ai Algiers

LET'S START WITH XGBOOST FOR CLASSIFICATION

EXTREME GRADIENT BOOSTING WITH XGBOOST

XGBOOST CLASSIFICATION :

- **Dataset :**

Pr =0.5

BASE / INITIAL MODEL

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	
<=50K	G	1	
<=50K	G	1	
>50K	B	0	
>50K	G	1	
>50K	N	1	
<=50K	N	0	

- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

Pr =0.5

BASE / INITIAL MODEL

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	
<=50K	G	1	
>50K	B	0	
>50K	G	1	
>50K	N	1	
<=50K	N	0	

- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

Pr =0.5

BASE / INITIAL MODEL

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5

- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

Pr =0.5 **BASE / INITIAL MODEL**

Salary

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5

- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5

Pr =0.5

BASE / INITIAL MODEL

Salary

- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5

Pr =0.5

BASE / INITIAL MODEL

Salary

-0.5 , 0.5 , 0.5 ... , -0.5

- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

school of ai Algiers

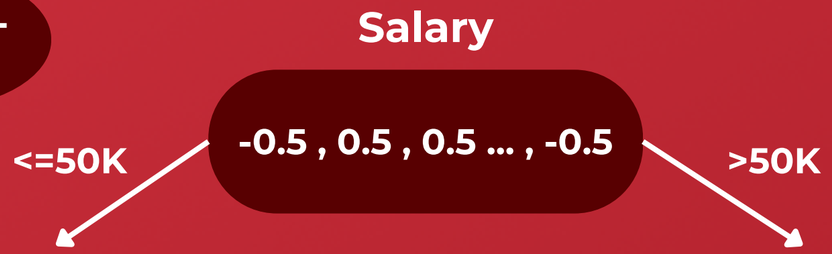
XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5

Pr =0.5

BASE / INITIAL MODEL



- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

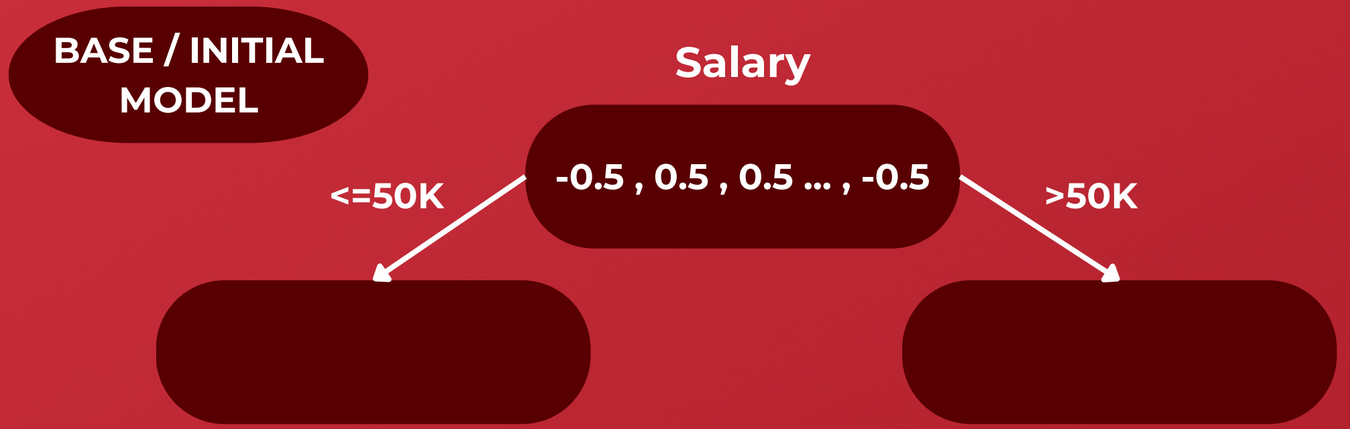
school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5

Pr =0.5



- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5

Pr =0.5

- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5

Pr =0.5

- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

school of ai Algiers

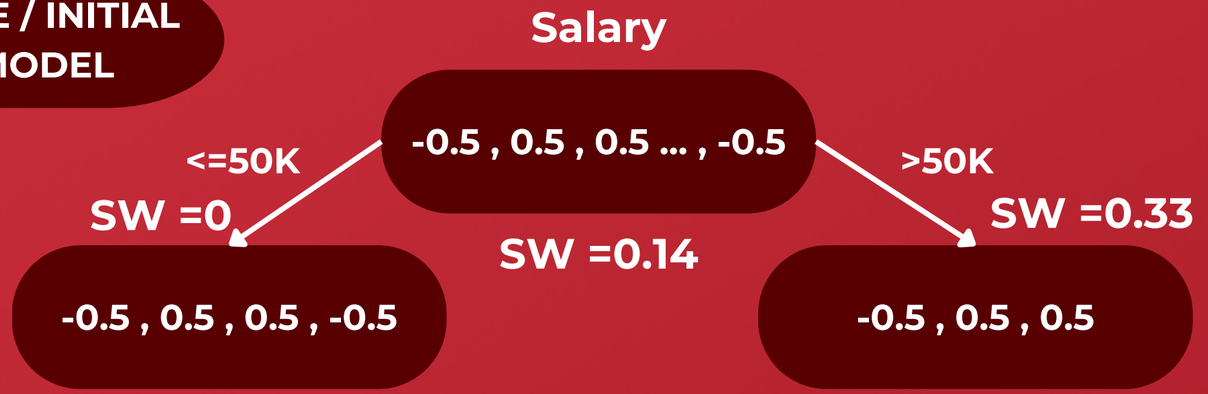
XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5

Pr =0.5

BASE / INITIAL MODEL



- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5

Pr =0.5

BASE / INITIAL MODEL

- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

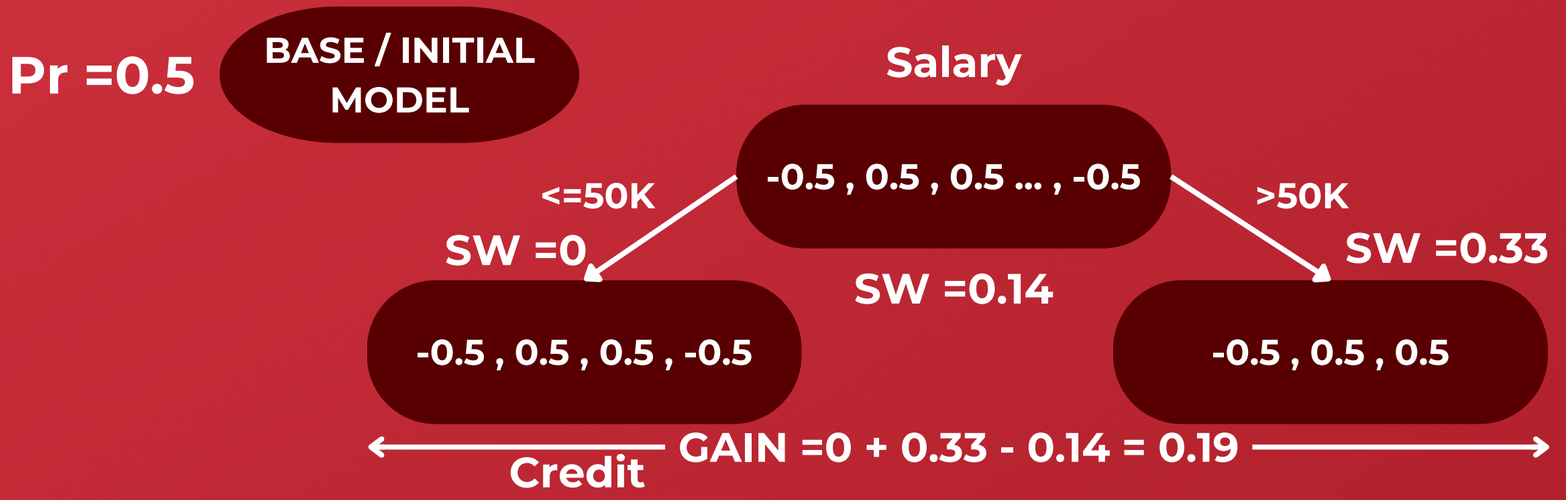
AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5



- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

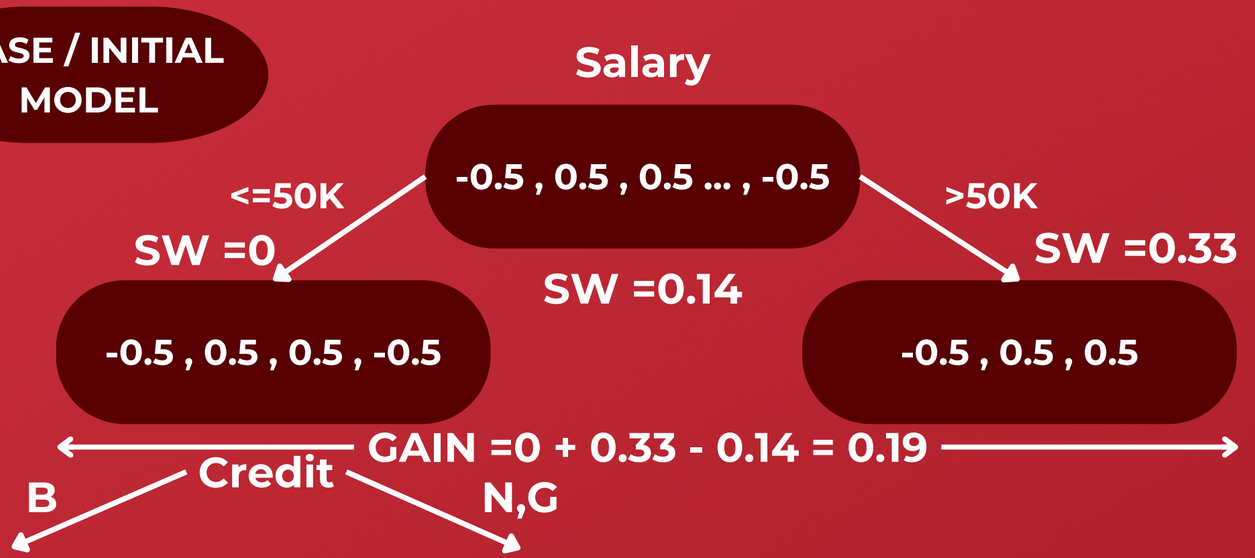
AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5



- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

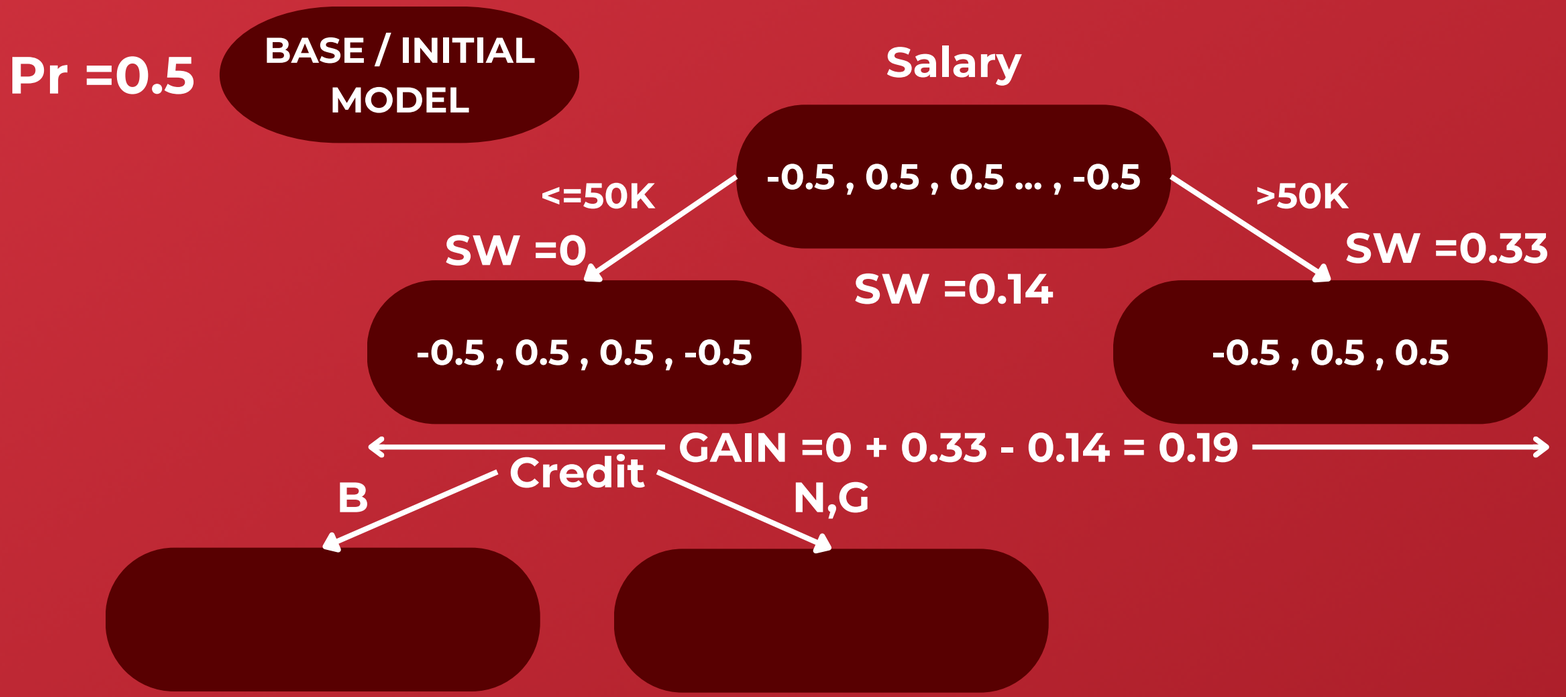
AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5



- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

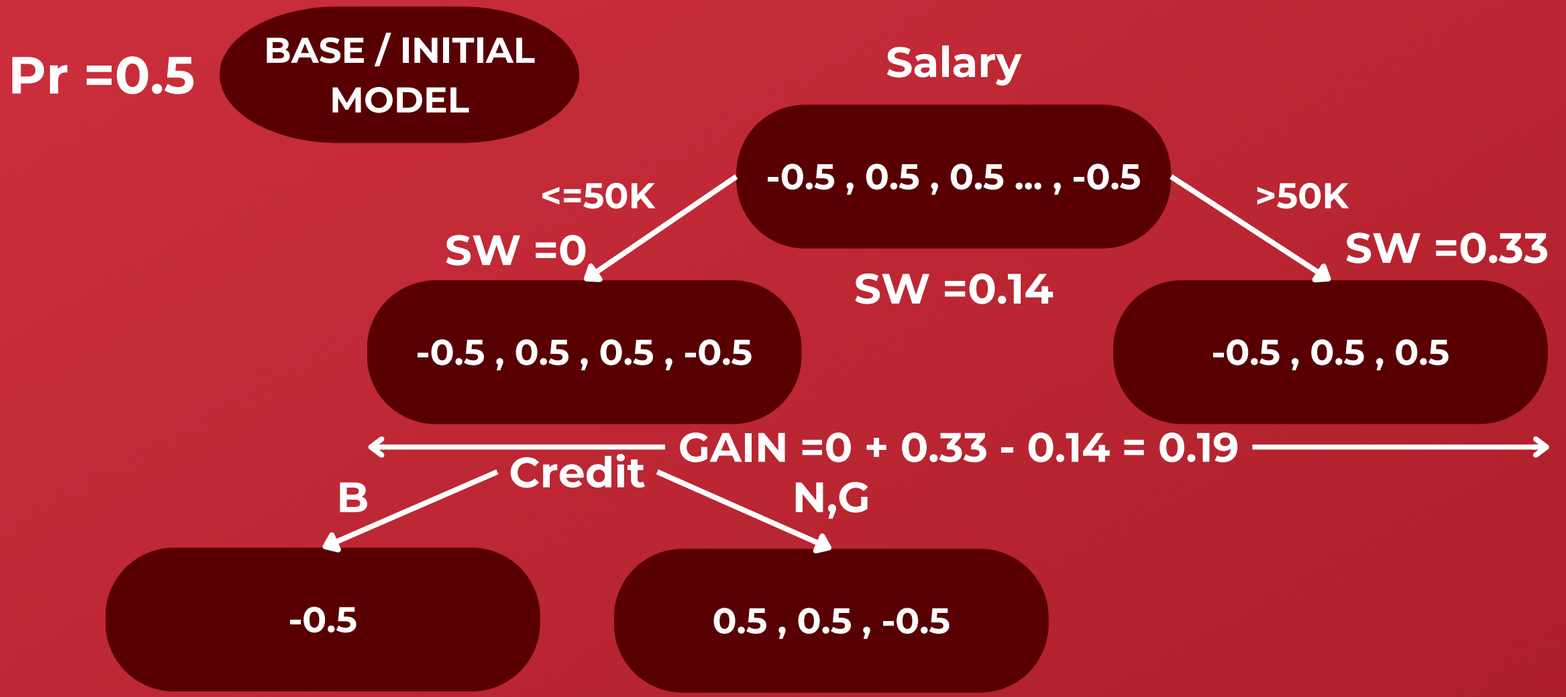
AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5



- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

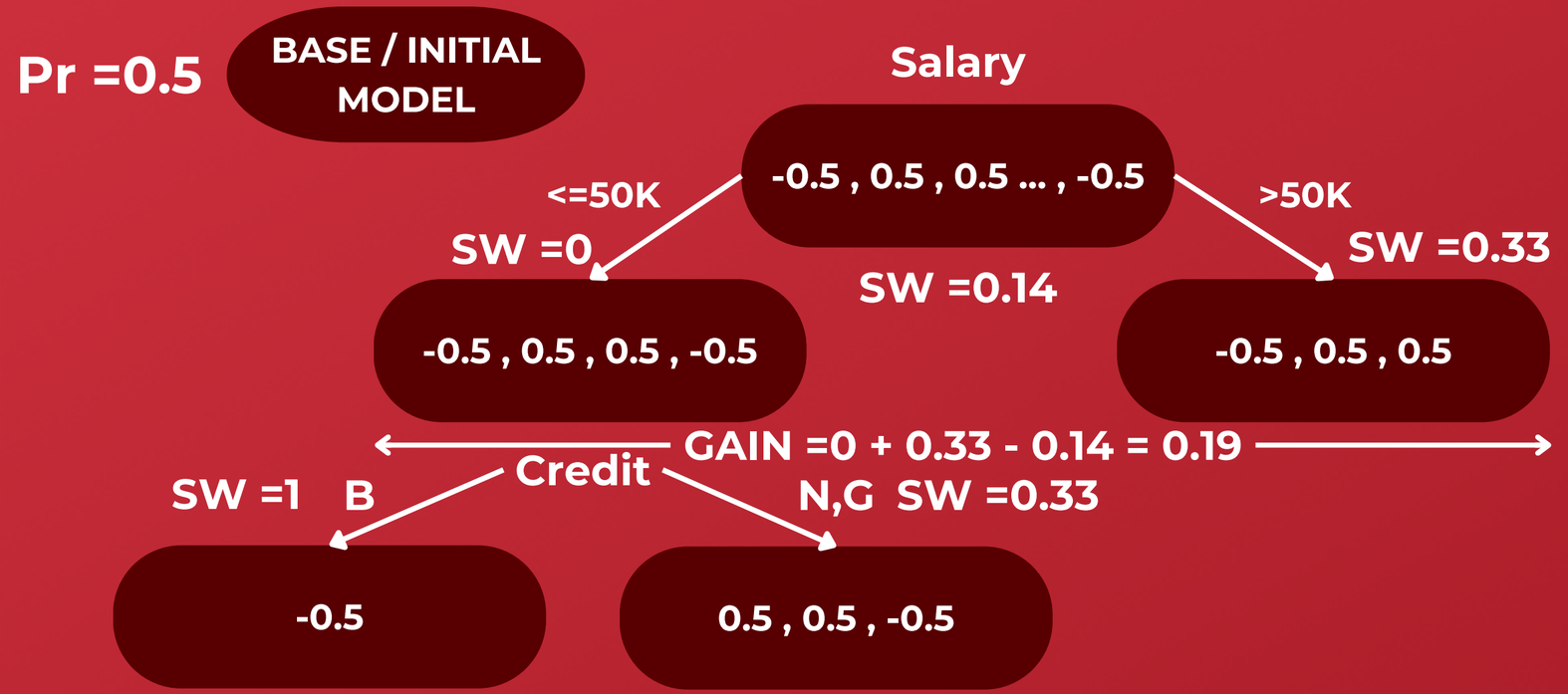
AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5



- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES=APROVAL-Pr , λ=0

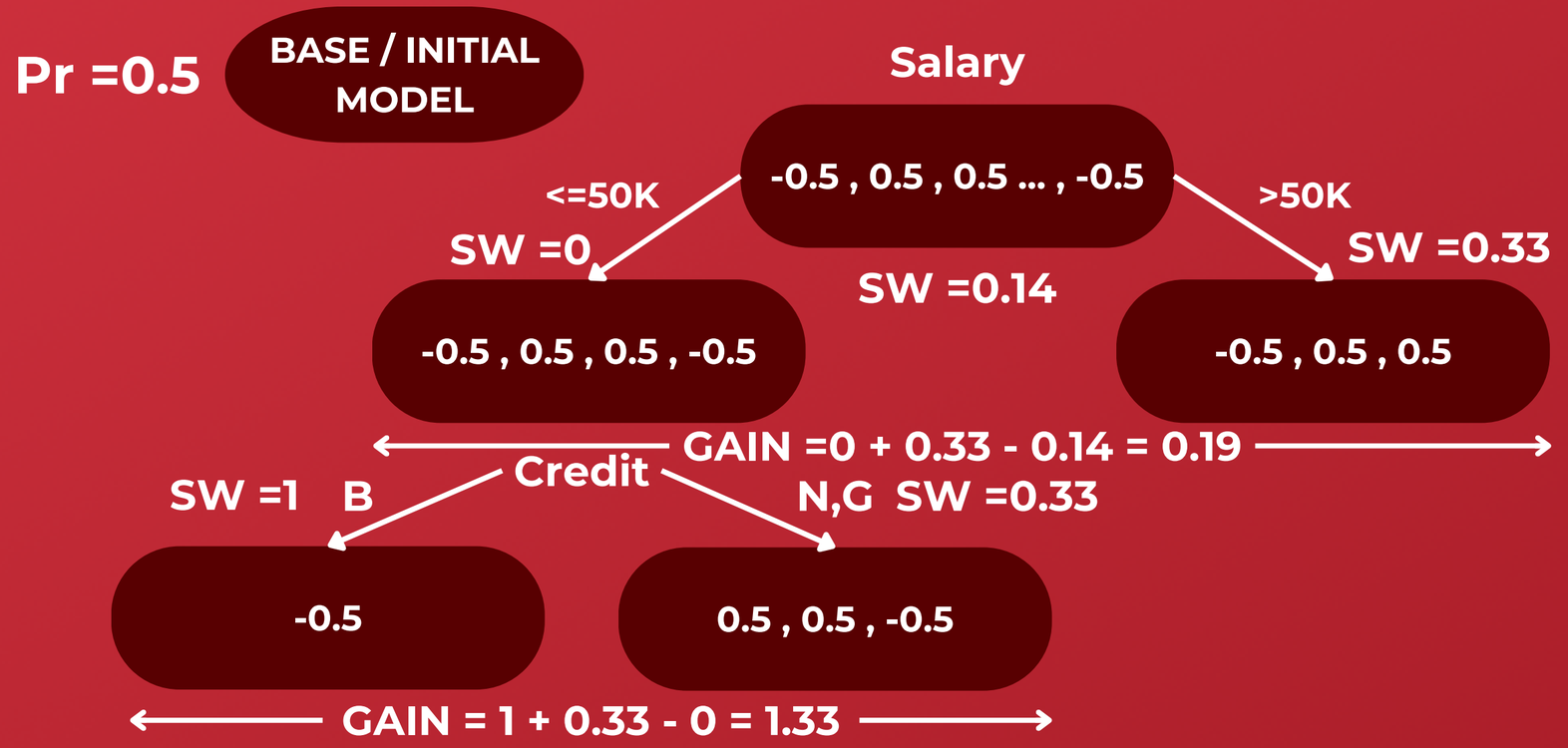
AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SALARY	CREDIT	APROVAL	RES
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5



- **STEPS :**
  - Calculate Residuals and Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

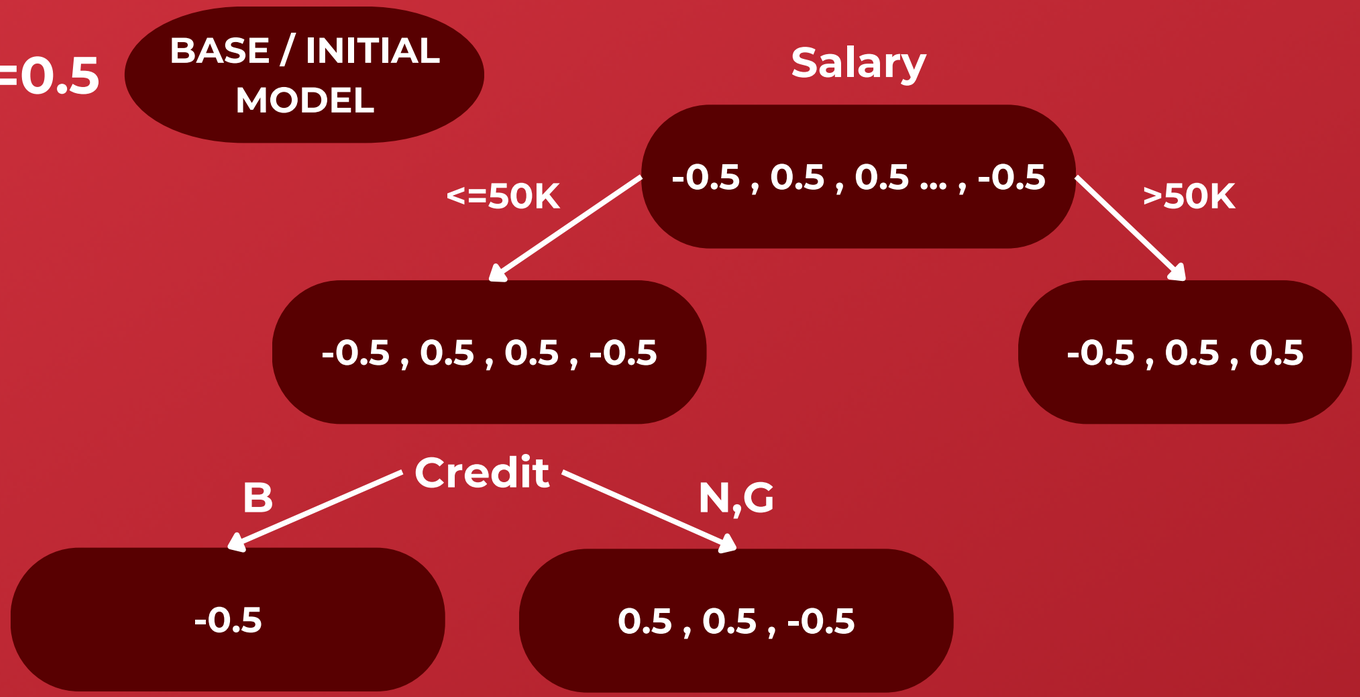
Knowing That : RES=APROVAL-Pr , λ=0

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **STEPS (COVER) :**
  - **Calculate the cover = ∑(Pr(1-Pr)) for each leaf**
  - **Compare each leaf cover and and the default minimum value of the cover (which is by default = 1)**
  - **Eliminate all the leaves that have a cover < minimum cover**



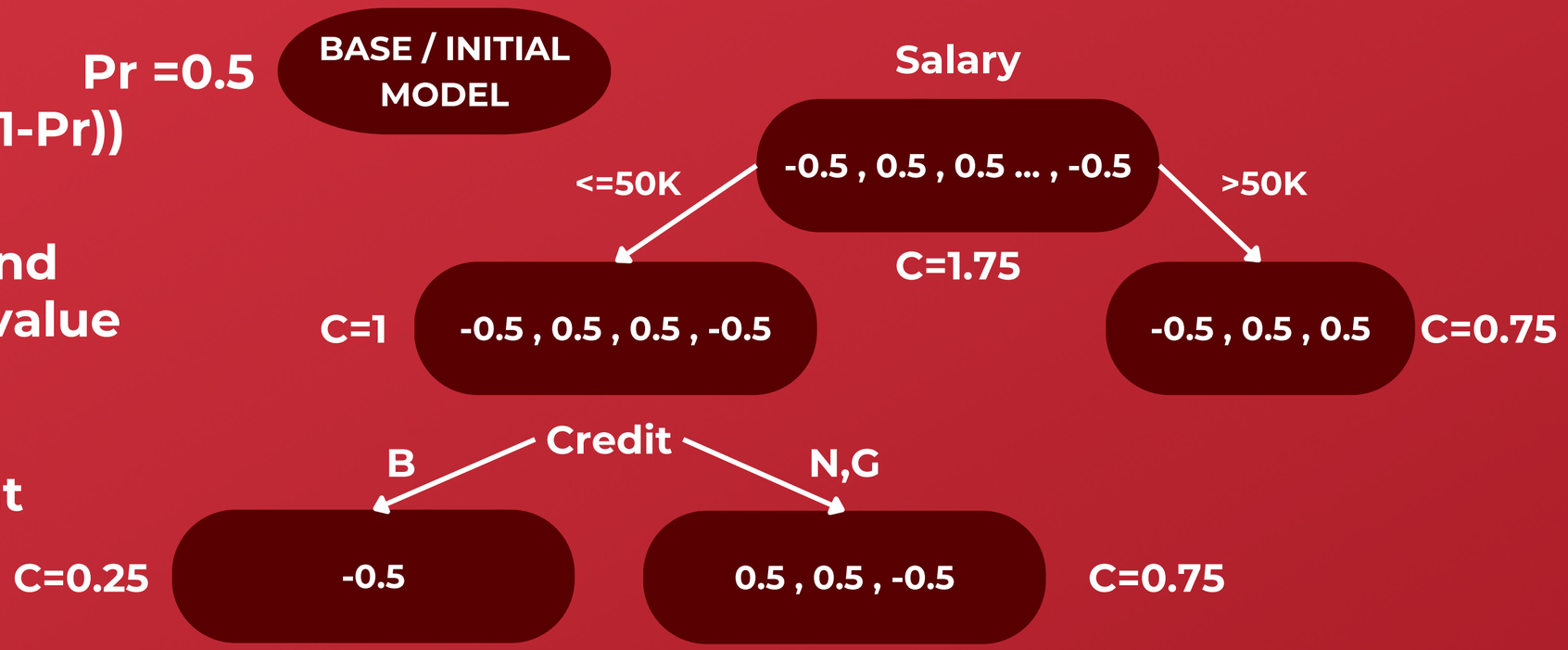
- **Example Minimum Cover = 1 :**
- **in Python :**
  By Default

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **STEPS (COVER) :**
  - **Calculate the cover = ∑(Pr(1-Pr)) for each leaf**
  - **Compare each leaf cover and and the default minimum value of the cover (which is by default = 1)**
  - **Eliminate all the leaves that have a cover < minimum cover**



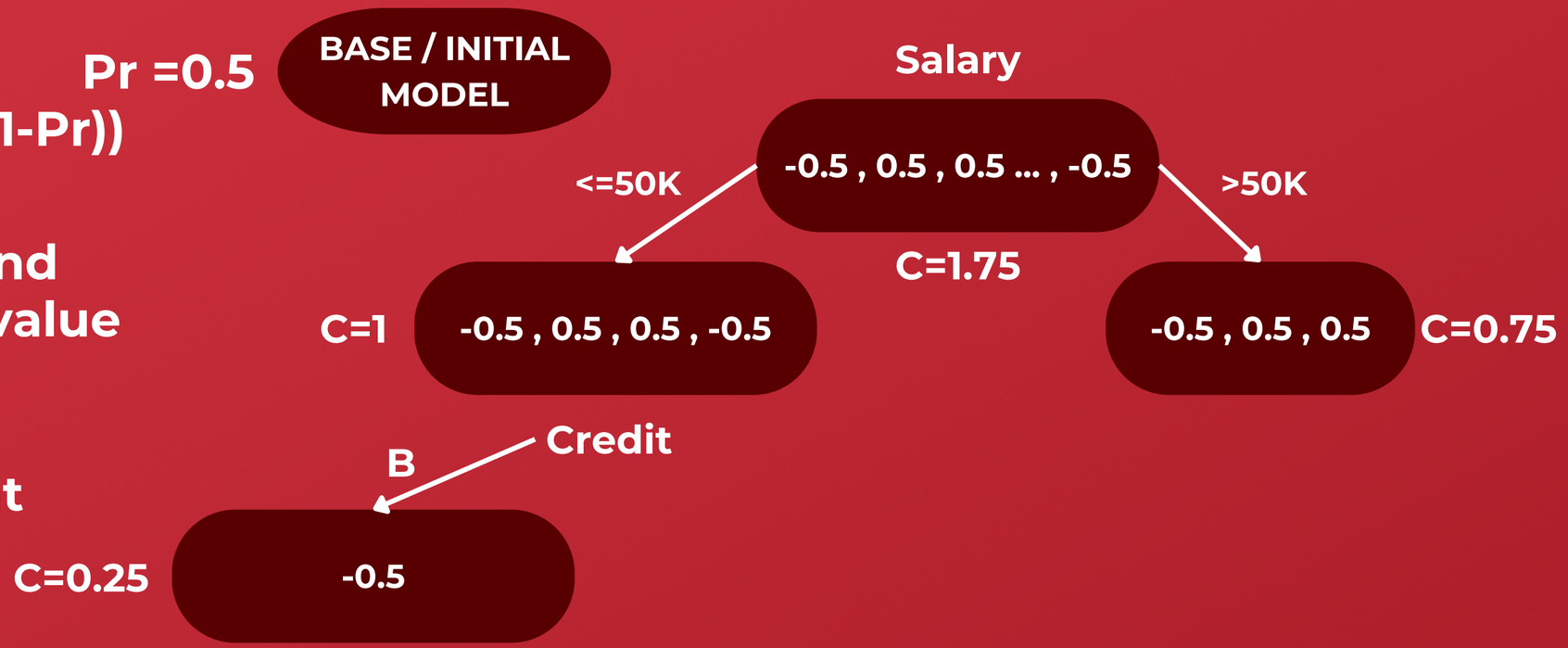
- **Example Minimum Cover = 1 :**
- **in Python :**
  By Default

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **STEPS (COVER) :**
  - **Calculate the cover = ∑(Pr(1-Pr)) for each leaf**
  - **Compare each leaf cover and and the default minimum value of the cover (which is by default = 1)**
  - **Eliminate all the leaves that have a cover < minimum cover**



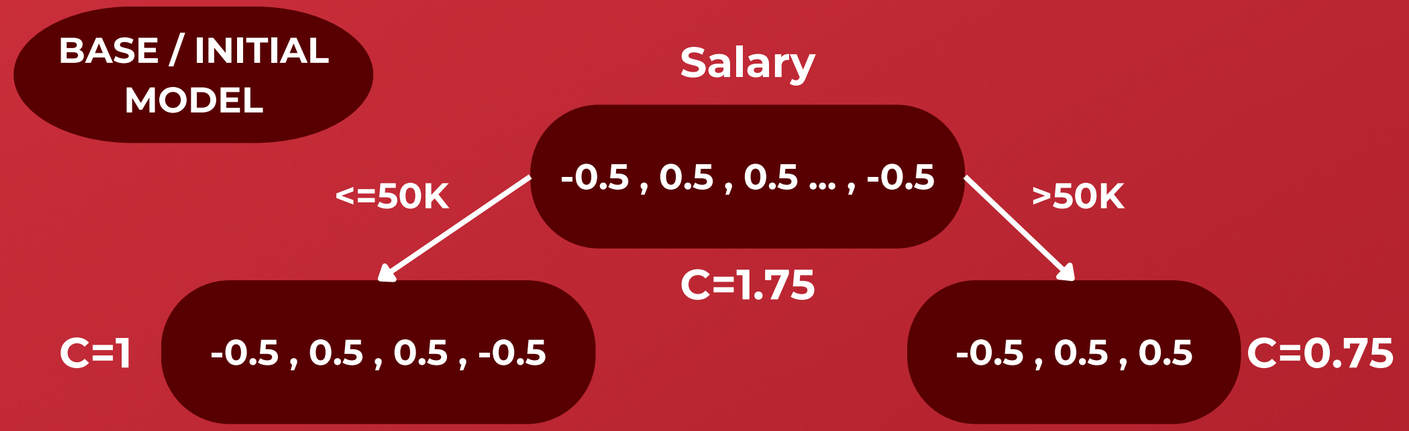
- **Example Minimum Cover = 1 :**
- **in Python :**
  By Default

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

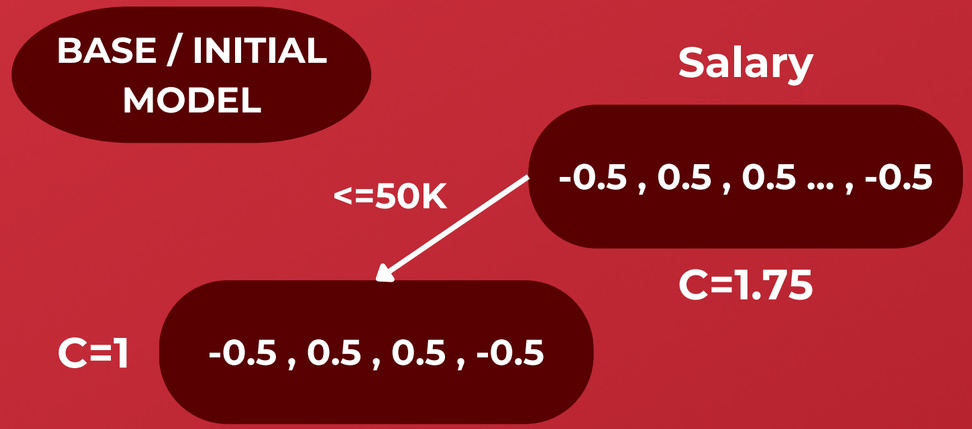
- **STEPS (COVER) :**
  - **Calculate the cover = ∑(Pr(1-Pr)) for each leaf**
  - **Compare each leaf cover and and the default minimum value of the cover (which is by default = 1)**
  - **Eliminate all the leaves that have a cover < minimum cover**



- **Example Minimum Cover = 1 :**
- **in Python :**
  By Default

XGBOOST CLASSIFICATION :

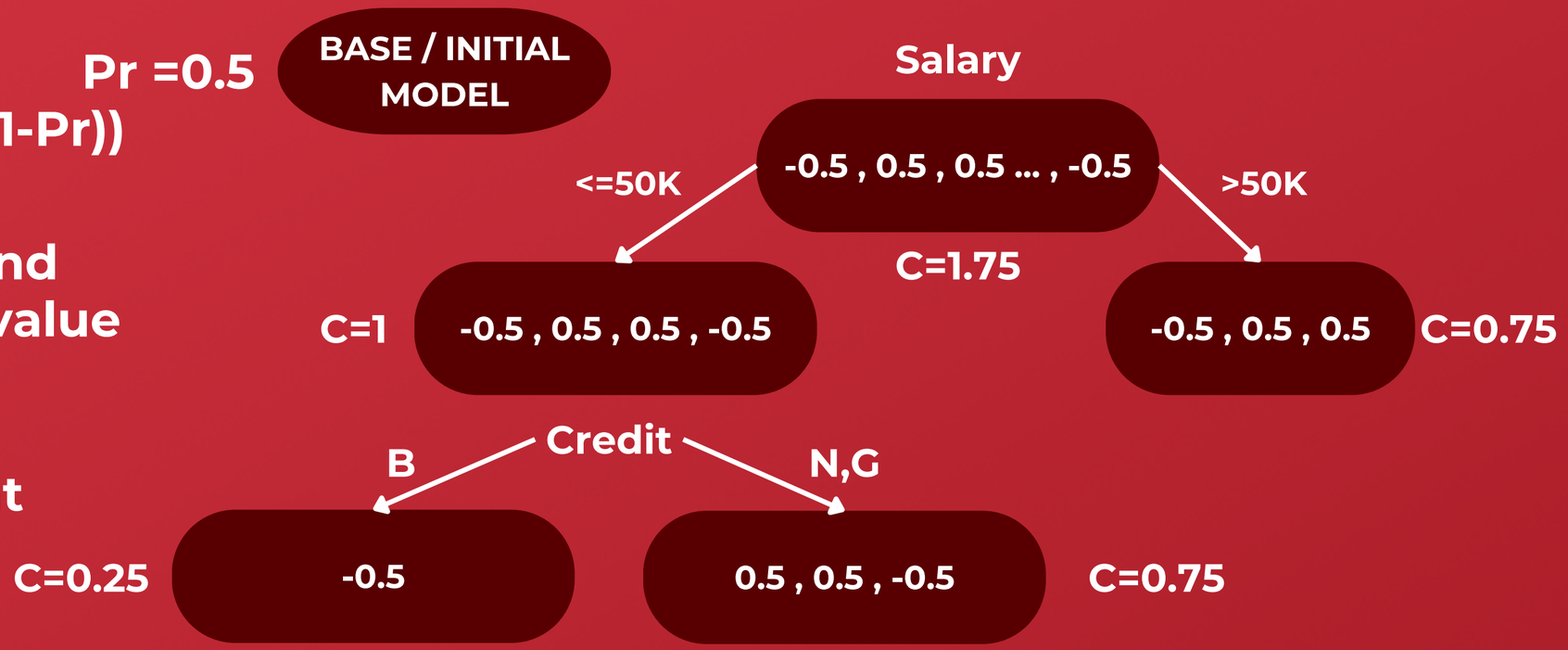
- **STEPS (COVER) :**
  - **Calculate the cover = ∑(Pr(1-Pr)) for each leaf**
  - **Compare each leaf cover and and the default minimum value of the cover (which is by default = 1)**
  - **Eliminate all the leaves that have a cover < minimum cover**



- **Example Minimum Cover = 1 :**
- **in Python :**
  By Default

XGBOOST CLASSIFICATION :

- **STEPS (COVER) :**
  - **Calculate the cover = ∑(Pr(1-Pr)) for each leaf**
  - **Compare each leaf cover and and the default minimum value of the cover (which is by default = 1)**
  - **Eliminate all the leaves that have a cover < minimum cover**



- **Example Minimum Cover = 3 :**
- **in Python : min_child_weight=3**

AI CAMP

school of ai Algiers

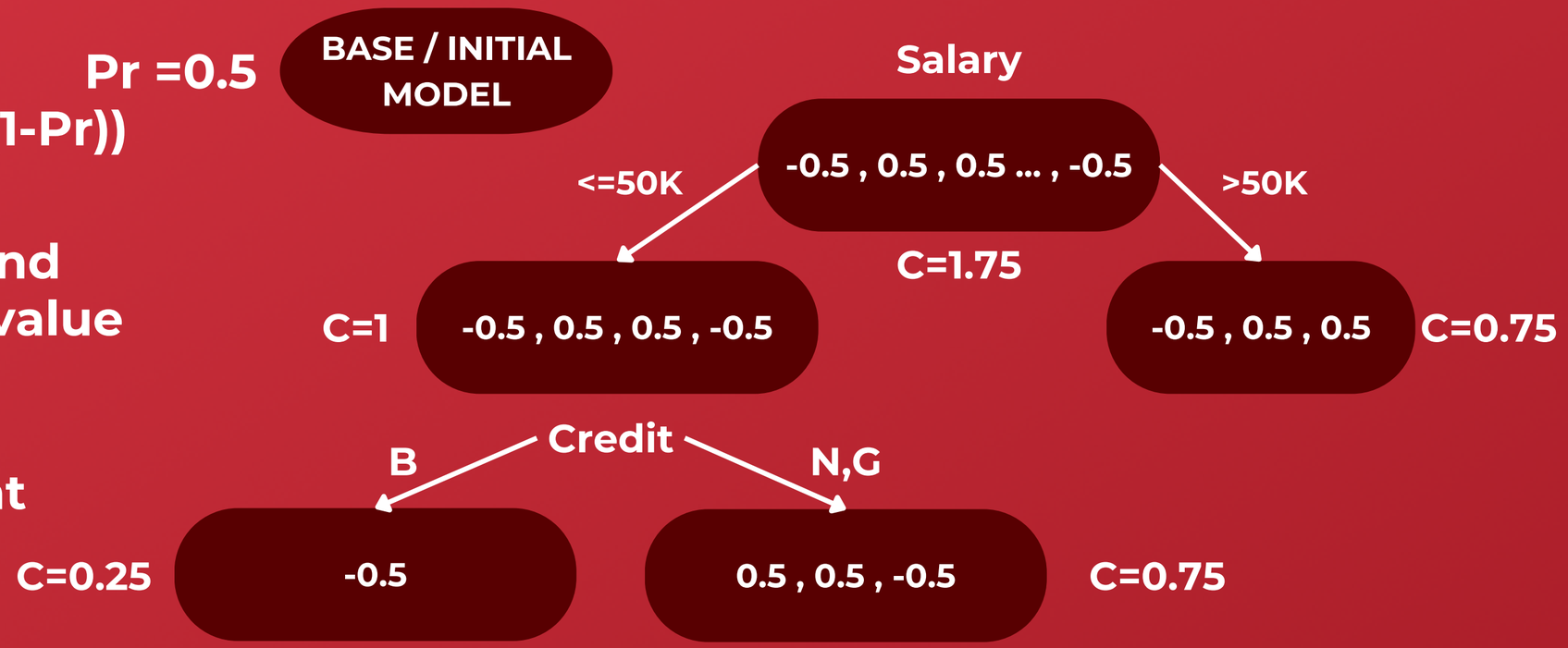
XGBOOST CLASSIFICATION :

- **STEPS (COVER) :** Pr =0.5 BASE / INITIAL MODEL
  - Calculate the cover = ∑(Pr(1-Pr)) for each leaf
  - Compare each leaf cover and and the default minimum value of the cover (which is by default = 1)
  - Eliminate all the leaves that have a cover < minimum cover

- Example Minimum Cover = 3 :
- in Python : min_child_weight=3

XGBOOST CLASSIFICATION :

- **STEPS (COVER) :**
  - Calculate the cover = ∑(Pr(1-Pr)) for each leaf
  - Compare each leaf cover and and the default minimum value of the cover (which is by default = 1)
  - Eliminate all the leaves that have a cover < minimum cover



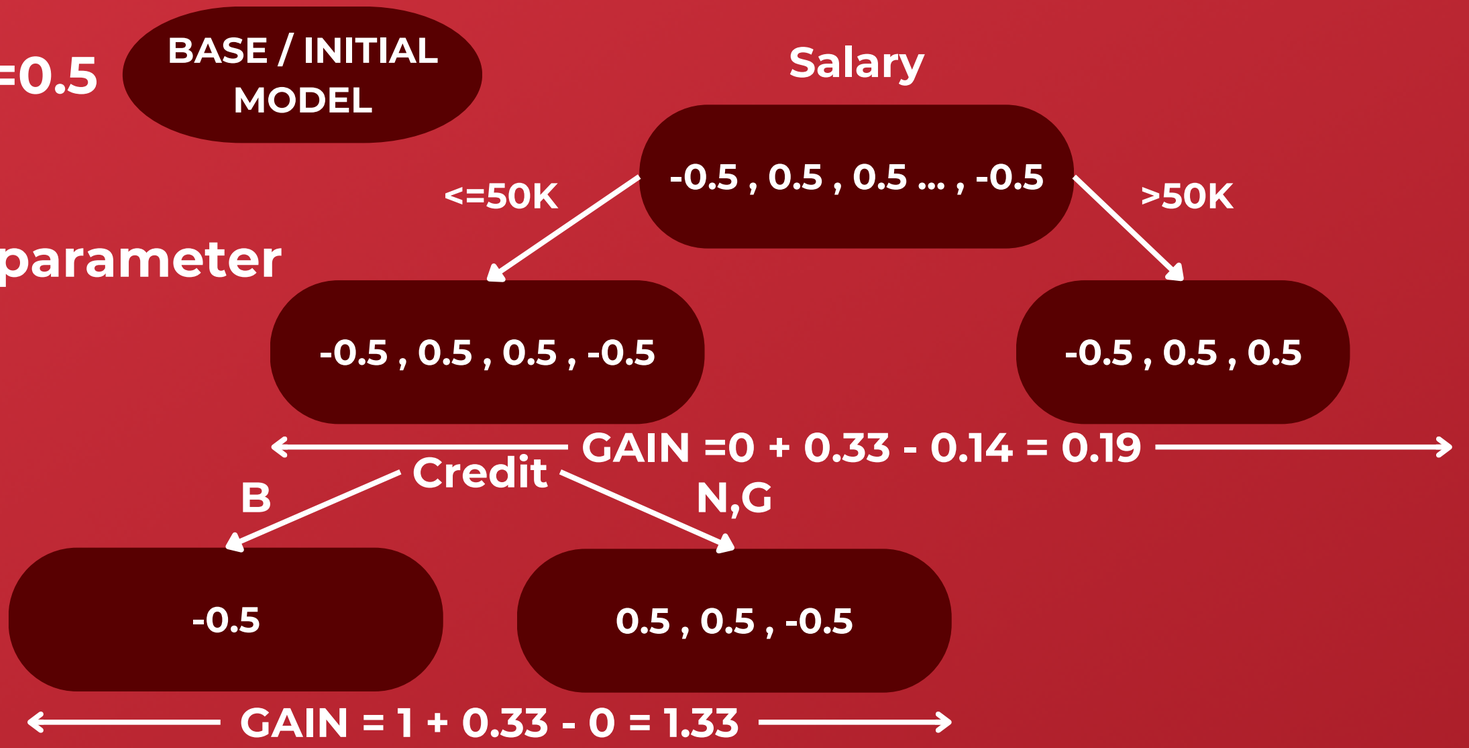
- **Example Minimum Cover = 0 :**
- **in Python :**
  min_child_weight=0

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **STEPS (PRUNING) :**
  - **Calculate the :**
    Gain of the branch - γ (Gamma)
    - **γ(Gamma) is Tree Complexity parameter**
  - **If The Gain - γ < 0 :**
  - **We Prune the branch**
  - **If we Prune we Calculate the :**
    Next Gain - γ
  - **Etc ...**



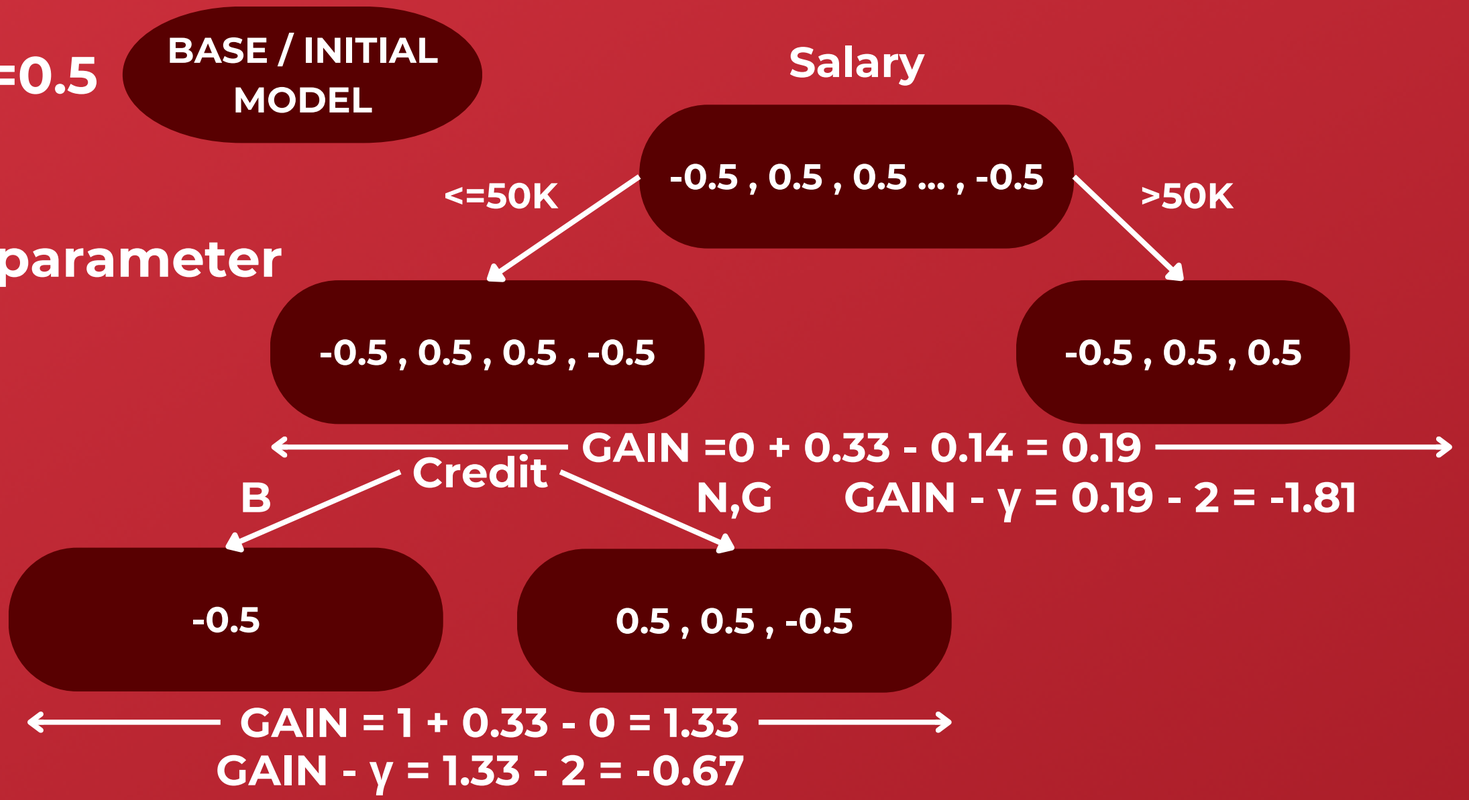
- **Example γ = 2 :**

AI CAMP

school of ai Algiers

XGBOOST CLASSIFICATION :

- **STEPS (PRUNING) :**
  - **Calculate the :**
    Gain of the branch - γ (Gamma)
    - **γ(Gamma) is Tree Complexity parameter**
  - **If The Gain - γ < 0 :**
  - **We Prune the branch**
  - **If we Prune we Calculate the :**
    Next Gain - γ
  - **Etc ...**



- **Example γ = 2 :**

AI CAMP

school of ai Algiers

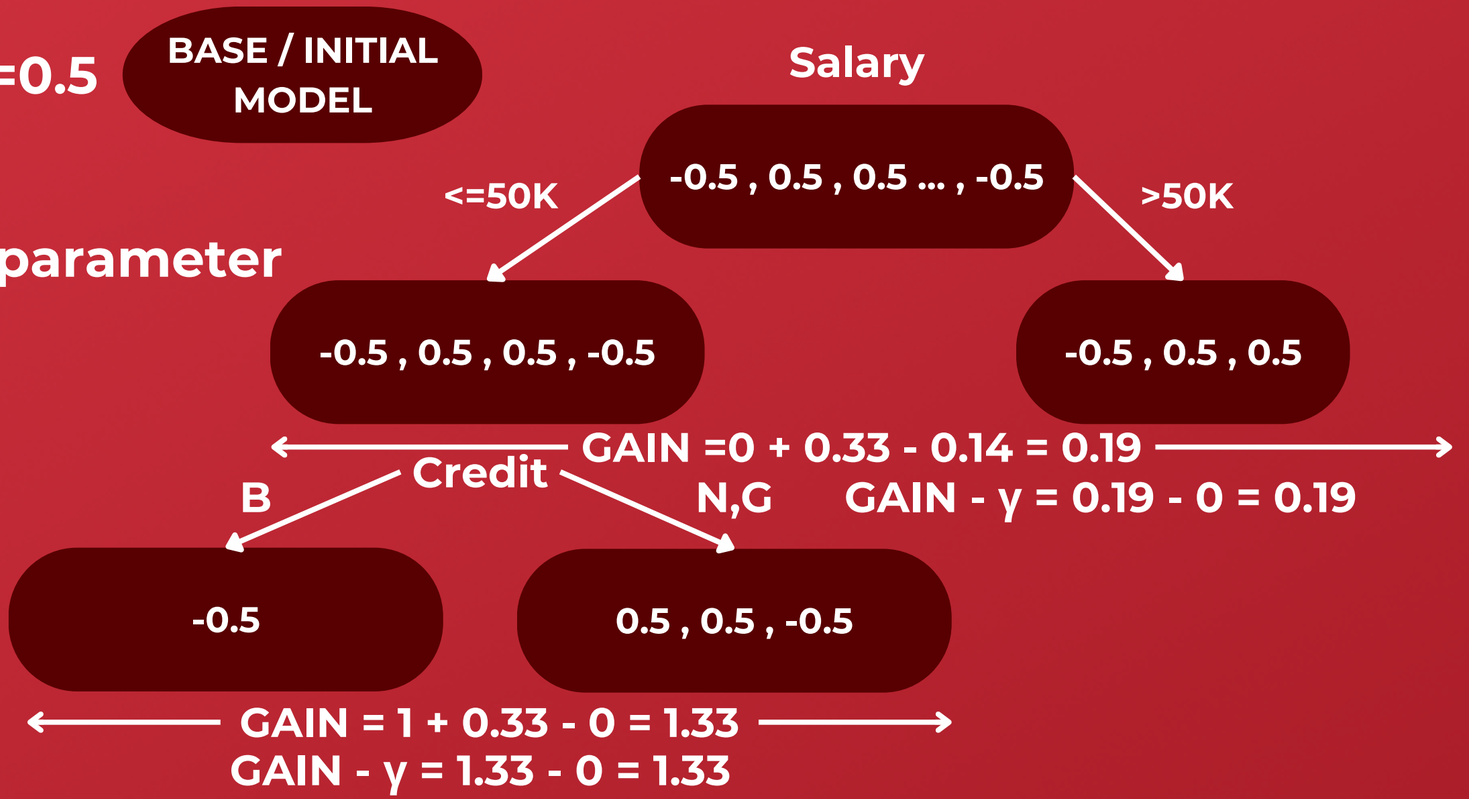
XGBOOST CLASSIFICATION :

- **STEPS (PRUNING) :** Pr =0.5 BASE / INITIAL MODEL
  - Calculate the :
    Gain of the branch - γ (Gamma)
    - γ(Gamma) is Tree Complexity parameter
  - If The Gain - γ < 0 :
  - We Prune the branch
  - If we Prune we Calculate the :
    Next Gain - γ
  - Etc ...

- Example γ = 2 :

XGBOOST CLASSIFICATION :

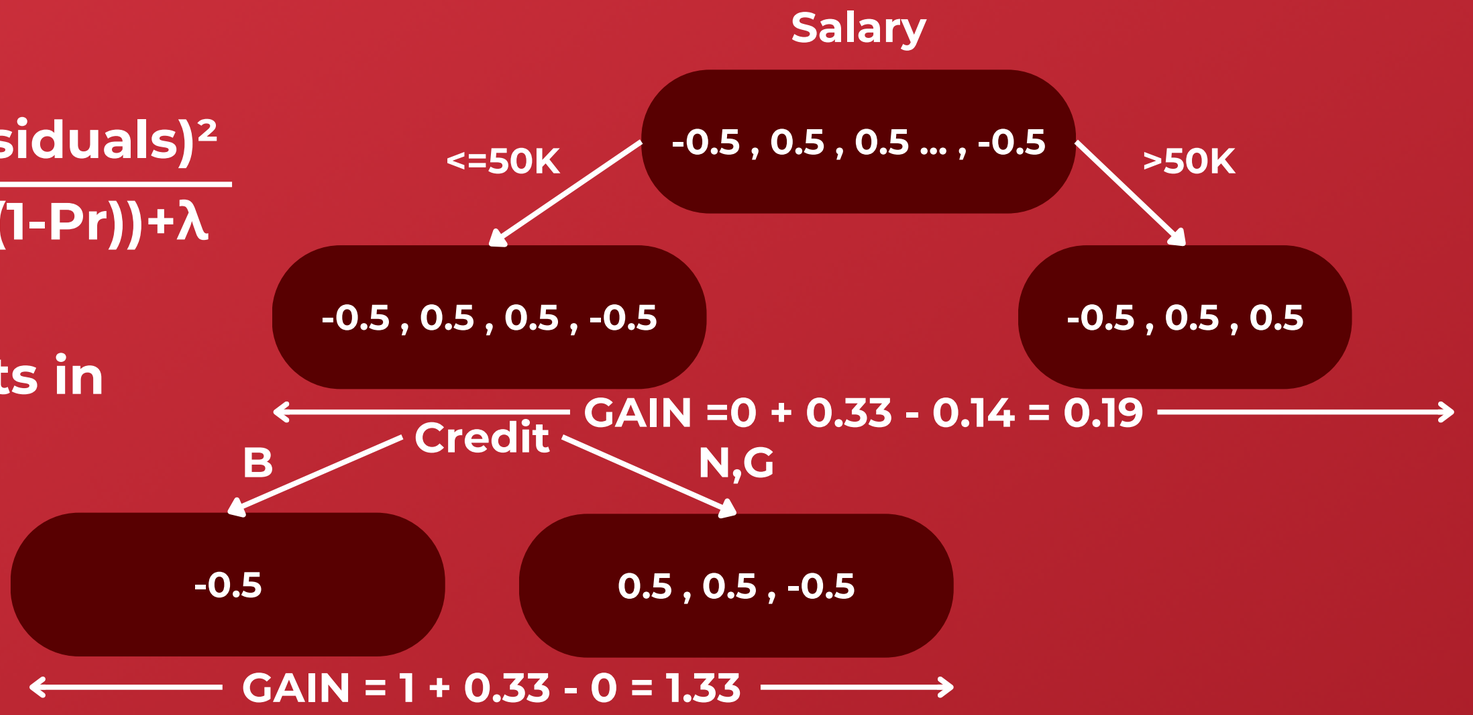
- **STEPS (PRUNING) :**
  - **Calculate the :**
    Gain of the branch - γ (Gamma)
    - **γ(Gamma) is Tree Complexity parameter**
  - **If The Gain - γ < 0 :**
  - **We Prune the branch**
  - **If we Prune we Calculate the :**
    Next Gain - γ
  - **Etc ...**



- **Example γ = 0 :**

XGBOOST CLASSIFICATION :

- **STEPS (PRUNING) :**
  - **Lastly, We Know That SW =** $\frac{\sum(\text{Residuals})^2}{\sum(Pr(1-Pr))+\lambda}$
  - **So λ (Lambda) is a Regularization Parameter and when λ>0, it results in more pruning by shrinking the Similarity Weights and branches gains .**
  - **Example γ=0 & λ=0:**

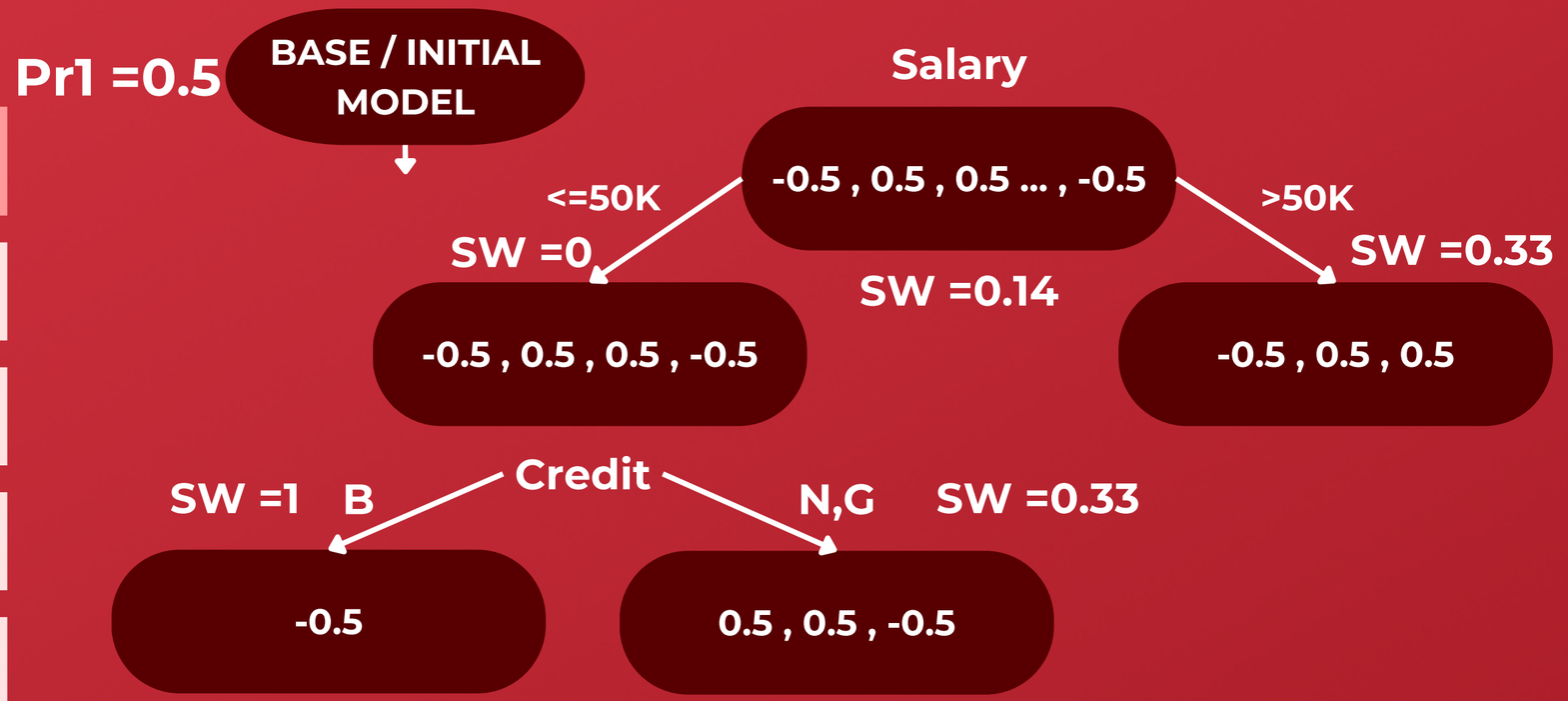


- **WHY PRUNING ? :**
  - **Pruning removes less important branches of decision trees, reducing complexity and making the model less likely to overfit and also, smaller pruned trees are often easier to understand and interpret**

XGBOOST CLASSIFICATION :

- **Dataset :**

SY	CR	AP	R1	Pr2	R2
<=50K	B	0	-0.5		
<=50K	G	1	0.5		
<=50K	G	1	0.5		
>50K	B	0	-0.5		
>50K	G	1	0.5		
>50K	N	1	0.5		
<=50K	N	0	-0.5		



- **FINAL STEPS :**
  - **Calculate the output of the Base Model using : log(odds) = log(Pr/1-Pr). In our case it is 0**
  - **Calculate the new probability of each record using : σ(0+ α(DT 1) + α(DT 2) + ... + α(DT N))**

AI CAMP

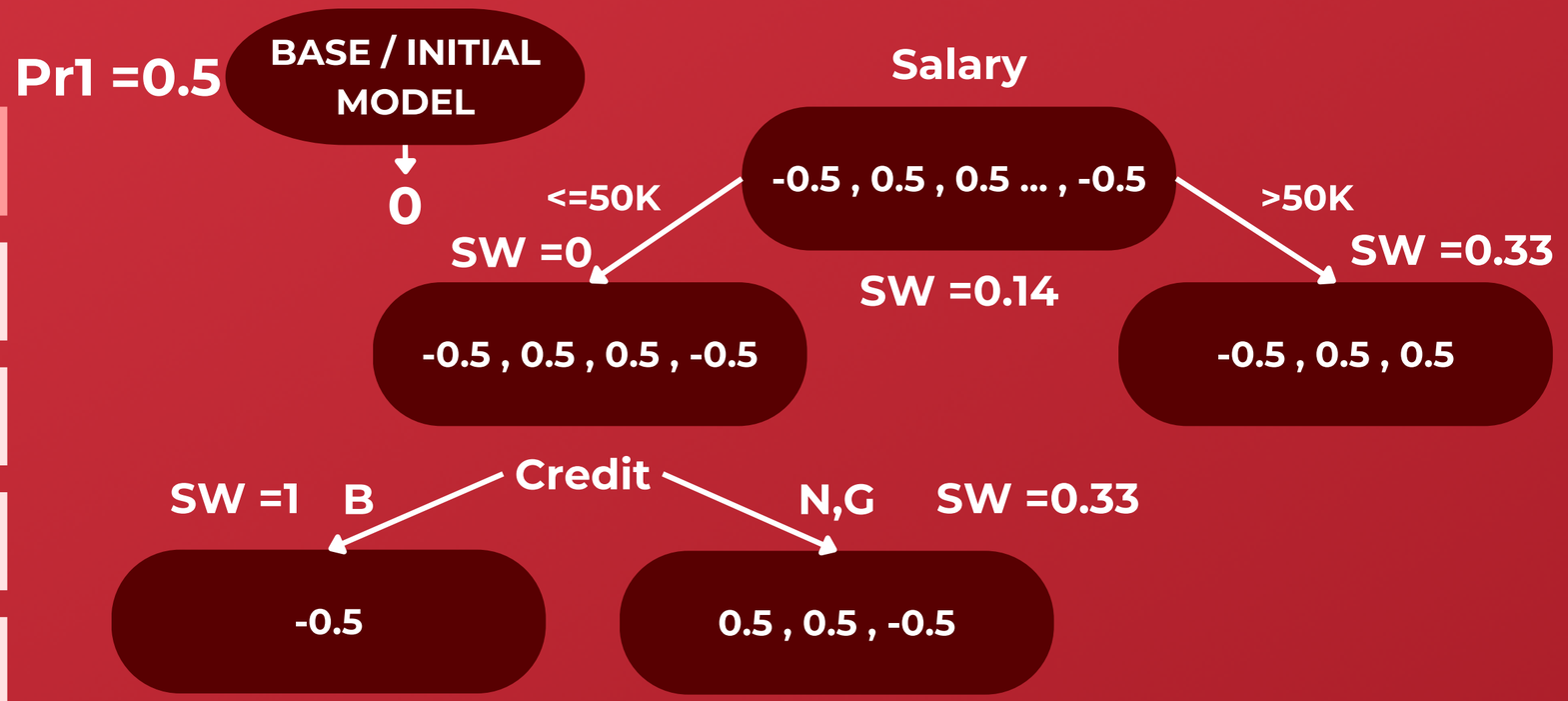
Knowing That : σ(x) =(1/1-e(x)) , α is the learning rate here = 0.1

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SY	CR	AP	R1	Pr2	R2
<=50K	B	0	-0.5		
<=50K	G	1	0.5		
<=50K	G	1	0.5		
>50K	B	0	-0.5		
>50K	G	1	0.5		
>50K	N	1	0.5		
<=50K	N	0	-0.5		



- **FINAL STEPS :**
  - **Calculate the output of the Base Model using : log(odds) = log(Pr/1-Pr). In our case it is 0**
  - **Calculate the new probability of each record using : σ(0+ α(DT 1) + α(DT 2) + ... + α(DT N))**

AI CAMP

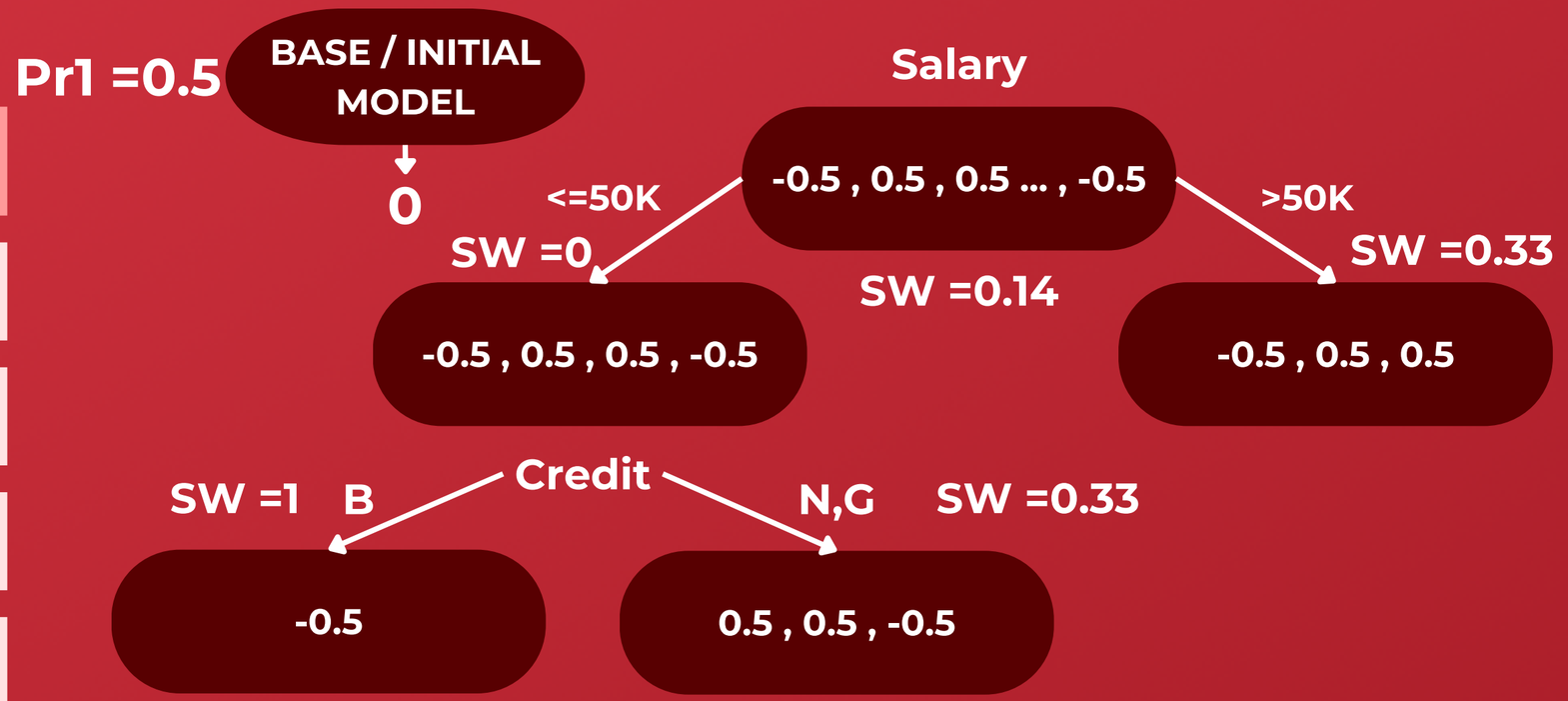
Knowing That : σ(x) =(1/1-e(x)) , α is the learning rate here = 0.1

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SY	CR	AP	R1	Pr2	R2
<=50K	B	0	-0.5	0.52	
<=50K	G	1	0.5		
<=50K	G	1	0.5		
>50K	B	0	-0.5		
>50K	G	1	0.5		
>50K	N	1	0.5		
<=50K	N	0	-0.5		



- **FINAL STEPS :**
  - **Calculate the output of the Base Model using : log(odds) = log(Pr/1-Pr). In our case it is 0**
  - **Calculate the new probability of each record using : σ(0+ α(DT 1) + α(DT 2) + ... + α(DT N))**

AI CAMP

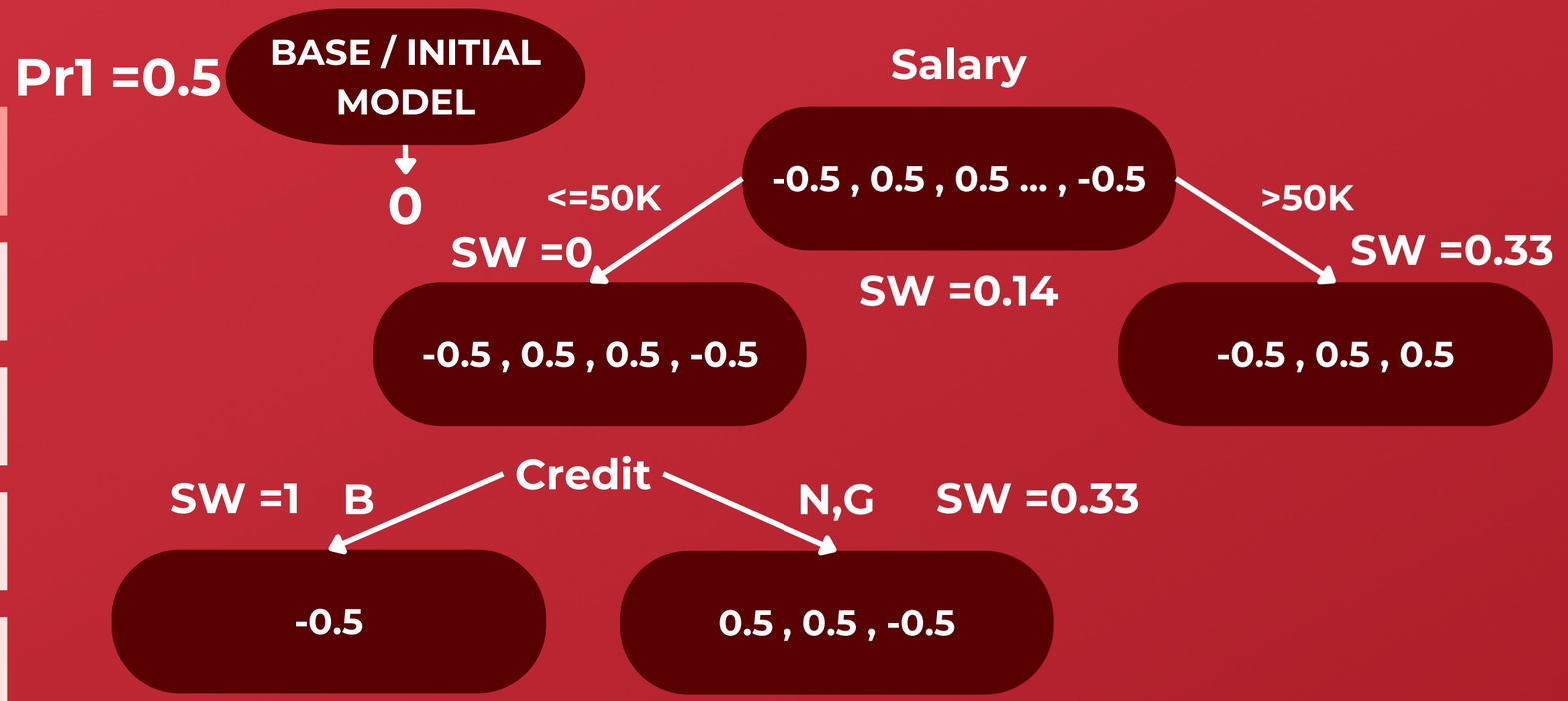
Knowing That : σ(x) =(1/1-e(x)) , α is the learning rate here = 0.1

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SY	CR	AP	R1	Pr2	R2
<=50K	B	0	-0.5	0.52	
<=50K	G	1	0.5	0.5	
<=50K	G	1	0.5	0.5	
>50K	B	0	-0.5	0.5	
>50K	G	1	0.5	0.5	
>50K	N	1	0.5	0.5	
<=50K	N	0	-0.5	0.5	



- **FINAL STEPS :**
  - **Calculate the output of the Base Model using : log(odds) = log(Pr/1-Pr). In our case it is 0**
  - **Calculate the new probability of each record using : σ(0+ α(DT 1) + α(DT 2) + ... + α(DT N))**

AI CAMP

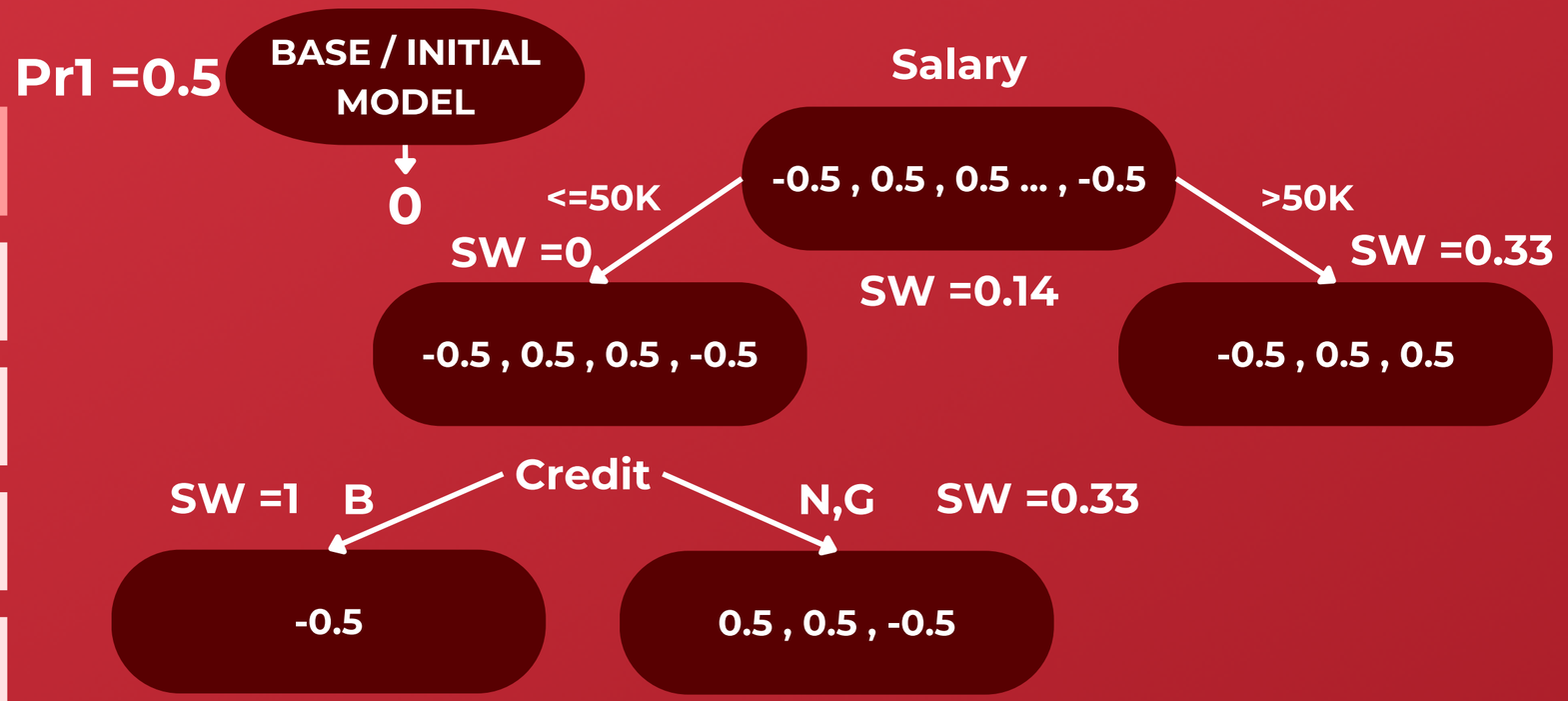
Knowing That : σ(x) =(1/1-e(x)) , α is the learning rate here = 0.1

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SY	CR	AP	R1	Pr2	R2
<=50K	B	0	-0.5	0.52	-0.48
<=50K	G	1	0.5	0.5	
<=50K	G	1	0.5	0.5	
>50K	B	0	-0.5	0.5	
>50K	G	1	0.5	0.5	
>50K	N	1	0.5	0.5	
<=50K	N	0	-0.5	0.5	



- **FINAL STEPS :**
  - **Calculate the output of the Base Model using : log(odds) = log(Pr/1-Pr). In our case it is 0**
  - **Calculate the new probability of each record using : σ(0+ α(DT 1) + α(DT 2) + ... + α(DT N))**

AI CAMP

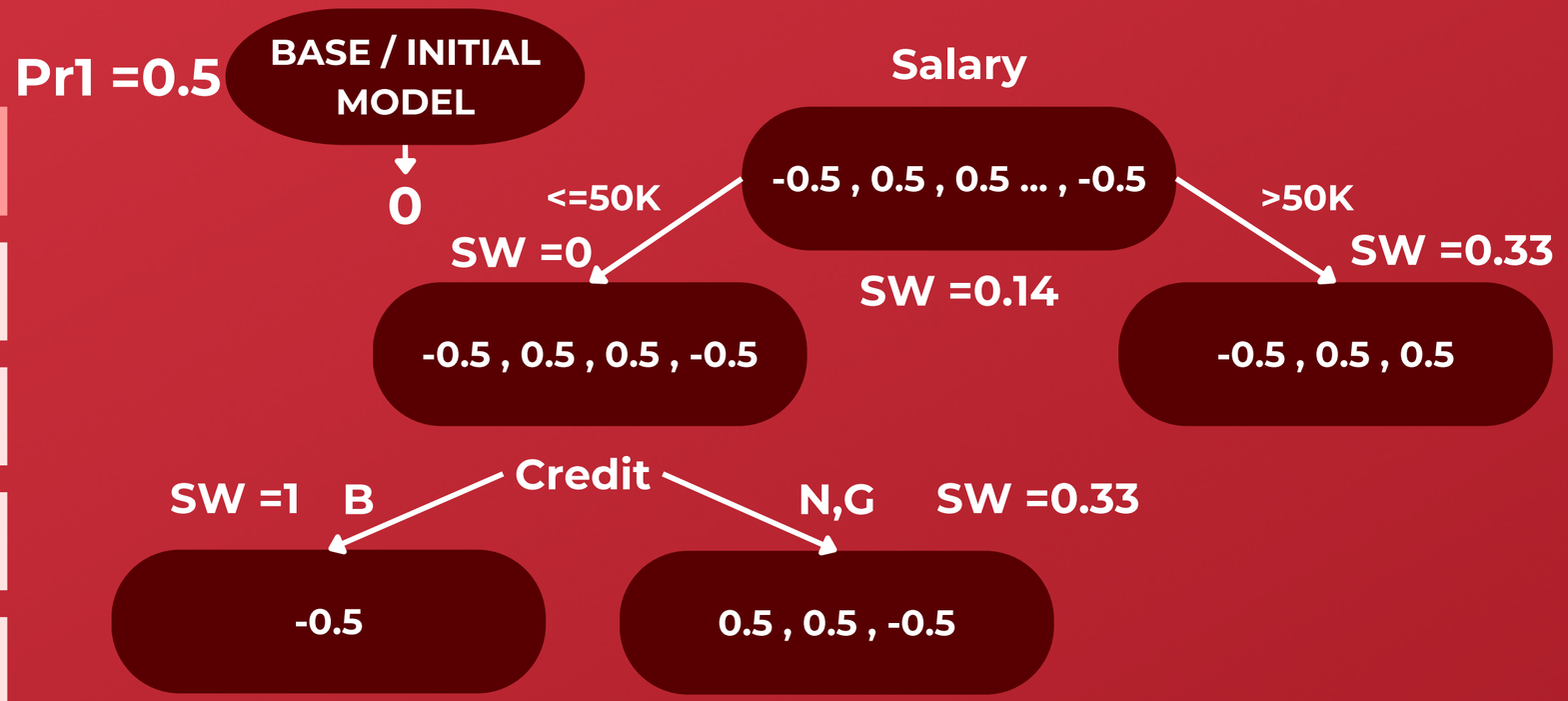
Knowing That : σ(x) =(1/1-e(x)) , α is the learning rate here = 0.1

school of ai Algiers

XGBOOST CLASSIFICATION :

- **Dataset :**

SY	CR	AP	R1	Pr2	R2
<=50K	B	0	-0.5	0.52	-0.48
<=50K	G	1	0.5	0.5	0.5
<=50K	G	1	0.5	0.5	0.5
>50K	B	0	-0.5	0.5	-0.5
>50K	G	1	0.5	0.5	0.5
>50K	N	1	0.5	0.5	0.5
<=50K	N	0	-0.5	0.5	-0.5



- **FINAL STEPS :**
  - **Calculate the output of the Base Model using : log(odds) = log(Pr/1-Pr). In our case it is 0**
  - **Calculate the new probability of each record using : σ(0+ α x (DT 1) + α x (DT 2) + ... + α x (DT N))**

AI CAMP

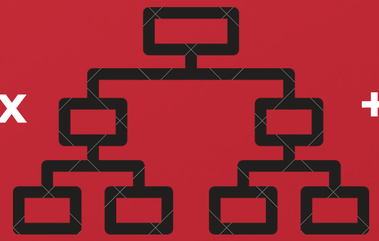
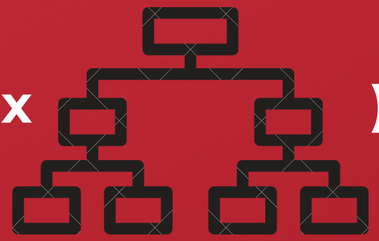
Knowing That : σ(x) =(1/1-e(x)) , α is the learning rate here = 0.1

school of ai Algiers

XGBOOST CLASSIFICATION :

DataSet → BASE / INITIAL MODEL → DT 1 → DT N

Probability = σ(0 + α x (DT 1) + ... + α x (DT N))

Probability = σ(BASE / INITIAL MODEL + α x  + ... + α x )

- **FINAL STEPS :**
  - **Calculate the output of the Base Model using : log(odds) = log(Pr/1-Pr). In our case it is 0**
  - **Calculate the new probability of each record using : σ(0+ α x (DT 1) + α x (DT 2) + ... + α x (DT N))**

AI CAMP

Knowing That : σ(x) =(1/1-e(x)) , α is the learning rate, here α= 0.1

school of ai Algiers

AI CAMP

school of ai Algiers

LET'S MOVE TO XGBOOST FOR REGRESSION

EXTREME GRADIENT BOOSTING WITH XGBOOST

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

BASE / INITIAL MODEL

EXP	GAP	SALARY	RES
2	Y	40k	
2.5	Y	42k	
3	N	52k	
4	N	60k	
4.5	Y	62k	

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

BASE / INITIAL MODEL

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	
3	N	52k	
4	N	60k	
4.5	Y	62k	

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

BASE / INITIAL MODEL

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

BASE / INITIAL MODEL

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k

Split 1
EXP

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :**

Avg Salary= 51k

BASE / INITIAL MODEL

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k

Split 1
EXP

-11 , -9 , 1 , 9 , 11

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

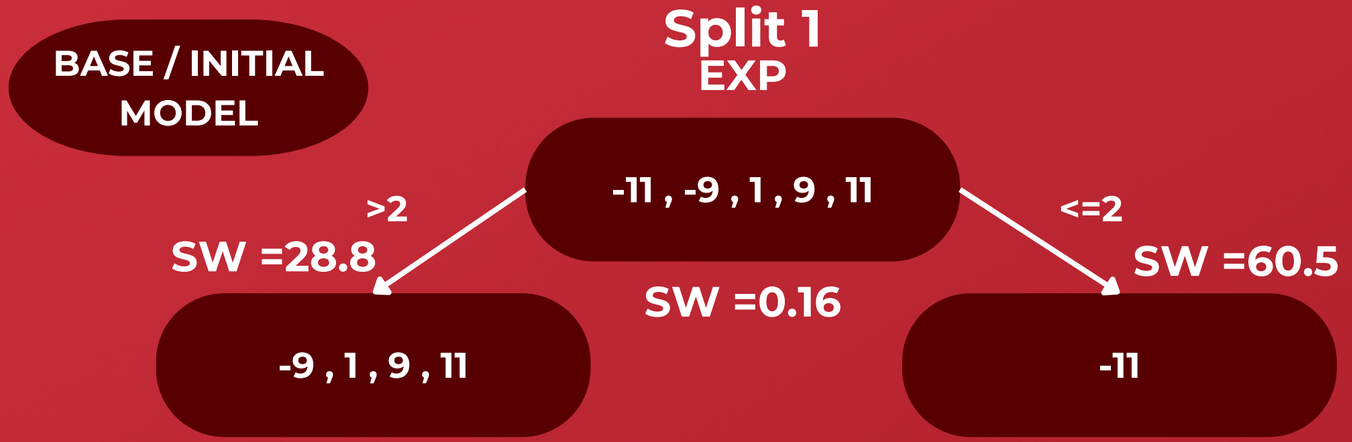
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k



- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

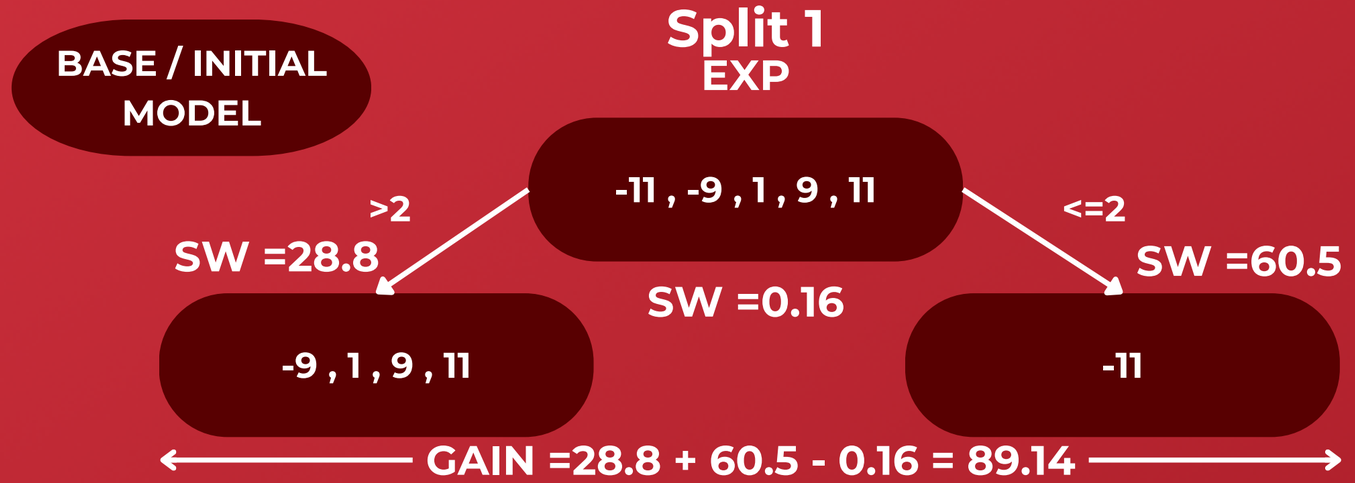
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k



- Gain Split 1 = 89.14

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

BASE / INITIAL MODEL

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k

Split 2
EXP

-11 , -9 , 1 , 9 , 11

- Gain Split 1 = 89.14

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

AI CAMP

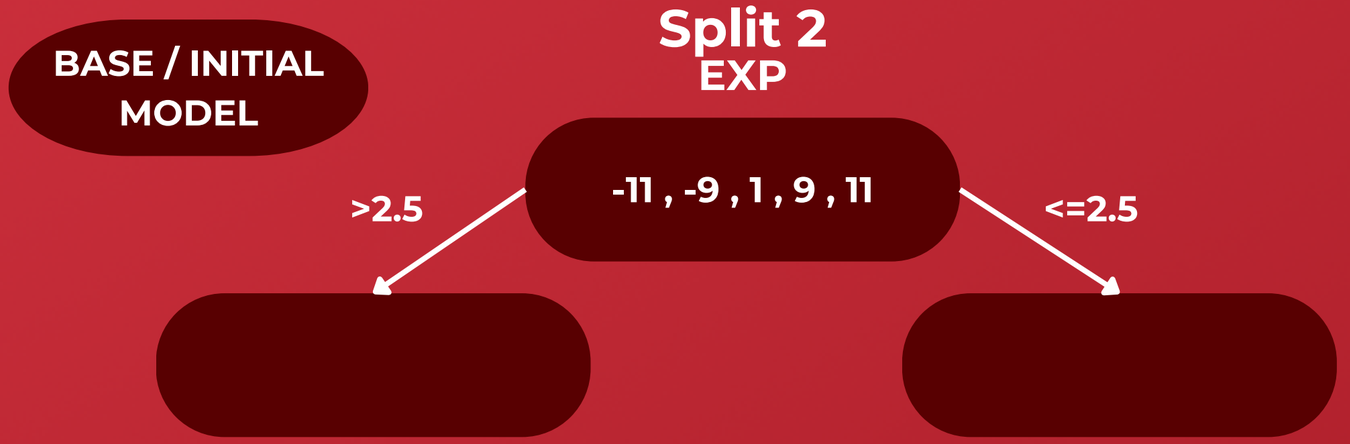
school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k

- Gain Split 1 = 89.14



- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k

- Gain Split 1 = 89.14

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

AI CAMP

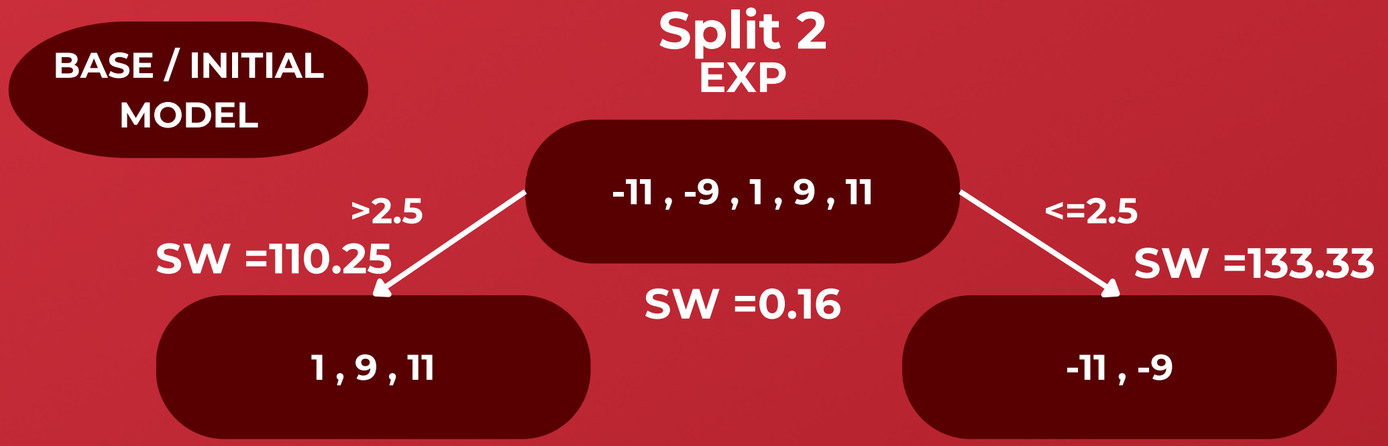
school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k

- Gain Split 1 = 89.14



- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

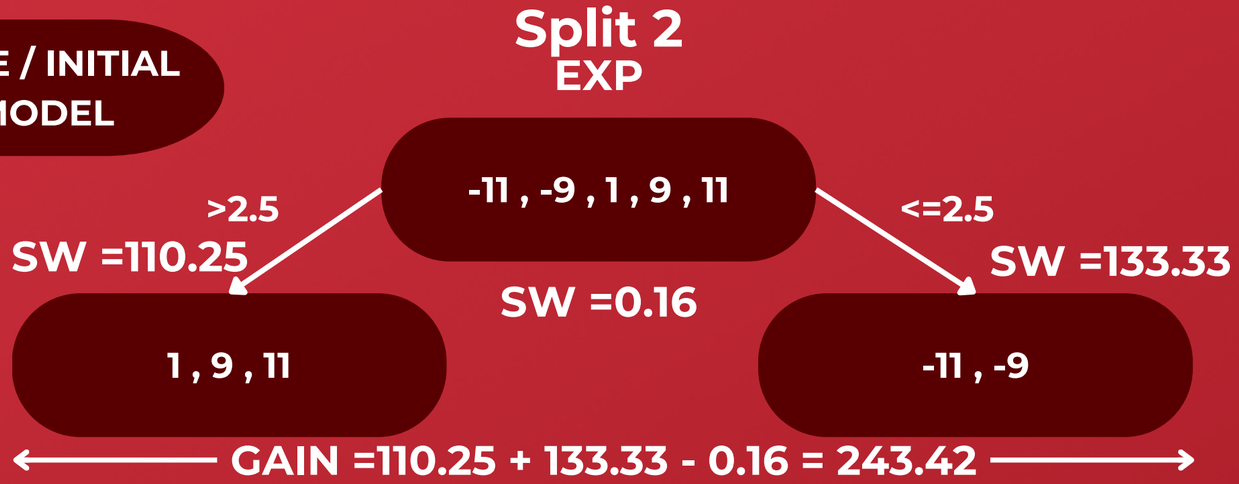
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k



- Gain Split 1 = 89.14
- Gain Split 2 = 243.42
- Gain Split 2 > Gain Split 1 So : We keep The Split 2

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

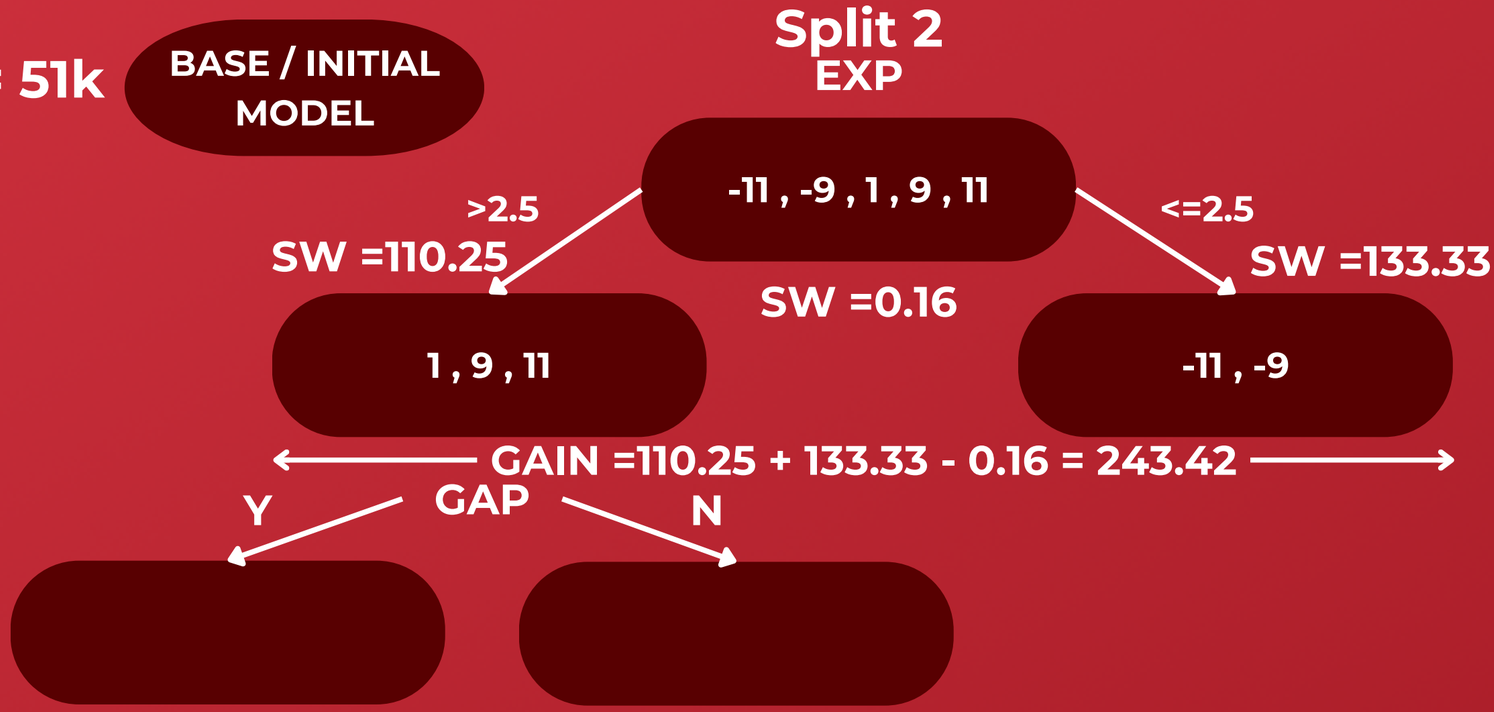
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k



- Gain Split 1 = 89.14
- Gain Split 2 = 243.42
- Gain Split 2 > Gain Split 1 So : We keep The Split 2

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

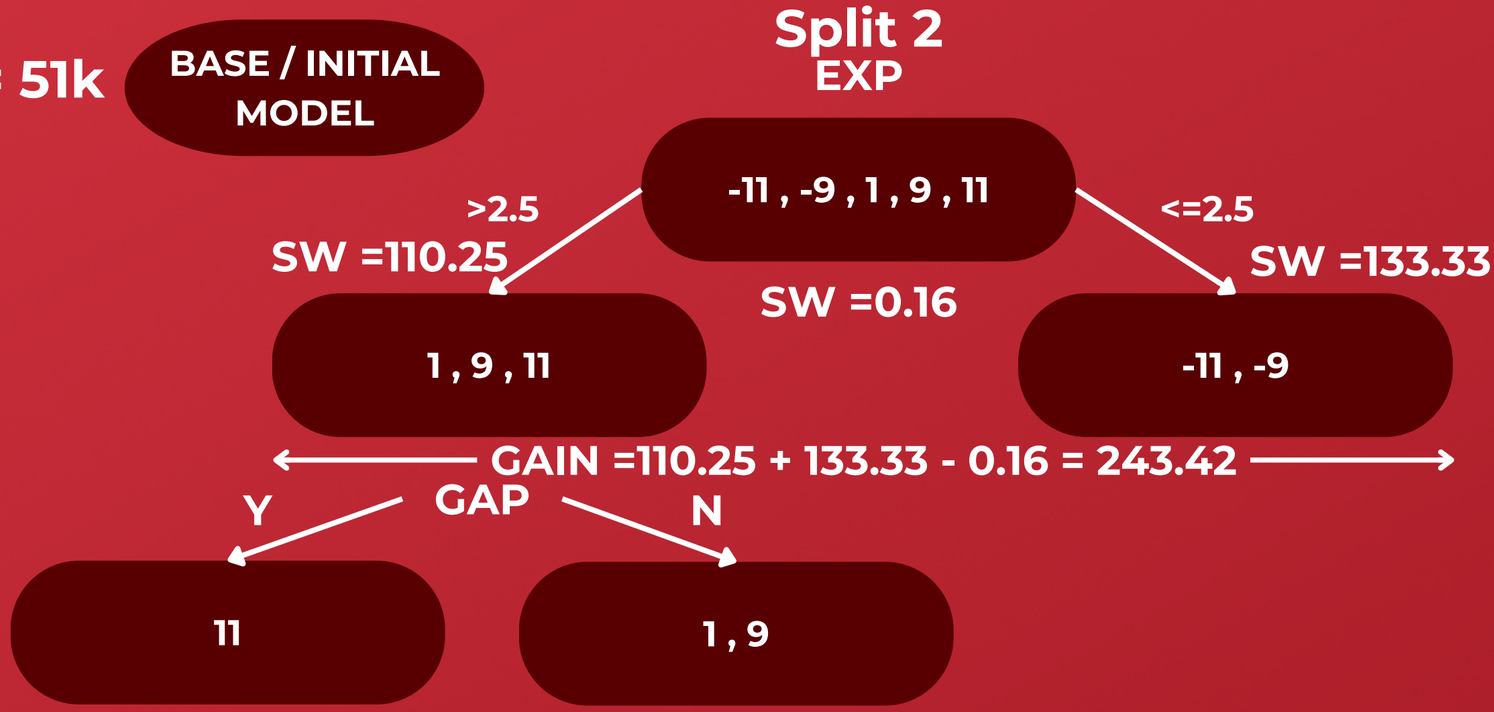
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k



- Gain Split 1 = 89.14
- Gain Split 2 = 243.42
- Gain Split 2 > Gain Split 1 So : We keep The Split 2

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

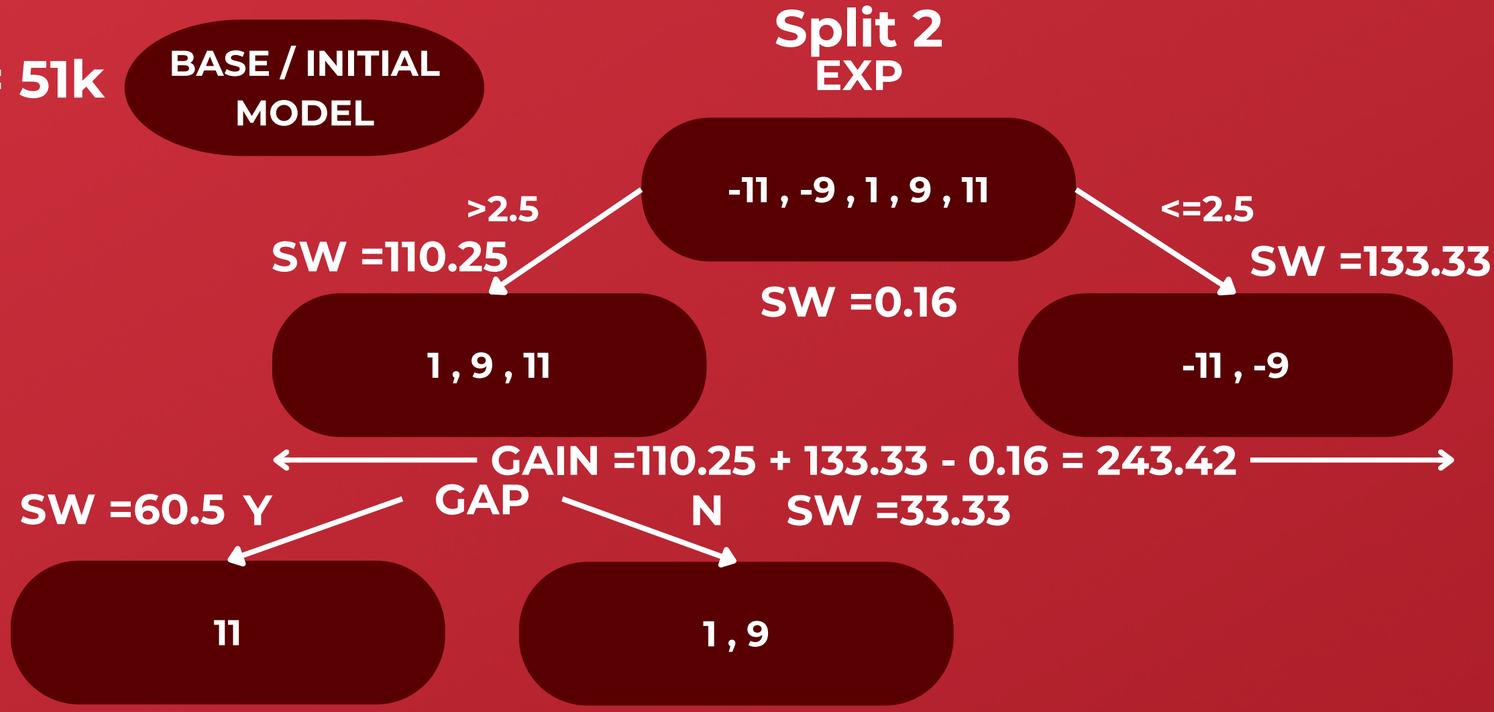
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k



- Gain Split 1 = 89.14
- Gain Split 2 = 243.42
- Gain Split 2 > Gain Split 1 So : We keep The Split 2

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

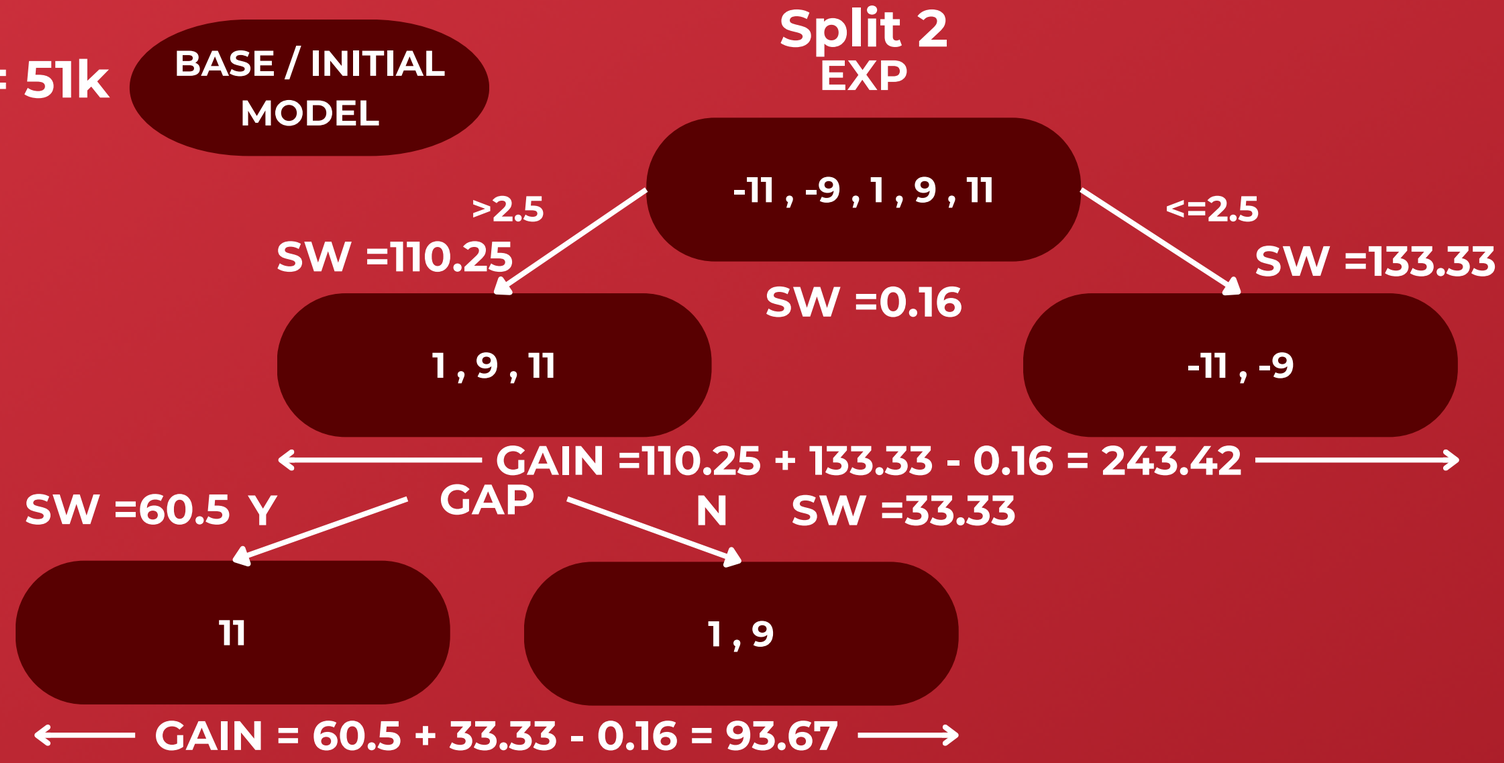
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EXP	GAP	SALARY	RES
2	Y	40k	-11k
2.5	Y	42k	-9k
3	N	52k	1k
4	N	60k	9k
4.5	Y	62k	11k



- Gain Split 1 = 89.14
- Gain Split 2 = 243.42
- Gain Split 2 > Gain Split 1 So : We keep The Split 2

- **STEPS :**
  - Construct tree with root
  - Calculate Similarity Weight = $\frac{\sum(\text{Residuals})^2}{\text{Nb of Res} + \lambda}$
  - Calculate the gain = SWL+SWR-SW(ROOT)

Knowing That : RES = Salary - Avg Salary , λ=1

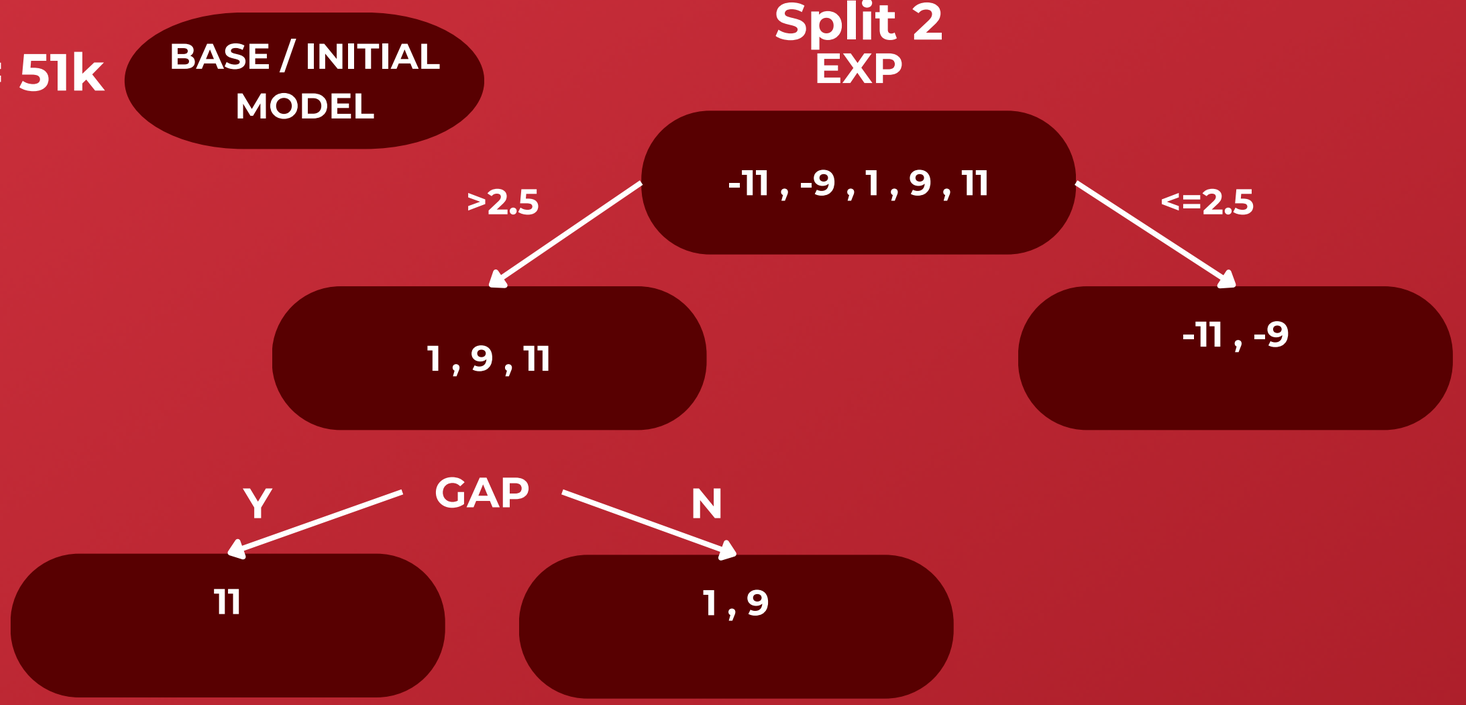
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EX	GP	SY	RS1	O/P	RS2
2	Y	40k	-11k		
2.5	Y	42k	-9k		
3	N	52k	1k		
4	N	60k	9k		
4.5	Y	62k	11k		



- **FINAL STEPS :**
  - Calculate the output of each bottom leaf using :
    Output = Avg of RES = (Sum Residuals)/(Nb Residuals)
  - Calculate the new probability of each record using :
    O/P =Base Model+ α(DT 1) + α(DT 2) + ... + α(DT N)

Knowing That : RS2 = Salary - Output , α is the learning rate = 0.5

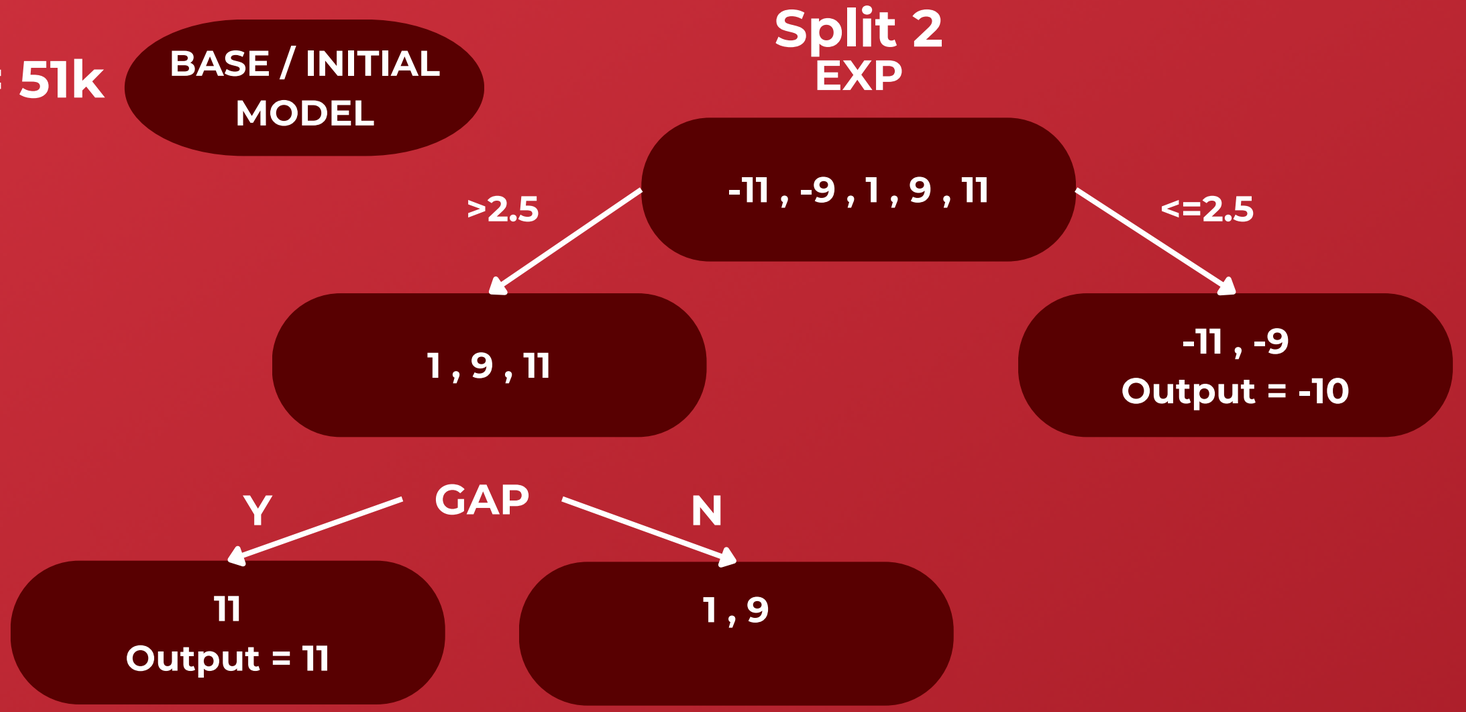
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EX	GP	SY	RS1	O/P	RS2
2	Y	40k	-11k		
2.5	Y	42k	-9k		
3	N	52k	1k		
4	N	60k	9k		
4.5	Y	62k	11k		



- **FINAL STEPS :**
  - Calculate the output of each bottom leaf using :
    Output = Avg of RES = (Sum Residuals)/(Nb Residuals)
  - Calculate the new probability of each record using :
    O/P =Base Model+ α(DT 1) + α(DT 2) + ... + α(DT N)

Knowing That : RS2 = Salary - Output , α is the learning rate = 0.5

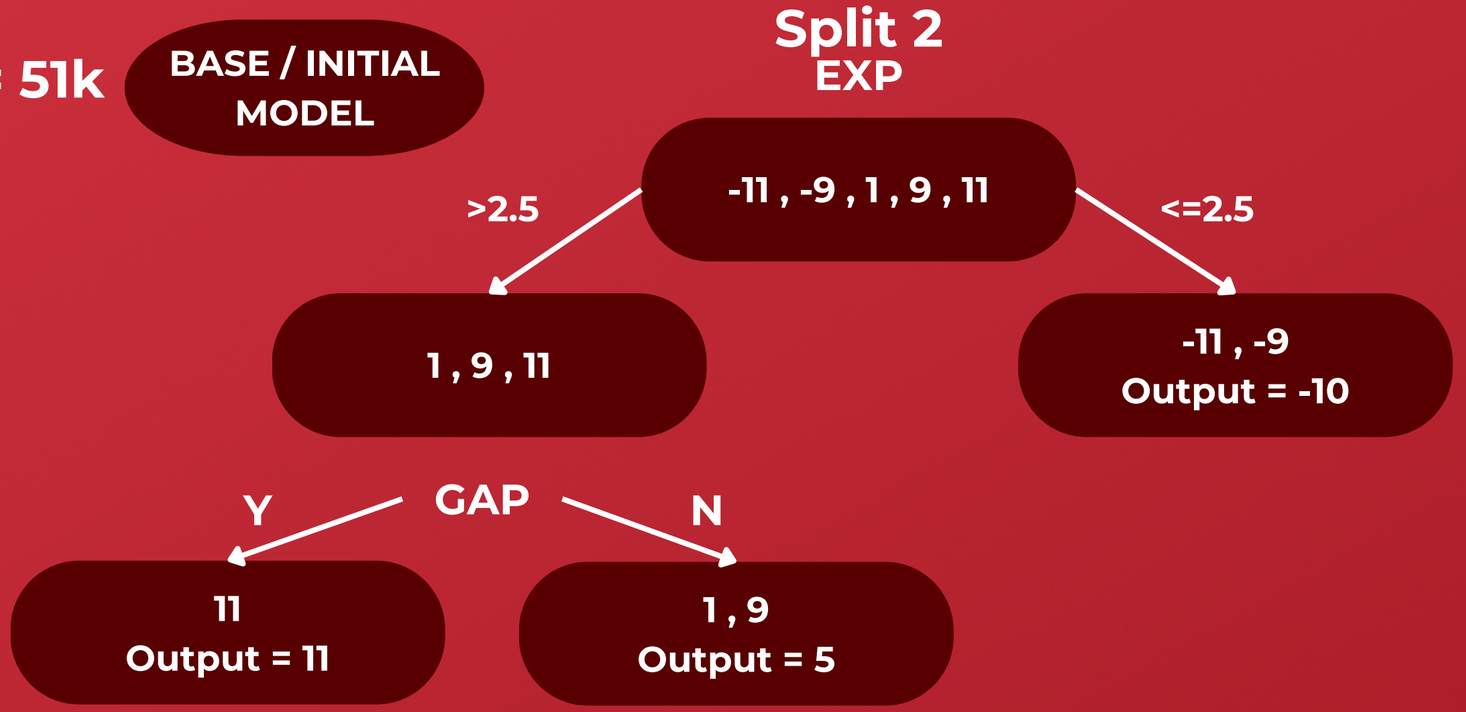
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EX	GP	SY	RS1	O/P	RS2
2	Y	40k	-11k		
2.5	Y	42k	-9k		
3	N	52k	1k		
4	N	60k	9k		
4.5	Y	62k	11k		



- **FINAL STEPS :**
  - Calculate the output of each bottom leaf using :
    Output = Avg of RES = (Sum Residuals)/(Nb Residuals)
  - Calculate the new probability of each record using :
    O/P =Base Model+ α(DT 1) + α(DT 2) + ... + α(DT N)

Knowing That : RS2 = Salary - Output , α is the learning rate = 0.5

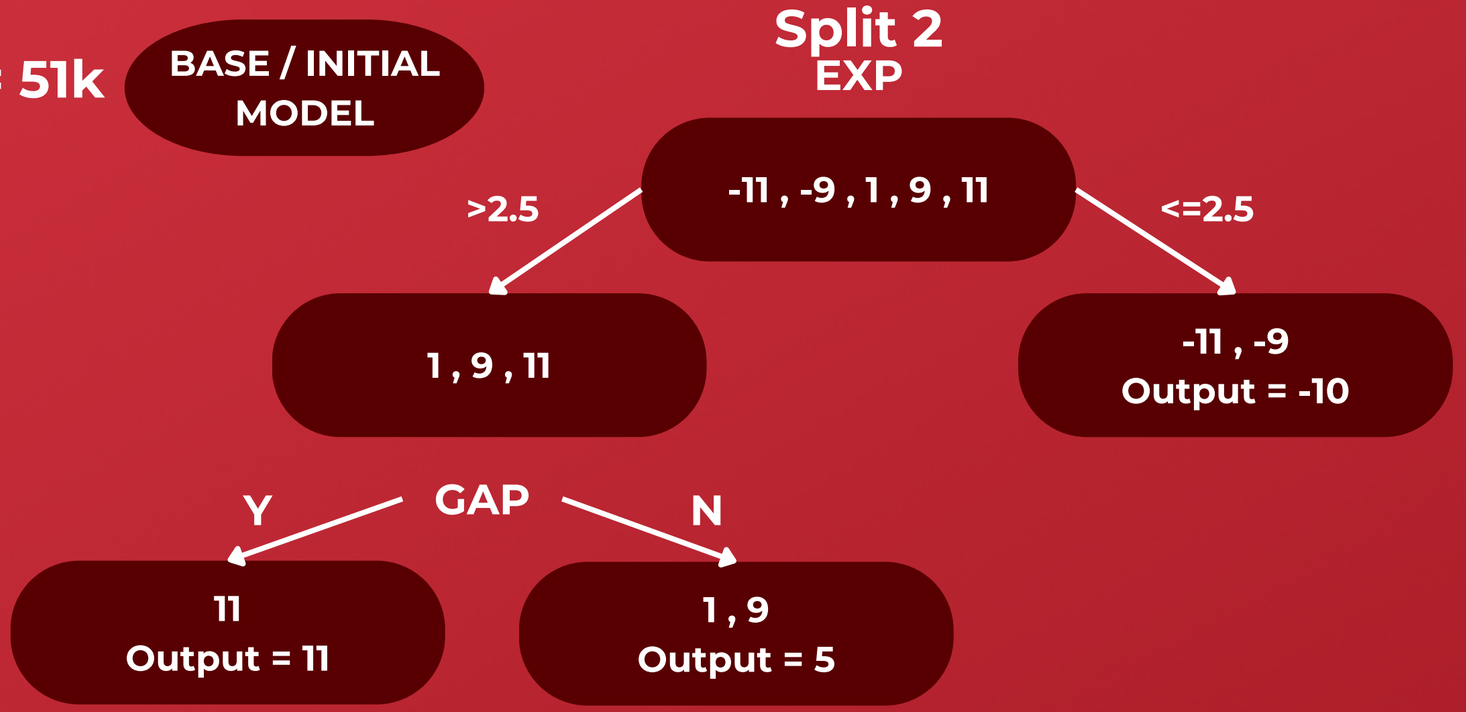
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EX	GP	SY	RS1	O/P	RS2
2	Y	40k	-11k	46k	
2.5	Y	42k	-9k		
3	N	52k	1k		
4	N	60k	9k		
4.5	Y	62k	11k		



- **FINAL STEPS :**
  - Calculate the output of each bottom leaf using :
    Output = Avg of RES = (Sum Residuals)/(Nb Residuals)
  - Calculate the new probability of each record using :
    O/P =Base Model+ α(DT 1) + α(DT 2) + ... + α(DT N)

Knowing That : RS2 = Salary - Output , α is the learning rate = 0.5

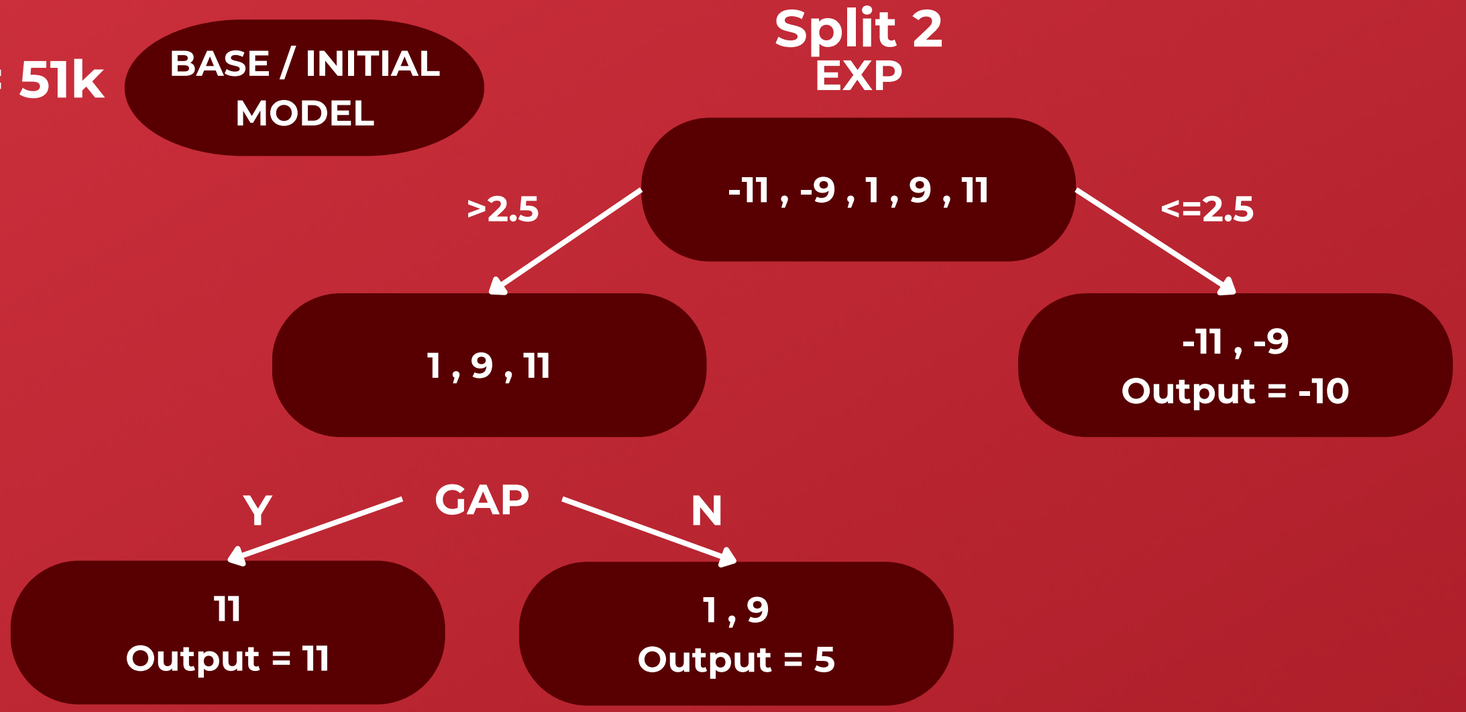
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :** Avg Salary= 51k

EX	GP	SY	RS1	O/P	RS2
2	Y	40k	-11k	46k	
2.5	Y	42k	-9k	46k	
3	N	52k	1k	53.5k	
4	N	60k	9k	53.5k	
4.5	Y	62k	11k	56.5k	



- **FINAL STEPS :**
  - Calculate the output of each bottom leaf using :
    Output = Avg of RES = (Sum Residuals)/(Nb Residuals)
  - Calculate the new probability of each record using :
    O/P =Base Model+ α(DT 1) + α(DT 2) + ... + α(DT N)

Knowing That : RS2 = Salary - Output , α is the learning rate = 0.5

AI CAMP

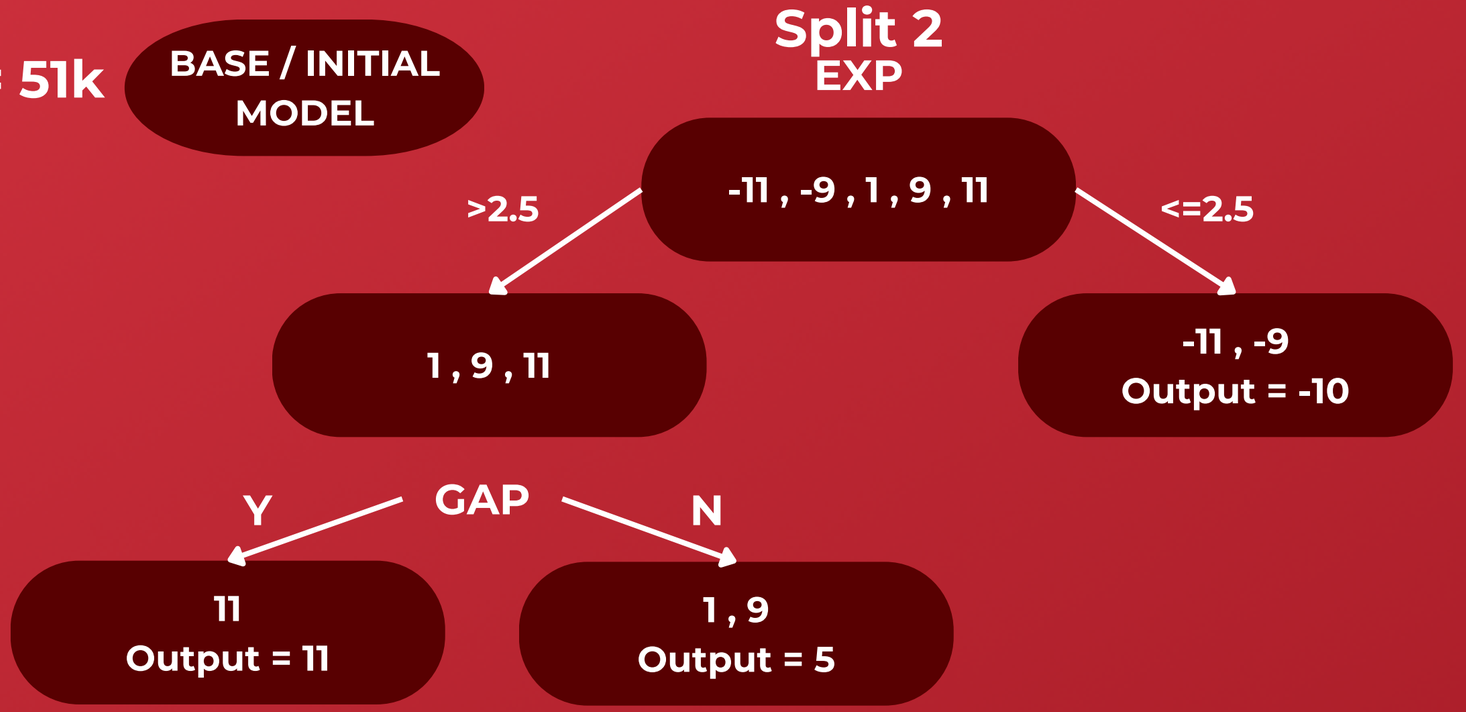
school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :**

Avg Salary= 51k

EX	GP	SY	RS1	O/P	RS2
2	Y	40k	-11k	46k	-6k
2.5	Y	42k	-9k	46k	
3	N	52k	1k	53.5k	
4	N	60k	9k	53.5k	
4.5	Y	62k	11k	56.5k	



- **FINAL STEPS :**
  - Calculate the output of each bottom leaf using :
    Output = Avg of RES = (Sum Residuals)/(Nb Residuals)
  - Calculate the new probability of each record using :
    O/P =Base Model+ α(DT 1) + α(DT 2) + ... + α(DT N)

Knowing That : RS2 = Salary - Output , α is the learning rate = 0.5

AI CAMP

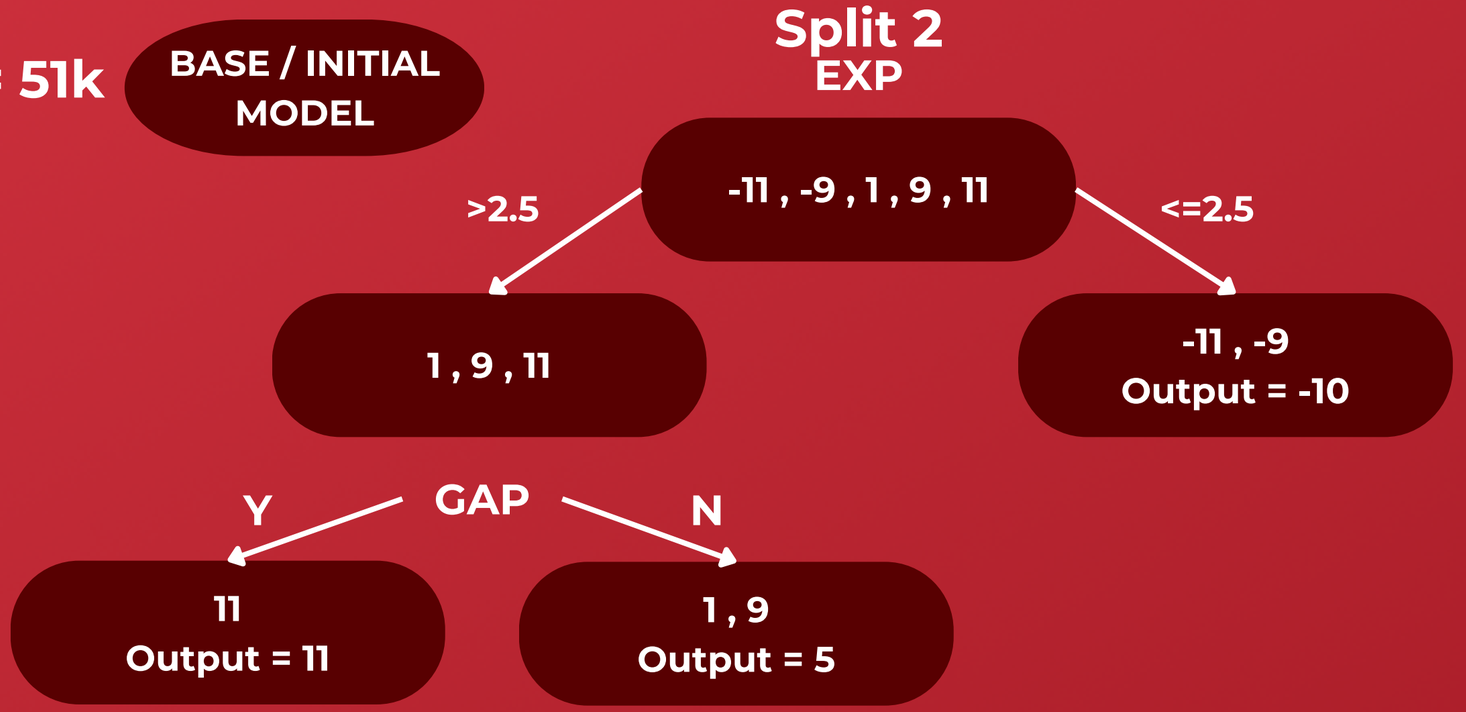
school of ai Algiers

XGBOOST REGRESSION :

- **Dataset :**

Avg Salary= 51k

EX	GP	SY	RS1	O/P	RS2
2	Y	40k	-11k	46k	-6k
2.5	Y	42k	-9k	46k	-4k
3	N	52k	1k	53.5k	-1.5k
4	N	60k	9k	53.5k	6.5k
4.5	Y	62k	11k	56.5k	5.5k



- **FINAL STEPS :**
  - Calculate the output of each bottom leaf using :
    Output = Avg of RES = (Sum Residuals)/(Nb Residuals)
  - Calculate the new probability of each record using :
    O/P =Base Model+ α(DT 1) + α(DT 2) + ... + α(DT N)

Knowing That : RS2 = Salary - Output , α is the learning rate = 0.5

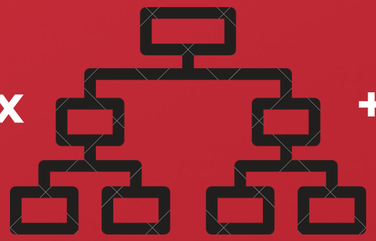
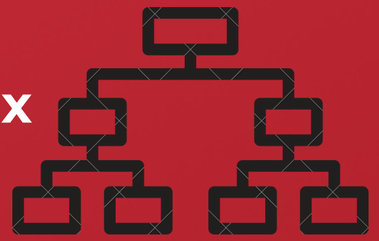
AI CAMP

school of ai Algiers

XGBOOST REGRESSION :

DataSet → BASE / INITIAL MODEL → DT 1 → DT N

Probability = INIT GUESS AVG + α x (DT 1) + ... + α x (DT N)

Probability = BASE / INITIAL MODEL + α x  + ... + α x 

- **FINAL STEPS :**
  - Calculate the output of each bottom leaf using :
    Output = Avg of RES = (Sum Residuals)/(Nb Residuals)
  - Calculate the new probability of each record using :
    O/P =Base Model+ α(DT 1) + α(DT 2) + ... + α(DT N)

Knowing That : RS2 = Salary - Output , α is the learning rate = 0.5

AI CAMP

school of ai Algiers

WHEN SHOULD I USE XGBOOST?

EXTREME GRADIENT BOOSTING WITH XGBOOST

WHEN TO USE XGBOOST :

- You have a large number of training samples
  - Greater than 1000 training samples and less 100 features
  - The number of features < number of training samples
- You have a mixture of categorical and numeric features

WHEN TO NOT USE XGBOOST :

- Image recognition
- Computer vision
- Natural language processing and understanding problems
- When the number of training samples is signicantly smaller than the number of features

AI CAMP

school of ai Algiers

HOW XGBOOST IS IMPLEMENTED IN PYTHON

EXTREME GRADIENT BOOSTING WITH XGBOOST

HOW TO IMPLEMENT XGBOOST :

- **Classification :**

```
from xgboost import XGBClassifier
model = XGBClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

- **Regression**

```
from xgboost import XGBRegressor
model = XGBRegressor()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

AI CAMP

school of ai Algiers

LET'S PRACTICE!

EXTREME GRADIENT BOOSTING WITH XGBOOST

XGBOOST : SOME HELPFUL RSOURCES

- XGBoost Paper : https://arxiv.org/pdf/1603.02754.pdf
- XGBoost Docs : https://xgboost.readthedocs.io

XGBoost: A Scalable Tree Boosting System

Tianqi Chen
University of Washington
tqchen@cs.washington.edu

Carlos Guestrin
University of Washington
guestrin@cs.washington.edu

ABSTRACT

Tree boosting is a highly effective and widely used machine learning method. In this paper, we describe a scalable end-to-end tree boosting system called XGBoost, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges. We propose a novel sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning. More importantly, we provide insights on cache access patterns, data compression and sharding to build a scalable tree boosting system. By combining these insights, XGBoost scales beyond billions of examples using far fewer resources than existing systems.

problems. Besides being used as a stand-alone predictor, it is also incorporated into real-world production pipelines for ad click through rate prediction [15]. Finally, it is the de-facto choice of ensemble method and is used in challenges such as the Netflix prize [3].

In this paper, we describe XGBoost, a scalable machine learning system for tree boosting. The system is available as an open source package[2]. The impact of the system has been widely recognized in a number of machine learning and data mining challenges. Take the challenges hosted by the machine learning competition site Kaggle for example. Among the 29 challenge winning solutions [3] published at Kaggle's blog during 2015, 17 solutions used XGBoost. Among these

AI CAMP

school of ai Algiers

AI CAMP

school of ai Algiers

Thank you for attending

any questions ?