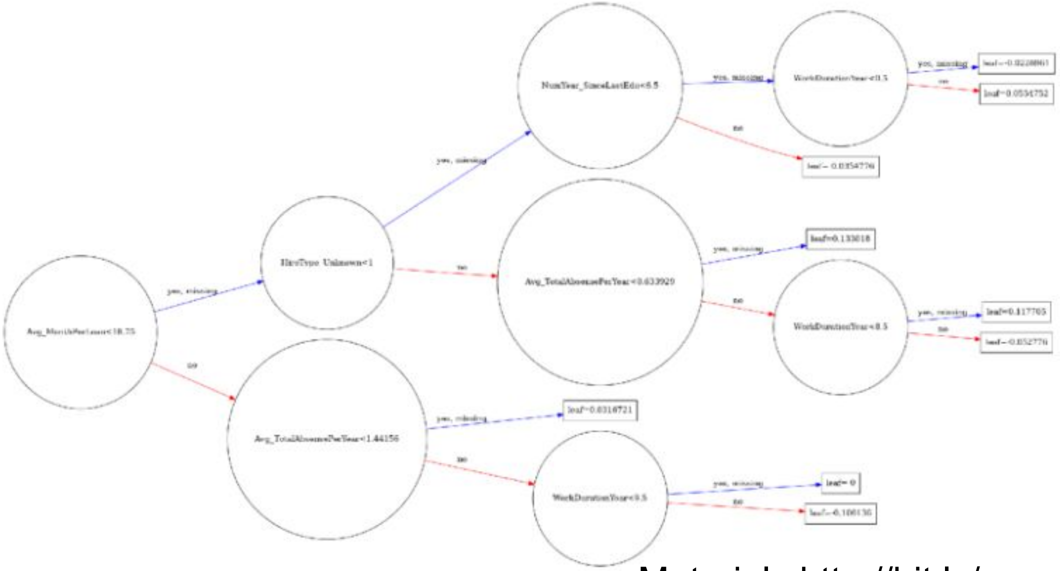
Boost Trees, Calibration, and Explainability



Materials http://bit.ly/exxon2-2

Agenda

Linear regression

Decision Trees

- Categorical features

Calibration and confidence

Explainability

Predicting the amount of rainfall

https://esan108.com/%E0%B8%9E%E0%B8%A3%E0%B8%B0%E0%B9%82%E0%B8%84%E0%B8%81%E0%B8%B4%E0%B8%99%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B8%AB%E0%B8%A1%E0%B8%B2%E0%B8%A2%E0%B8%96%E0%B8%B6%E0%B8%87.html

Predicting the amount of rainfall

Cloth	Corn	Grass	Water	Beer	Rainfall
4	6	3	10	0	**76950**
5	1	0	0	7	**30234**
6	0	3	5	7	**123456**
5	0	3	12	0	**89301**
4	3	0	6	7	**?**

$$h_{\boldsymbol{\theta}}(\mathbf{x_1}) = \theta_0 + \theta_1 x_{1,1} + \theta_2 x_{1,2} + \theta_3 x_{1,3} + \theta_4 x_{1,4} + \theta_5 x_{1,5}$$

Where θs are the parameter of the model

Xs are values in the table

(Linear) Regression

$h_{\boldsymbol{\theta}}(\mathbf{x}_1) = \theta_0 + \theta_1 x_{1,1} + \theta_2 x_{1,2} + \theta_3 x_{1,3} + \theta_4 x_{1,4} + \theta_5 x_{1,5}$

We can rewrite

Assume x_0 is always 1

n is dimension of x

$$h_\theta(\mathbf{x}_i) = \Sigma_{j=0}^{n} \theta_j x_{i,j}$$

Picking θ

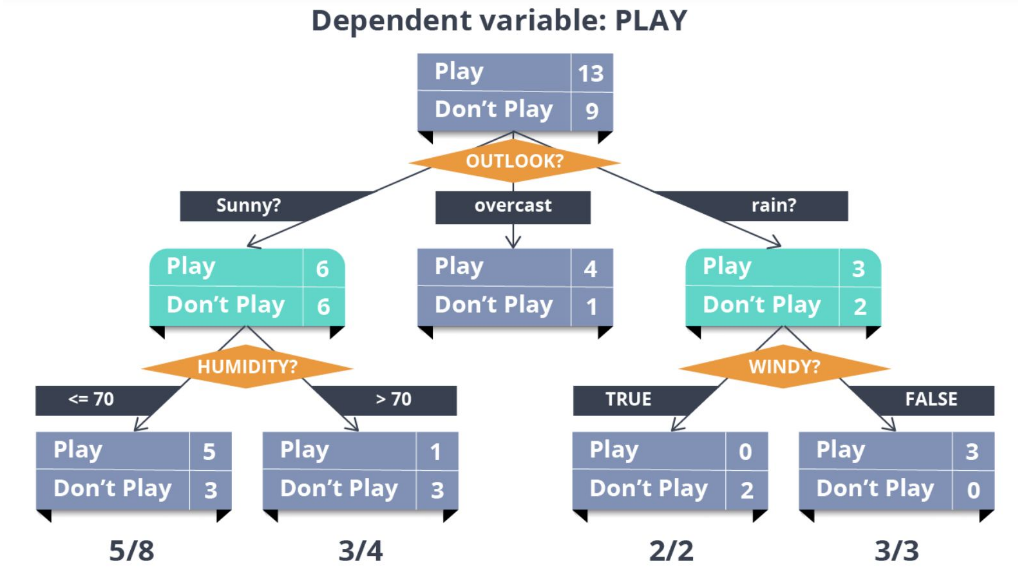
$$h_\theta(\mathbf{x}_i) = \Sigma_{j=0}^{n} \theta_j x_{i,j} = \theta^T \mathbf{x}_i$$

- Random until you get the best performance?
  - *Solve by gradient descent (next lecture)*

DECISION TREES

Decision Trees

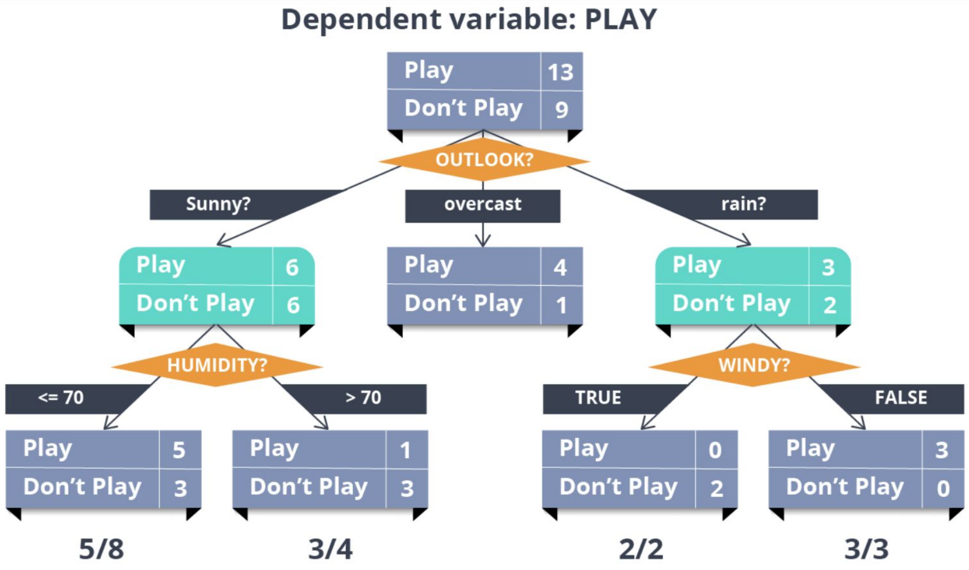
A tree structure that separates data into groups by the feature attributes
Can be used for classification and regression



https://www.edureka.co/blog/decision-trees/

What’s a good decision tree?

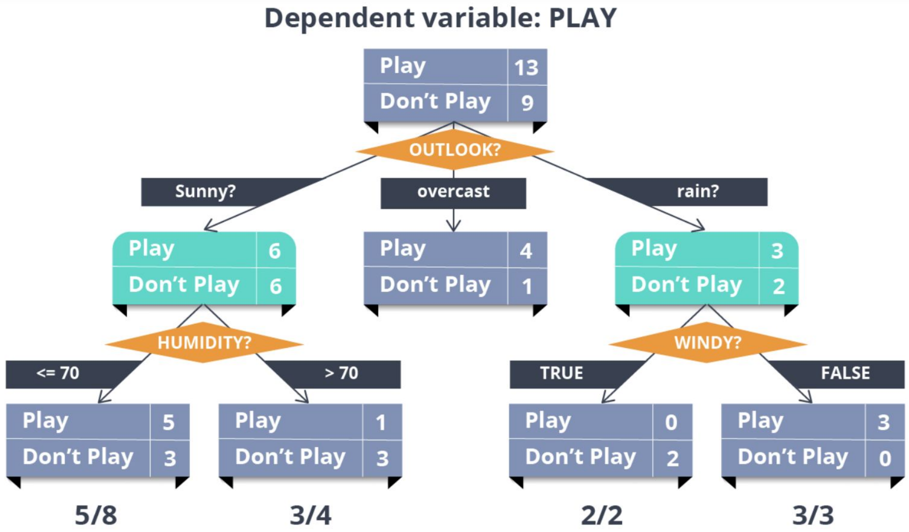
Separates the data nicely
Within a certain budget (smaller trees) - less overfitting



How to create a good decision tree?

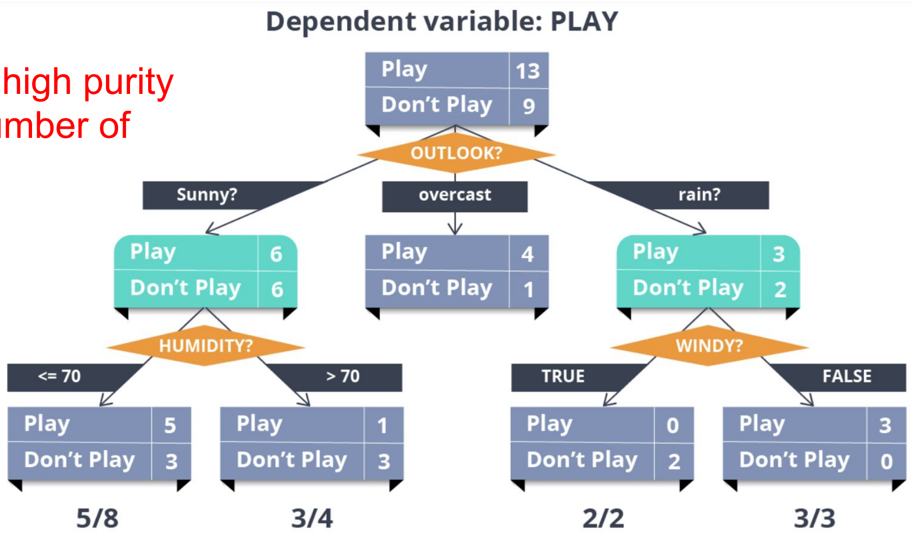
Pick the attribute that best separates the classes
Keep doing it until a leave contains entirely one class or you decide it’s not worth it to add more nodes

How to determine the best attribute automatically?



Purity (Entropy)

Want trees that give high purity with the minimum number of nodes



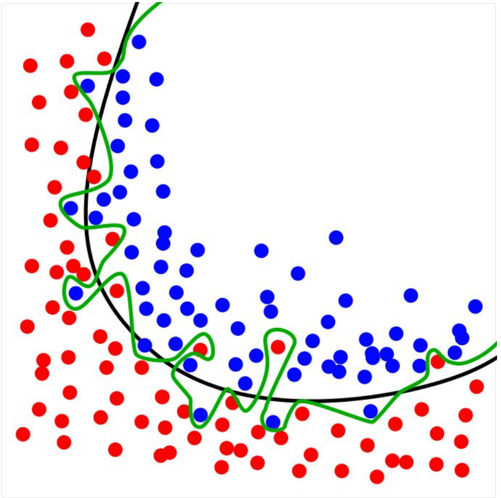
Low purity or high entropy

High purity or low entropy

Problems with Decision Trees

Can overfitting easily

Susceptible to noise or badly labelled data



TREE ENSEMBLE MODEL

Tree ensemble model

Ensemble types are models that combine multiple models together

A group of experts voting on a subject

Can lead to less overfitting

Tree ensemble = Multiple trees = Random Forest!

Bagging

Create multiple subsets of data
Each subset is used to train a different tree
The final answer is the average or mode

Less overfitting and can handle mislabeled data

Random Forest

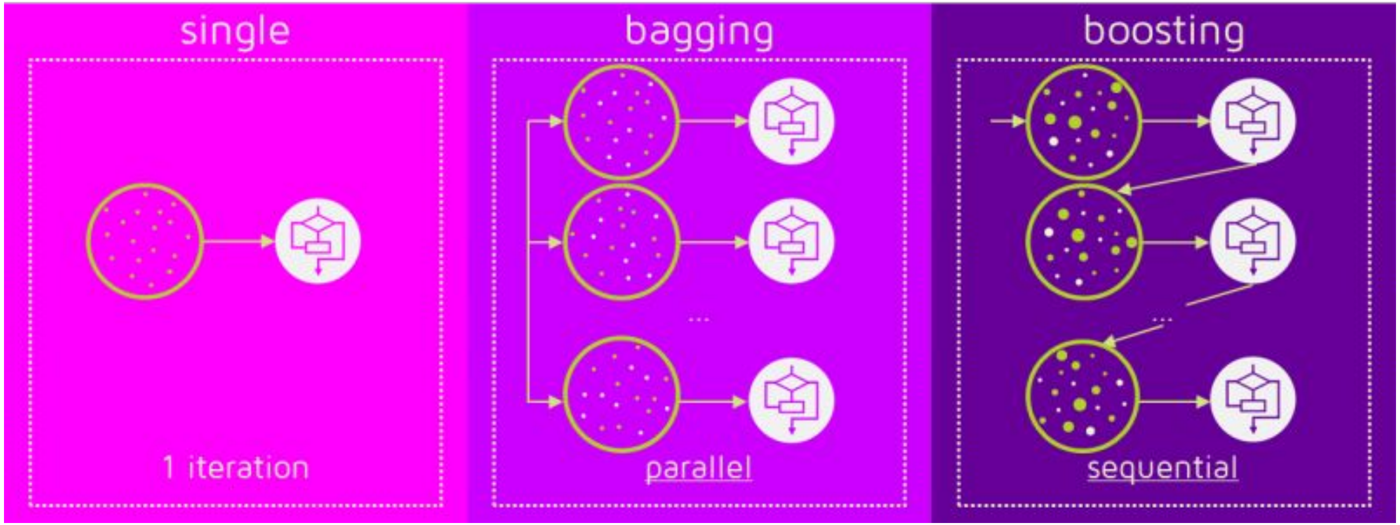
We can also use bagging on features

Each tree has different training samples AND set of features

Boosting vs Bagging

Boosting is another way to create multiple trees

But boosting is iterative, the next tree is based on the errors from the previous trees



https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/

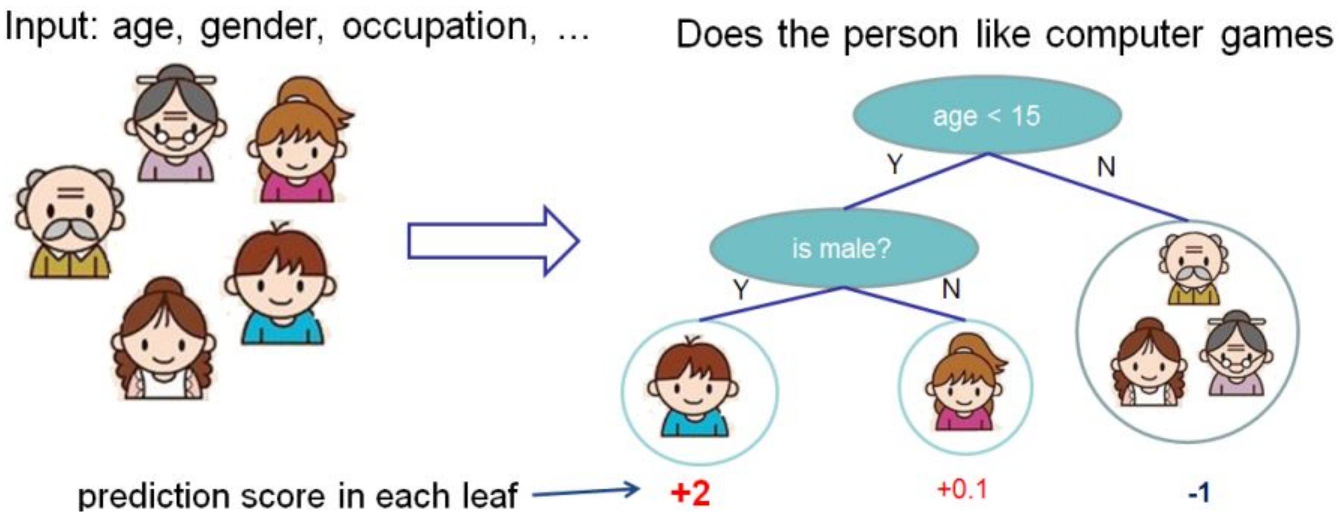
Gradient Boosting

A method of boosting that use gradient-based methods

Tree Gradient Boosting

Similar to decision tree

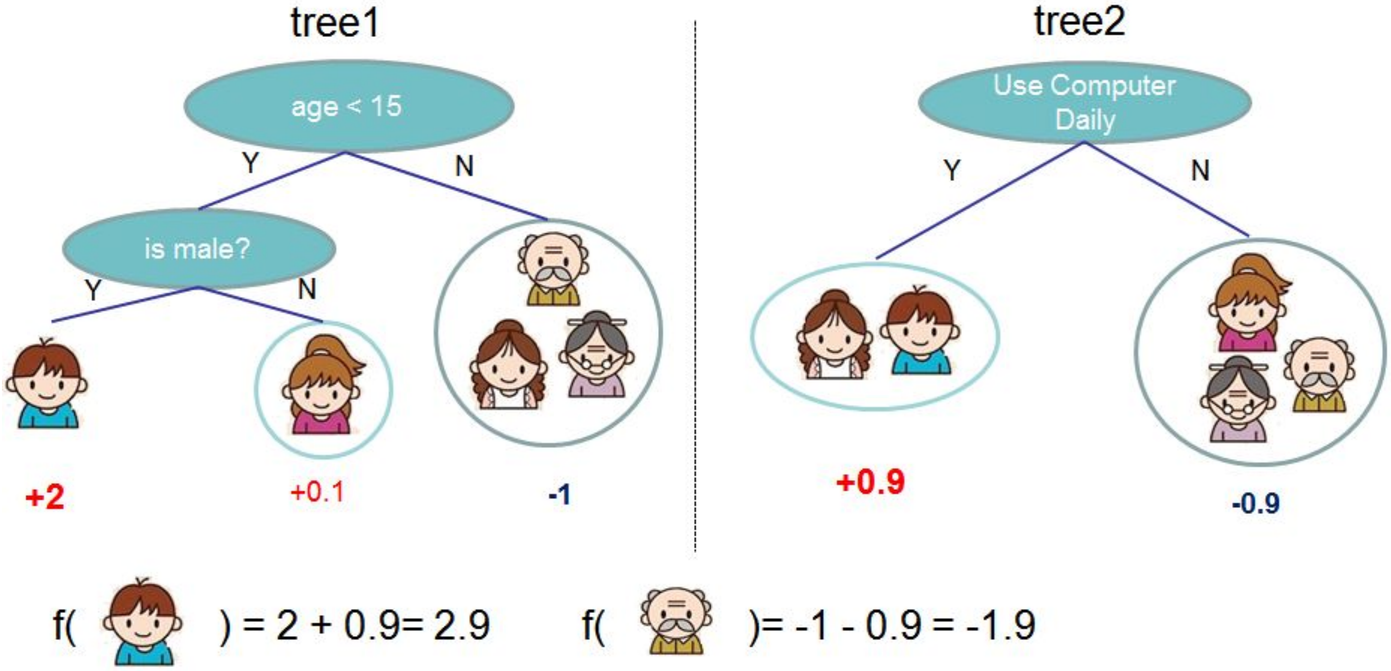
Difference is the leaf node contains a score



https://xgboost.readthedocs.io/en/latest/tutorials/model.html

Tree Gradient Boosting

Multiple trees with different rules. The subsequent tree try to correct the errors from the previous trees



Extreme Gradient Boosting (XGBoost)

Super popular Tree Boosting library

Highly recommended for spreadsheets type of input data

```
model = XGBClassifier(
    n_jobs=16,
    n_estimators=400,
    max_depth=4,
    objective="binary:logistic",
    learning_rate=0.07,
    subsample=0.9,
    min_child_weight=6,
    colsample_bytree=.9,
    scale_pos_weight=0.8,
    gamma=8,
    reg_alpha=6,
    reg_lambda=1.3)
```

Objective <- type of problem you want to solve

Max_depth <- max depth of tree, higher more overfitting
Min_child_weight <- how strong must the leave be, higher less overfitting
Gamma <- when to stop splitting early

Reg_alpha, reg_lambda <- reduce overfitting

Scale_pos_weight <- weight for class imbalance

https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

Notes on feature encoding

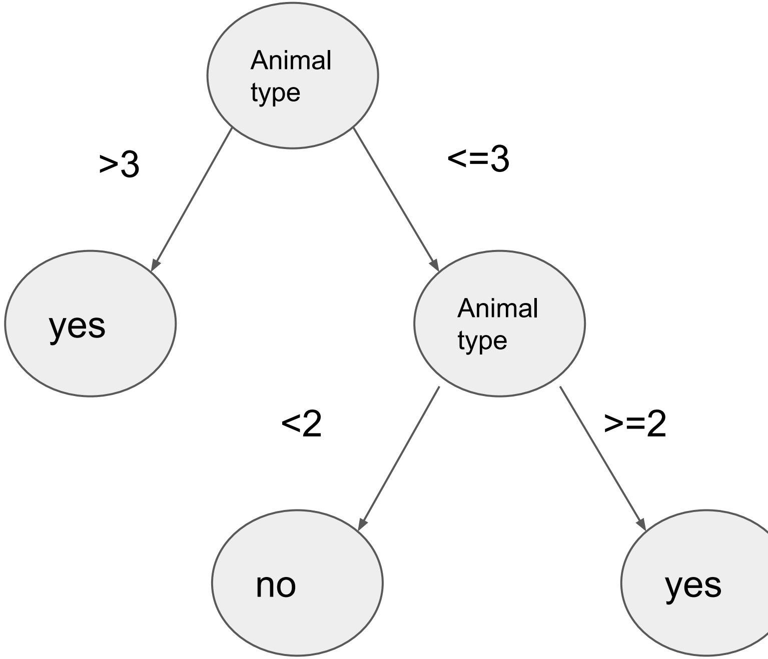
Categorical features does not mean anything

Type of animal

Animal type =
- 1 if mouse
- 2 if bird
- 3 if dog
- 4 if insect

Makes it hard to do decision trees

Is it green?



One hot encoding

Split categorical features into multiple binary features

Type of animal (as one hot)

Is_mouse = (0,1)

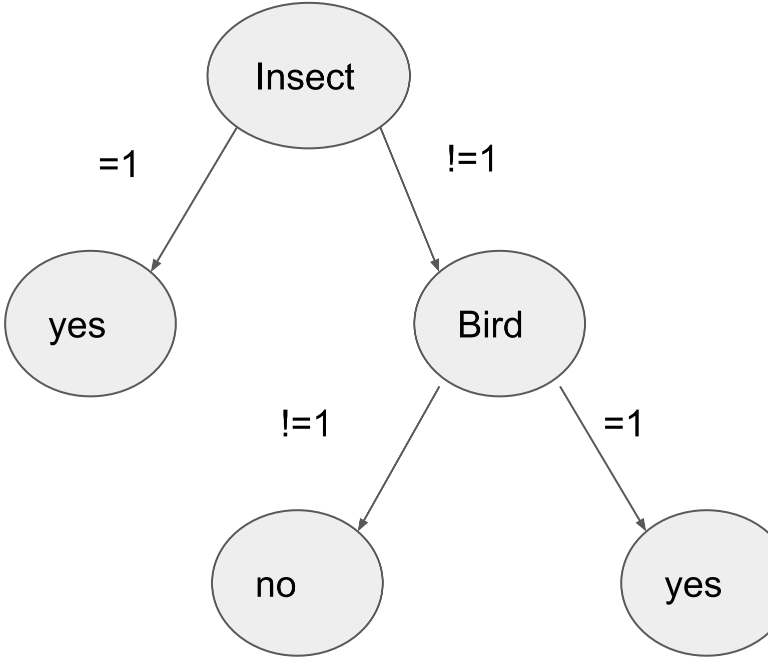
Is_bird = (0,1)

Is_dog = (0,1)

Is_insect = (0,1)

Doesn't change much

Is it green?



Target encoding

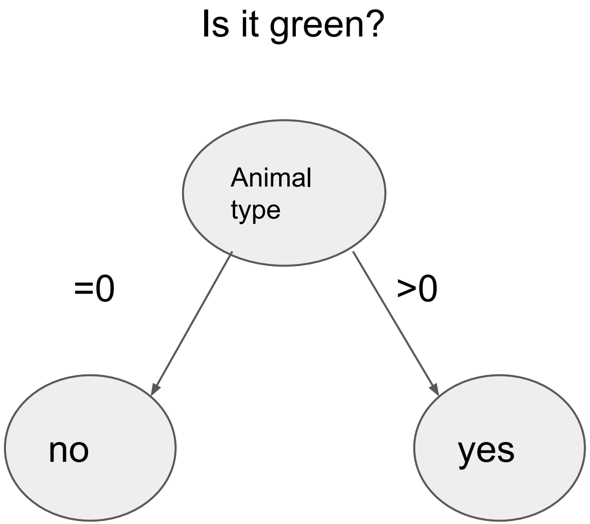
Encode information by looking at how the feature correlates with the final answer

Encoded feature = P(answer = yes| feature value)

Animal type =
- 0 if mouse
- 0.3 if bird
- 0 if dog
- 0.5 if insect

Need some further smoothing to improve this.

https://dl.acm.org/citation.cfm?id=507538



Other XGBoost variants

LightGBM

CatBoost

Different ways to handle categorical encoding.

Different ways to do node splitting (faster)

https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db

Agenda

Linear regression

Decision Trees

- Categorical features

Calibration and confidence

Explainability

Confidence

ML products

Customer facing

Recommendation systems, maps (traffic estimate), speech2text

Best guess by the model
Mostly automatic (check deposit by app)

Internal facing

Loan applications, demand forecasting

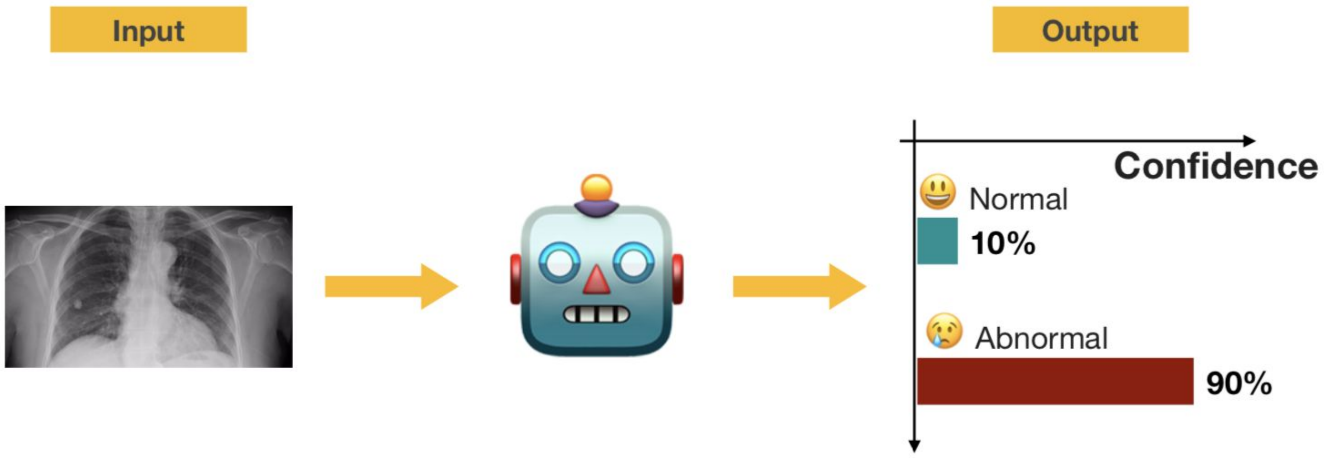
Can say “I’m not sure”
Human in the loop. Machine-assisted

Requires confidence level

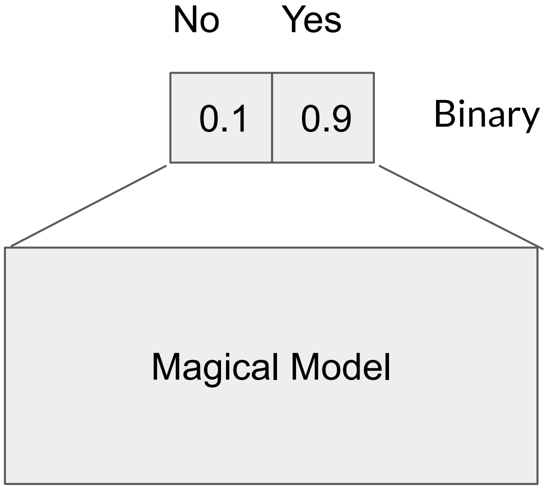
Confidence score

Practical models are not only accurate, but need to be able to state its confidence

Confidence = probability of being correct



A naive way to give confidence score



Binary probability values

Confidence level = 0.9?

Is this person smiling?

What about regression task?

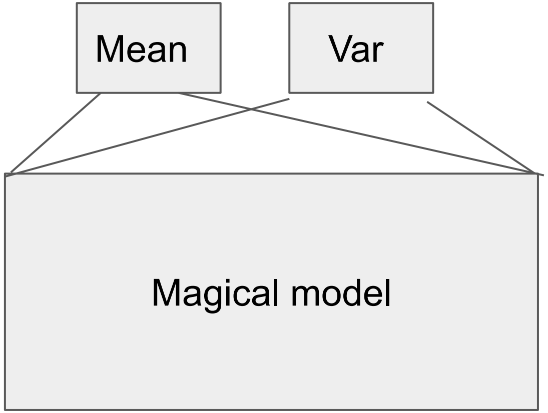
18

Magical Model

Confidence level = ???

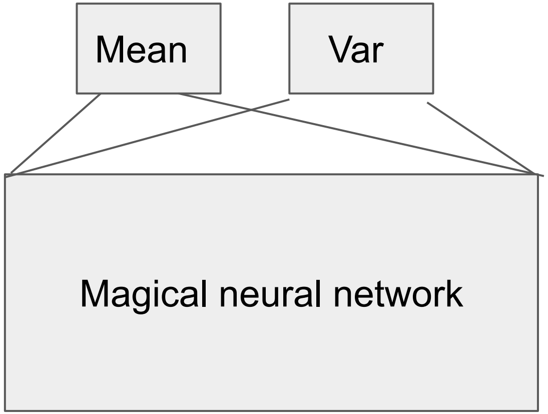
What is the age of this person?

A Naive way for regression (1994!)



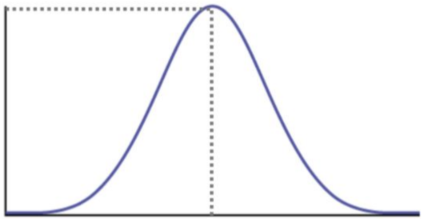
Estimating the mean and variance of the target probability distribution. IEEE 1994

A Naive way for regression (1994!)



Confidence score = $\frac{1}{\sqrt{2\pi\sigma^2}} e^{\overset{0}{\cancel{\frac{-(\mu-\mu)^2}{2\sigma^2}}}}$

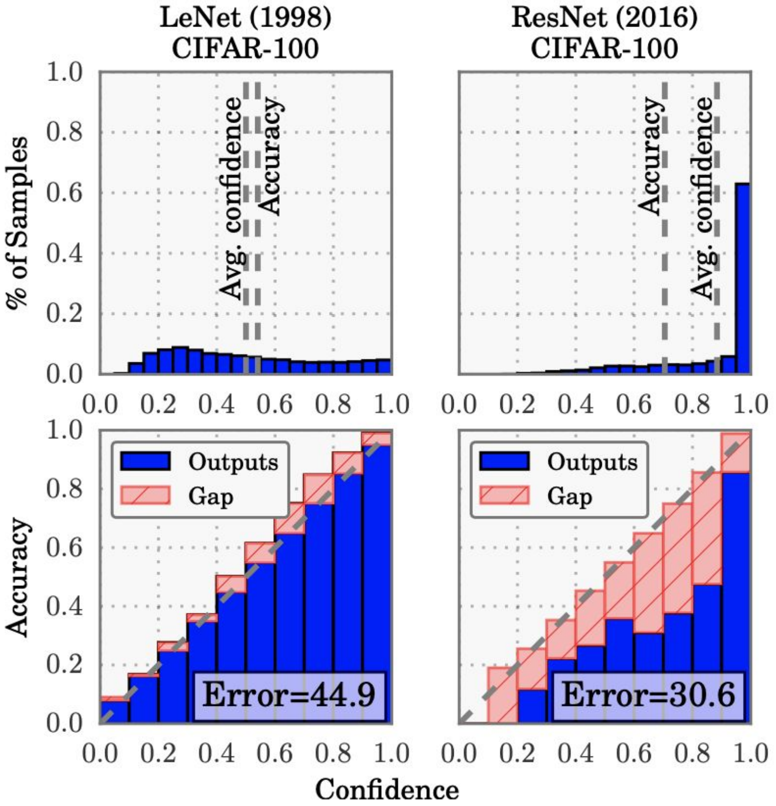
$$= \frac{1}{\sqrt{2\pi\sigma^2}}$$



Predicted value = $\mu(x)$

Estimating the mean and variance of the target probability distribution. IEEE 1994

Poorly calibrated confidence



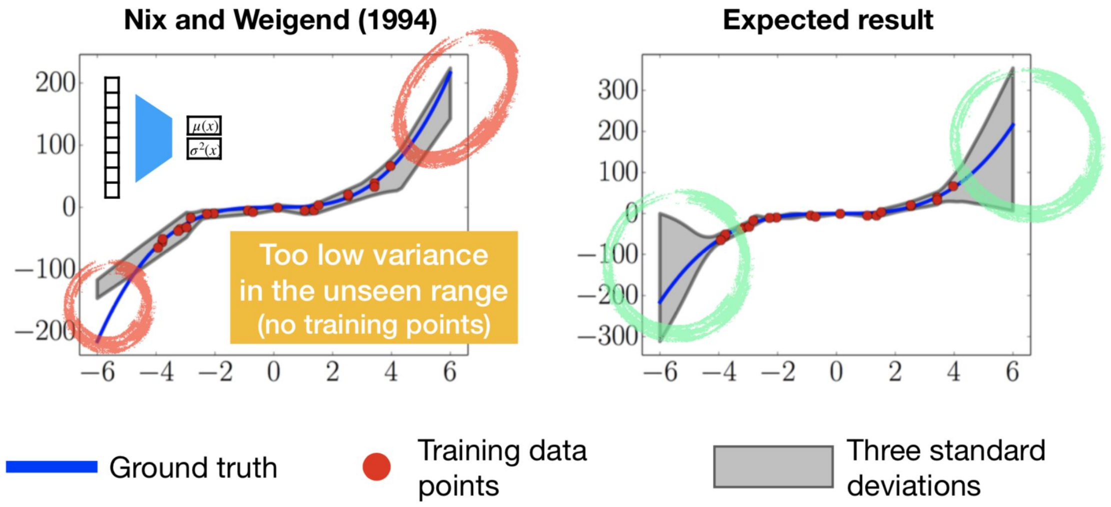
ML models are always overconfident!

Confidence = Probability of being correct

On Calibration of Modern Neural Networks JMLR 2017

Out of distribution problem

Expect high uncertainty or high variance in unseen input range



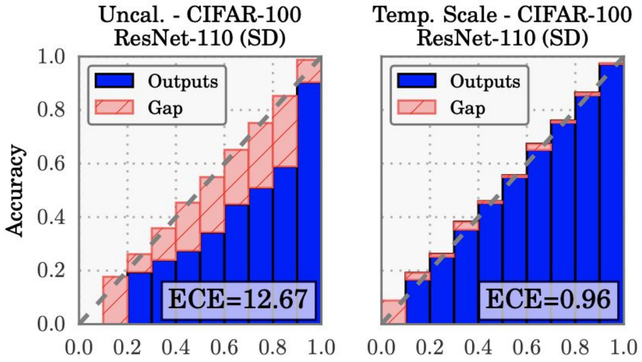
Simple and scalable predictive uncertainty estimation using deep ensembles. 2017

Model calibration

Make the confidence output follows the probability of being correct.

How?

Need a seperate training set to train the calibration (calibration set)



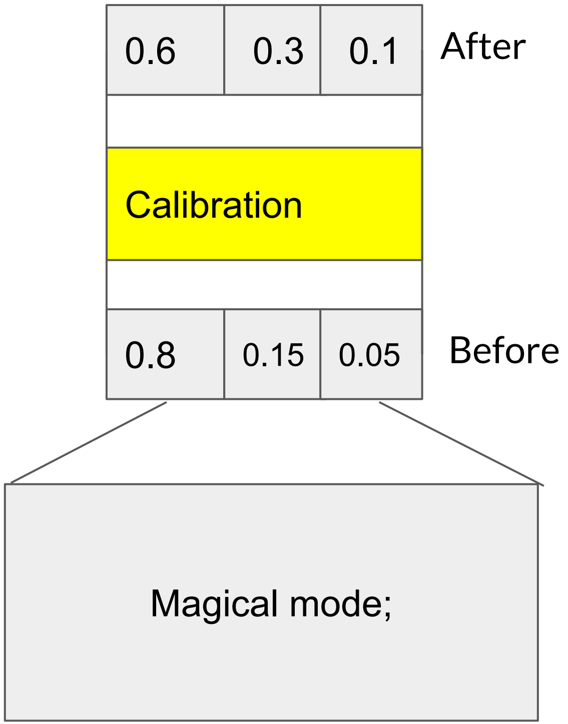
Overview of methods

1 Calibration

2 Ensemble

Calibration

Post processing after model output



Calibration - Temperature scaling

- Given the logic vector **z**, the calibrated prediction is

$$\mathbf{q} = \sigma_{sm}(\frac{\mathbf{z}}{T}) \quad \sigma_{sm}(Z)_i = \frac{exp(z_i)}{\sum_i exp(z_i)}$$

where ***T*** > 0 is a positive scalar, called temperature

- ***T*** is tuned to minimize negative log likelihood (NLL) in val. dataset.

	P1	P1	P3
T	0.8	0.15	0.05
10	0.35	0.33	0.32
2	0.41	0.30	0.29
0.5	0.67	0.18	0.15
0.01	1.00	5.90E-29	2.68E-33

Other calibration methods

Histogram binning

Bayesian binning into quartiles (BBQ)

Matrix and vector scaling (model on top of model)

Isotonic regression (model on top of model)

Try different methods on your dataset. No absolute best.

Overview of methods

1 Calibration

2 Ensemble

Combining models

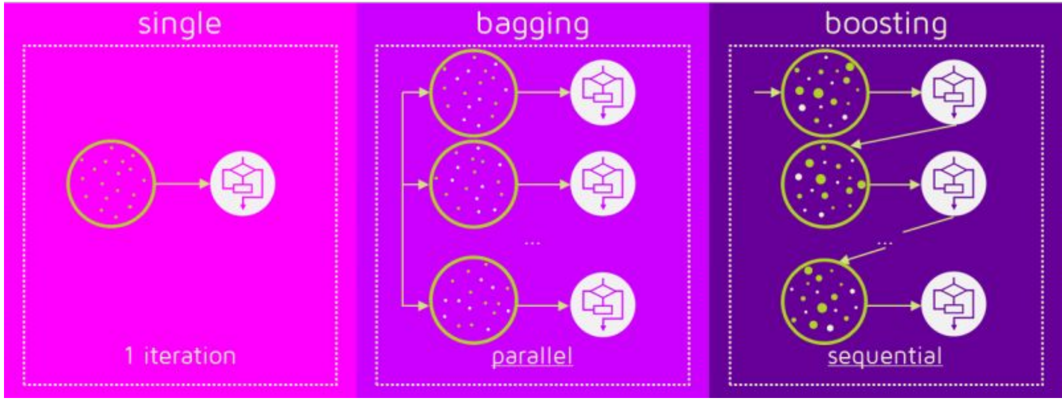
Create multiple models

Calculate mean and variance of the answers!

Multiple models can be just from baggings

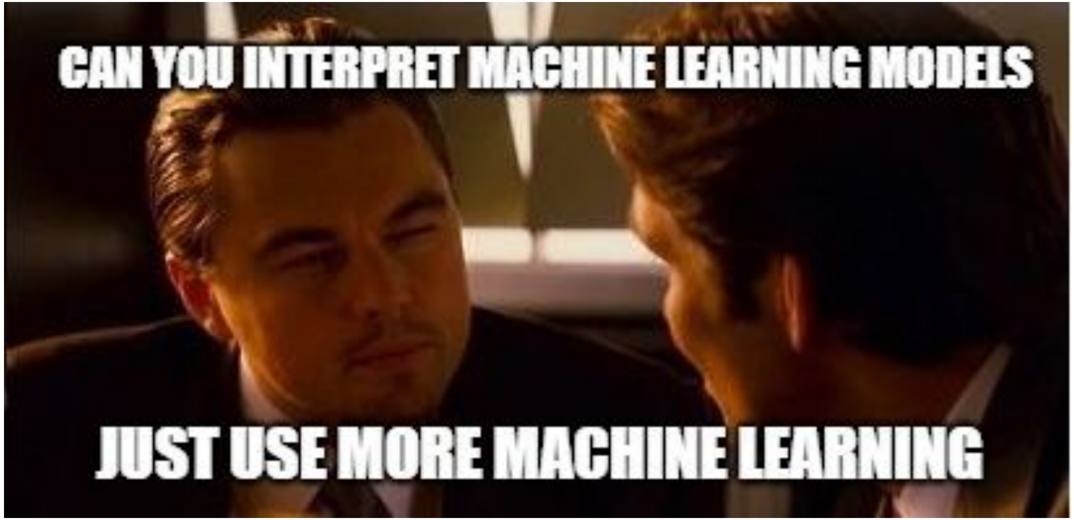
For neural networks, it can be models with different initializations

Cons: need to keep a lot of models around



"Simple and scalable predictive uncertainty estimation using deep ensembles." 2017.

So we got the confidence, can we say why?



Two levels of understanding

Model level

- Describes the model tendency
- Talks about the behavior on training data

Output level

- Attributes model decision for a given test sample to different features

Model level: Simple example

Linear regression

$$h_{\boldsymbol{\theta}}(\mathbf{x}_1) = \theta_0 + \theta_1 x_{1,1} + \theta_2 x_{1,2} + \theta_3 x_{1,3} + \theta_4 x_{1,4} + \theta_5 x_{1,5}$$

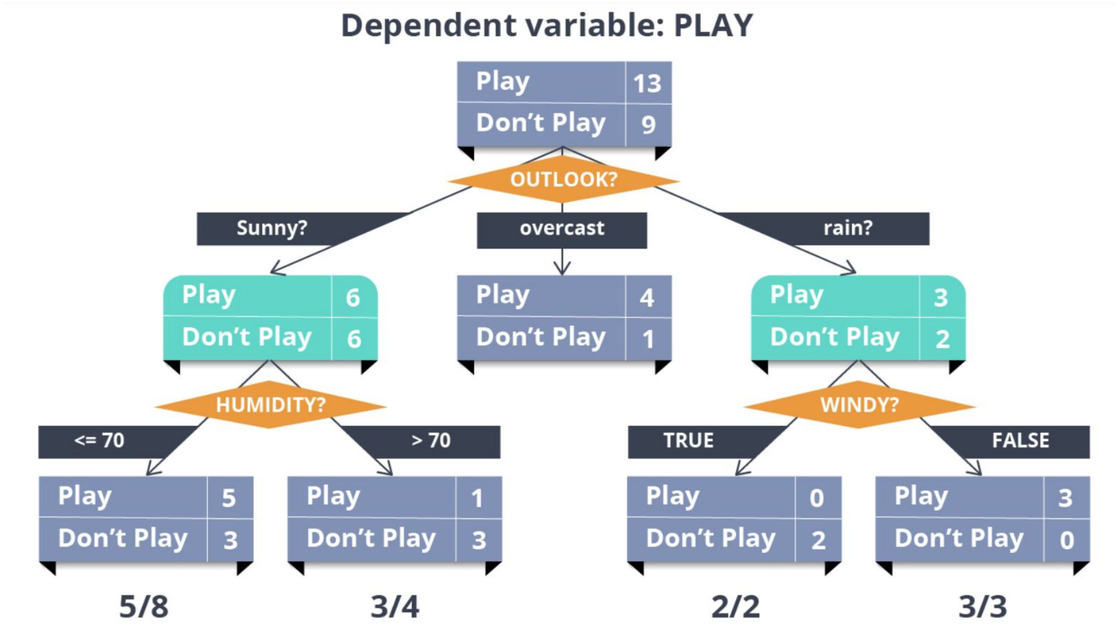
$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i,$$

You can say which feature is important from size of the coefficients.

Pros: Simple, easy to understand

Cons: Only model linear effects

Model level: tree-based (feature importance in XGBoost)



https://www.edureka.co/blog/decision-trees/

Assign scores based on how the features are use

A node that splits better (better purity at children): high score

A node with more training samples: high weight

Usefulness of model level

Feature selection

Gives you confidence that the model is learning reasonable things

Two levels of understanding

Model level

- Describes the model tendency
- Talks about the behavior on training data

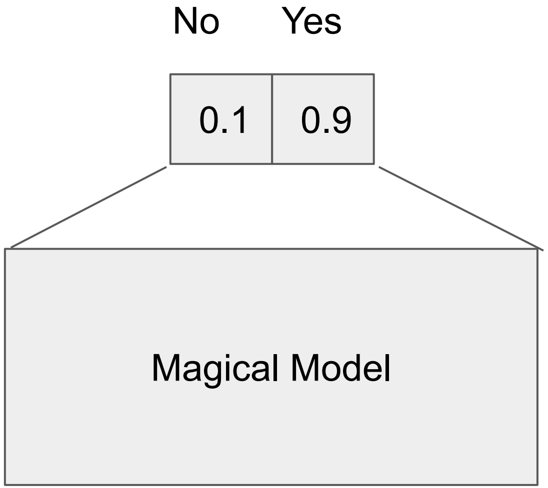
Output level

- Attributes model decision for a given test sample to different features

Output level: key ideas

A feature is important if I tweak the feature and the output change a lot.

Talks about a particular input example

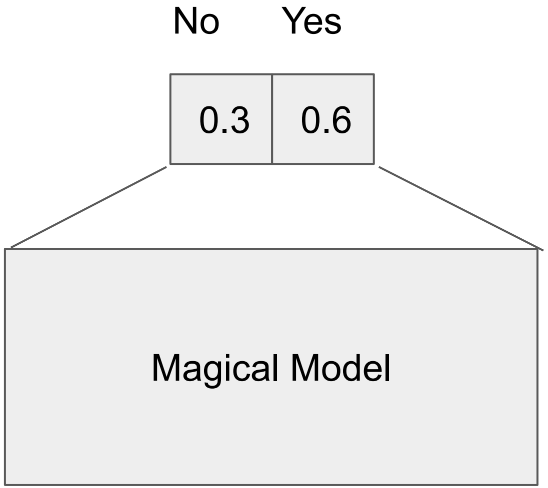


Is this person smiling?

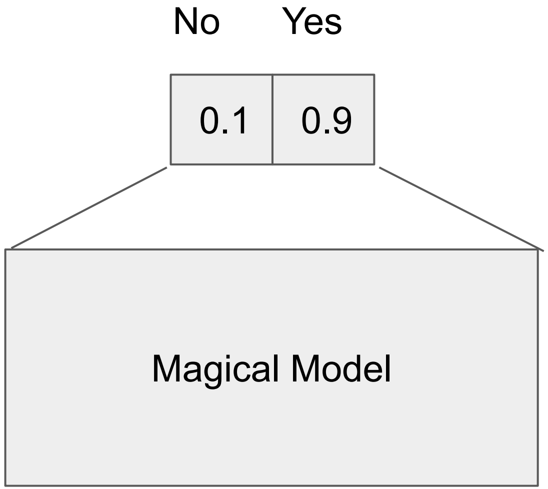
Output level: key ideas

A feature is important if I tweak the feature and the output change a lot.

Talks about a particular input example

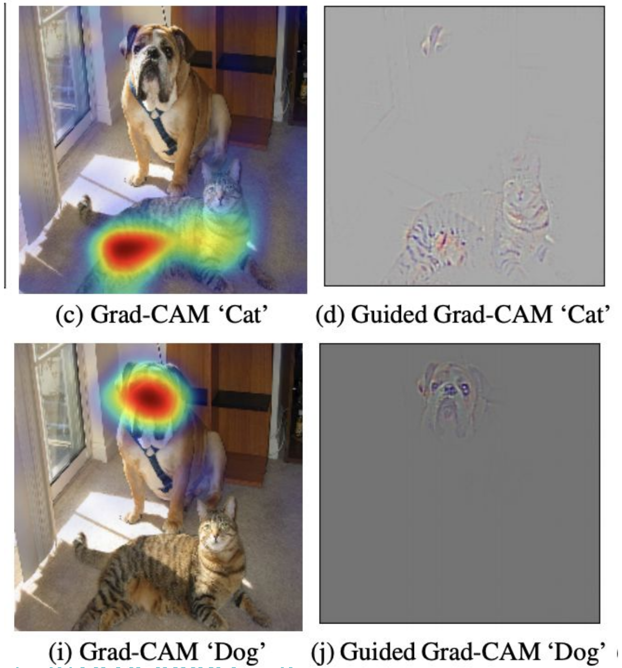


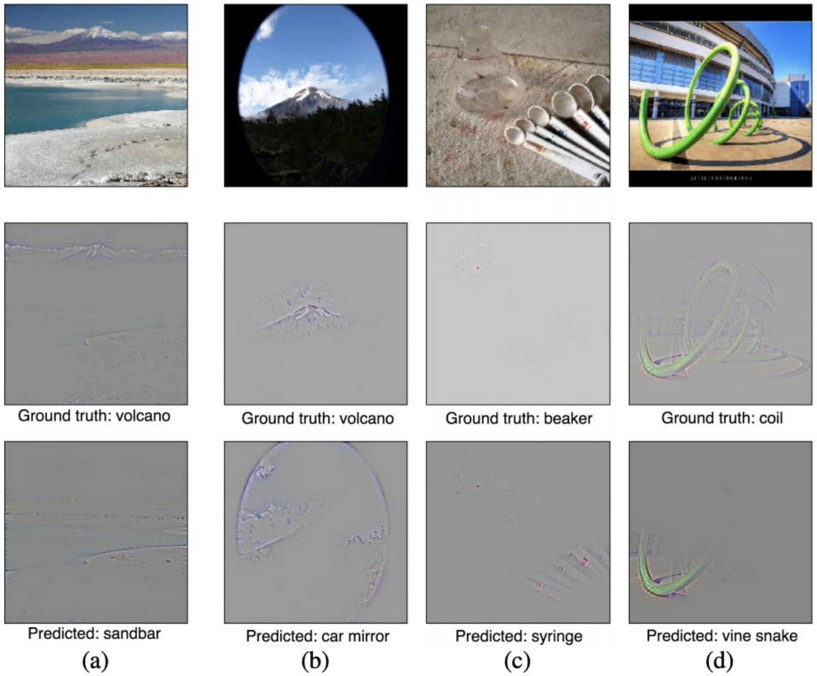
Is this person smiling?



Is this person smiling?

Output level: Gradient-weighted Class Activation Mapping (Grad-CAM)





https://arxiv.org/pdf/1610.02391.pdf
https://github.com/jacobgil/keras-grad-cam

Output level: key ideas

A feature is important if I tweak the feature and the output change a lot.

Mostly useful for images types

Have a simpler model (**surrogate model**) explains the complicated model.

Converting things to linear regression

Additive feature attribution

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i,$$

Simplified input features z’ have binary values

z’ can recover original features x, $x = h_x(z')$ (This mapping depends on x)

Goal: make $g(z') = f(h_x(z'))$. Then we can explain f in terms of simplified features. **(Solved by optimization)**

Example of simplified inputs

Original x = Bag of words features (counts)

There is a black cat and a white cat.

$z = 0$ if count is 0

$z = 1$ if count is not 0

$h_x(z) = x$ if $z = 1$

$h_x(z) = 0$ if $z = 0$

There	1	→	There	1
is	1		is	1
a	2		a	1
black	1		black	1
cat	2		cat	1
and	1		and	1
white	1		white	1
dog	0		dog	0

Example of simplified inputs

Original x = image

z = 0 if patch is not present

z = 1 if patch is present

$h_x(z)$ = x if, z=1

$h_x(z)$ = x with missing patch, if z=0

Additive feature attribution

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i,$$

Many methods fall is additive feature attribution. For example LIME

Feature	Value
odor=foul	True
gill-size=broad	True
stalk-surface-above-ring=silky	True
spore-print-color=chocolate	True
stalk-surface-below-ring=silky	True

https://github.com/marcotcr/lime

Introducing SHAP

SHAP is also an additive feature attribution, but gives credit to **expected** attribution

Expected attribution

Original x = image

z = 0 if patch is not present

z = 1 if patch is present

$h_x(z)$ = x if, z=1

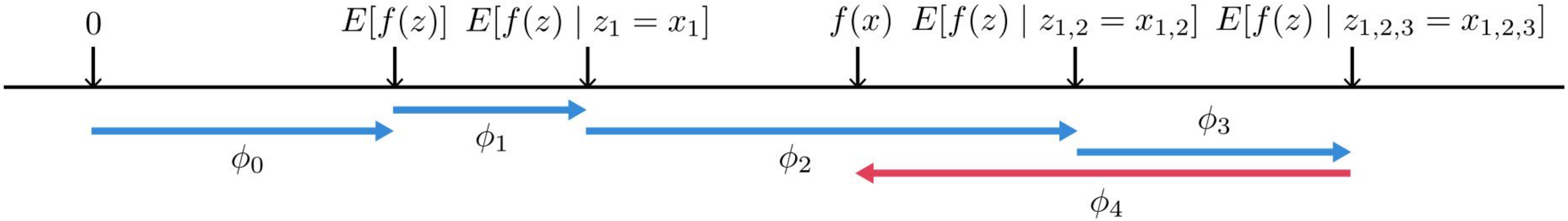
$h_x(z)$ = x with missing patch, if z=0

This simplified input talks about this patch.

What happens if we change the other inputs?

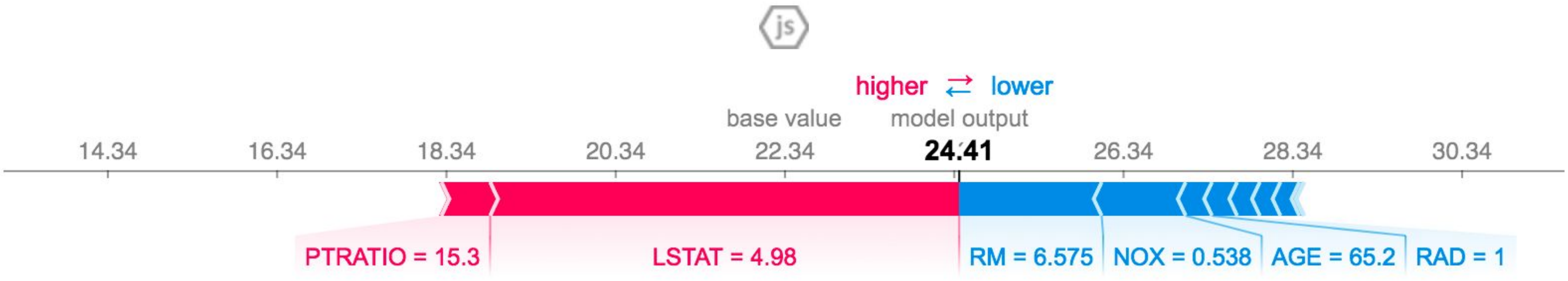
Expected contribution talks about the contribution of the feature regardless of the other contributions

SHAP


0
$E[f(z)]$
$E[f(z) \mid z_1 = x_1]$
$f(x)$
$E[f(z) \mid z_{1,2} = x_{1,2}]$
$E[f(z) \mid z_{1,2,3} = x_{1,2,3}]$
ϕ_0
ϕ_1
ϕ_2
ϕ_3
ϕ_4

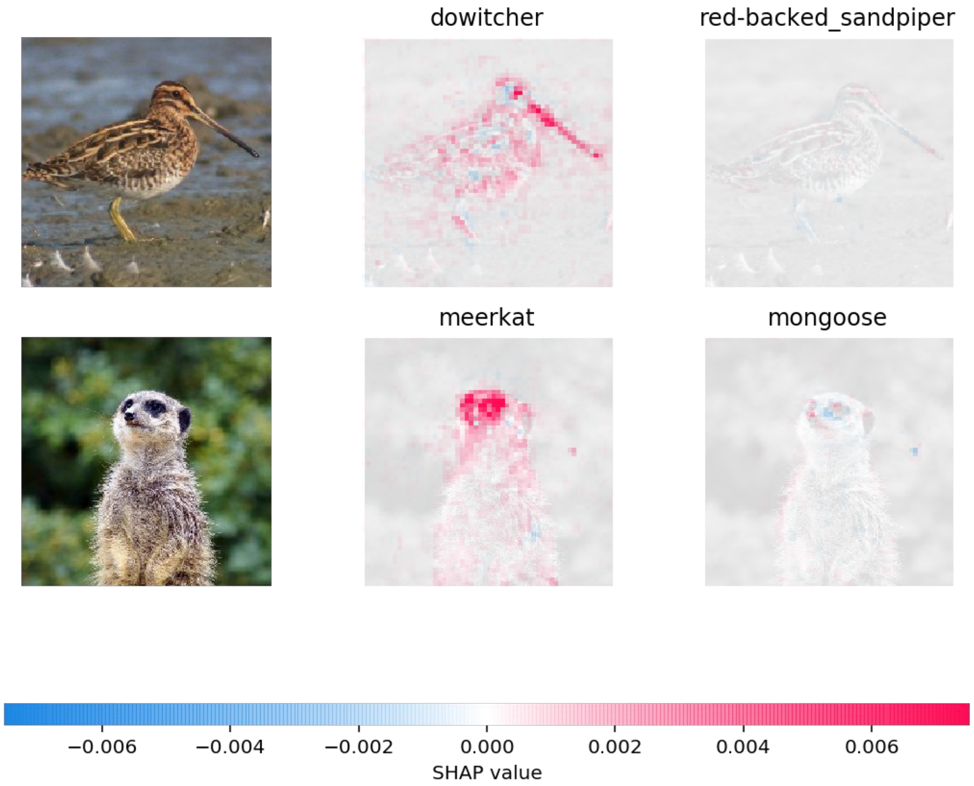
SHAP example

1. CRIM - per capita crime rate by town
2. ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS - proportion of non-retail business acres per town.
4. CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
5. NOX - nitric oxides concentration (parts per 10 million)
6. RM - average number of rooms per dwelling
7. AGE - proportion of owner-occupied units built prior to 1940
8. DIS - weighted distances to five Boston employment centres
9. RAD - index of accessibility to radial highways
10. TAX - full-value property-tax rate per $10,000
11. PTRATIO - pupil-teacher ratio by town
12. B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT - % lower status of the population
14. MEDV - Median value of owner-occupied homes in $1000's



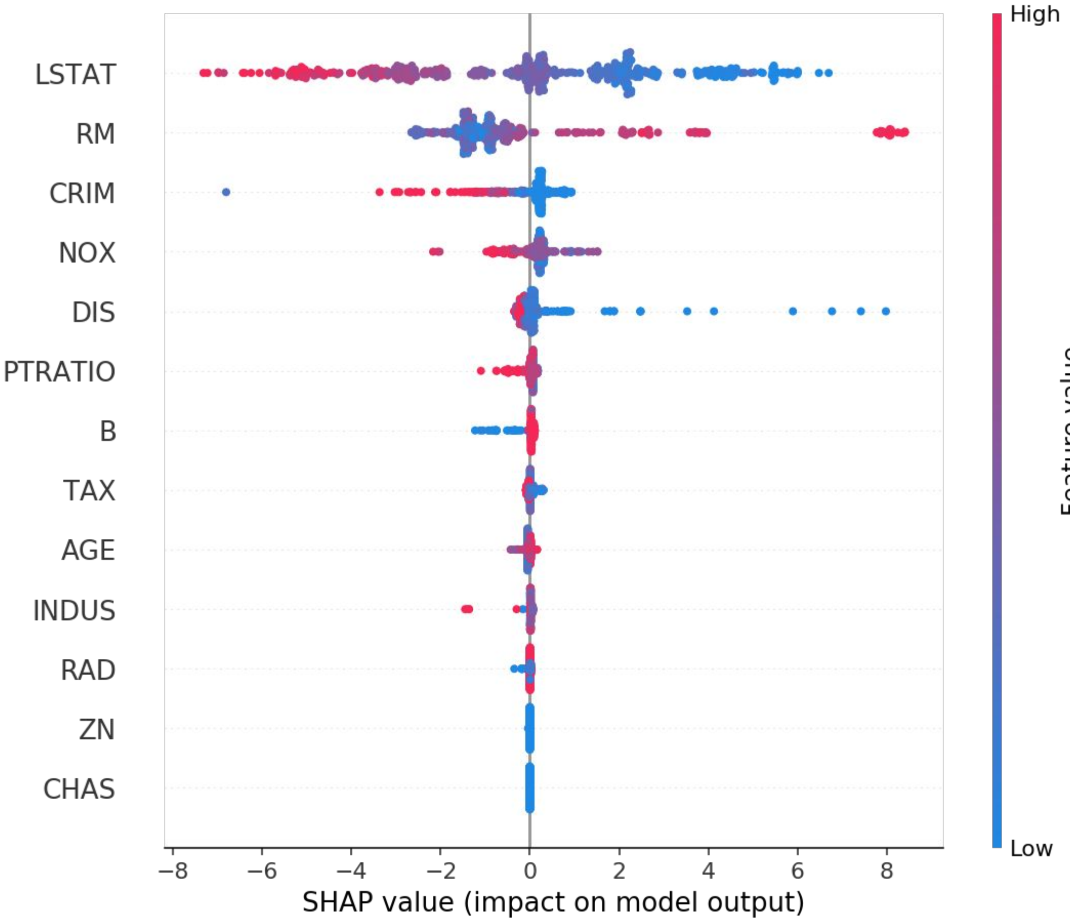
https://github.com/slundberg/shap

SHAP example



SHAP example

1. CRIM - per capita crime rate by town
2. ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS - proportion of non-retail business acres per town.
4. CHAS - Charles River dummy variable (1 if tract bounds river; 0 othe
5. NOX - nitric oxides concentration (parts per 10 million)
6. RM - average number of rooms per dwelling
7. AGE - proportion of owner-occupied units built prior to 1940
8. DIS - weighted distances to five Boston employment centres
9. RAD - index of accessibility to radial highways
10. TAX - full-value property-tax rate per $10,000
11. PTRATIO - pupil-teacher ratio by town
12. B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT - % lower status of the population
14. MEDV - Median value of owner-occupied homes in $1000's



Notes on additive attribution

If the features are correlated, it's hard to divide the attribution

Feature_1 = height

Feature_2 = height*2

Feature_3 = food

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i,$$

Weight = feature_1*1 + feature_2*0.5 + Feature_3*0.8

Weight = feature_1*0.5 + feature_2*0.75 + Feature_3*0.8

Notes on attributions

It does not necessary tell how to improve. SHAP is an approximation of the model that is accurate at the point of prediction.

If the explainer says the sales is low because of weak marketing

Does not mean increasing marketing will improve sales.

Agenda

Linear regression

Decision Trees

- Categorical features

Calibration and confidence

Explainability

LAB

~~Linear regression~~

Decision Trees - XGBoost

Categorical features - Target Encoding

Explainability - XGBoost (feature importance - model level)

- SHAP (output level)

Calibration and confidence - Ensemble