Supervised learning – Decision Trees (1)

Parcours Progis

Etudes, Medias, communication, Marketing

Bahareh Afshinpour

06.11.2025

Université Grenoble Alpes

Sciences Po Grenoble

References

- https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/
- https://www.youtube.com/watch?v=pR-Of1ua6Dc

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Recall on TP3

```
num=data.select_dtypes(include='number')
num.head()
```

	Customer Lifetime Value	Income	Monthly Premium Auto	Months Since Last Claim	Months Since Policy Inception	Number of Open Complaints	Number of Policies	Total Claim Amount
0	2763.519279	56274	69	32	5	0	1	384.811147
1	6979.535903	0	94	13	42	0	8	1131.464935
2	12887.431650	48767	108	18	38	0	2	566.472247
3	7645.861827	0	106	18	65	0	7	529.881344
4	2813.692575	43836	73	12	44	0	1	138.130879

y

X

```
num_filter=num[num['Number of Policies']==1]
y = num_filter['Customer Lifetime Value']
X = num_filter.drop(columns=['Number of Open Complaints',
                             'Customer Lifetime Value',
                             'Months Since Last Claim',
                             'Total Claim Amount',
                             'Months Since Policy Inception',
                             'Income' ])
X.head()
```

	Monthly Premium Auto	Number of Policies
0	69	1
4	73	1
16	67	1
17	101	1
20	74	1

Only we consider two features

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("shape of X_train ",X_train.shape)
print("shape of X_test ",X_test.shape)
```

```
shape of X_train  (2600, 2)
shape of X_test  (651, 2)
```

```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit(X_train, y_train)
print (reg.intercept_)
#linear_predict_all=reg.predict(X)
```

```
4.398932507227073
```

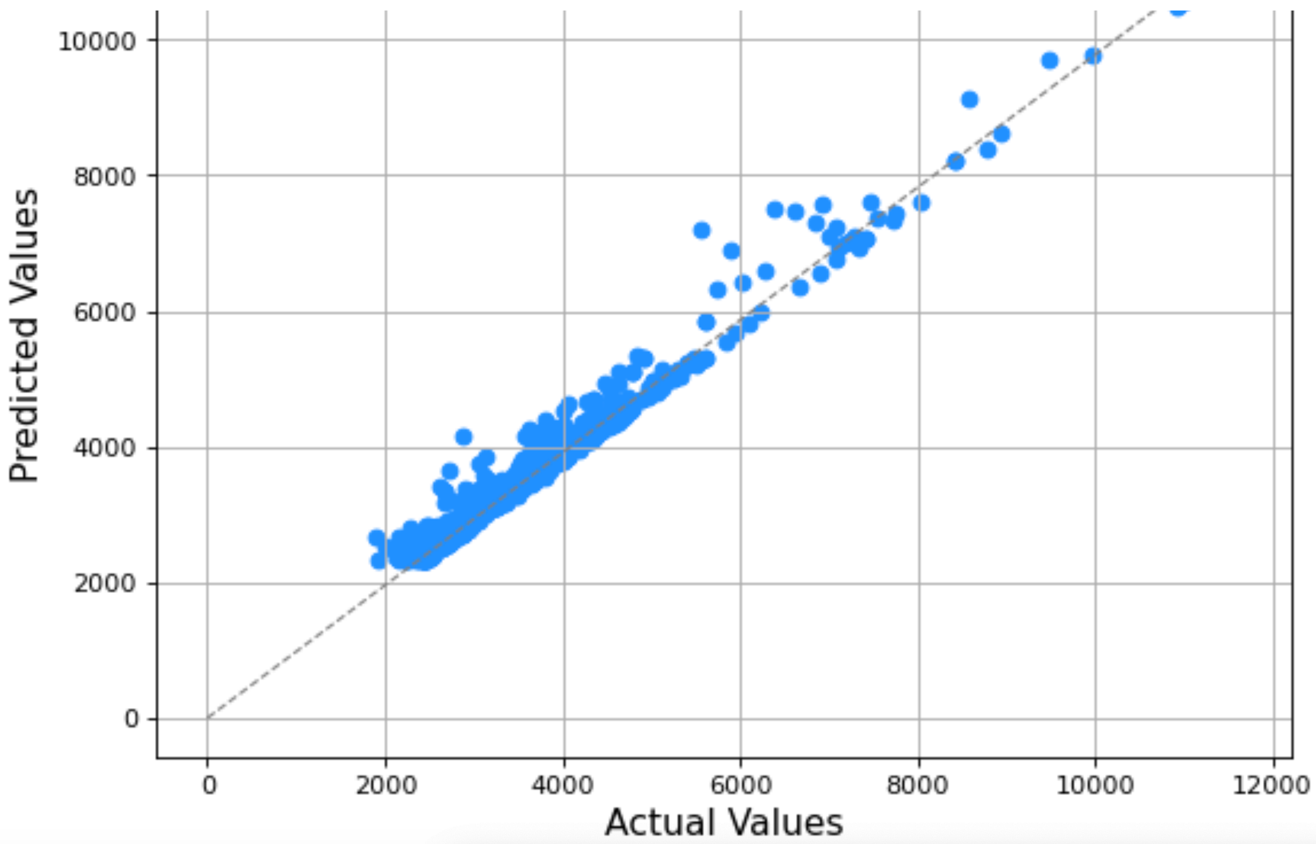
```
test_p = reg.predict(X_test)
train_p = reg.predict(X_train)
from sklearn.metrics import r2_score
print('R-Squared for Test set: %0.2f' % r2_score(y_true=y_test, y_pred=test_p))
```

```
R-Squared for Test set: 0.97
```

```
plt.figure(figsize=(8, 6), dpi=80)
plt.scatter(y_test, test_p, color='dodgerblue')
plt.plot([0, max(y_test)], [0, max(test_p)], color='gray', lw=1, linestyle='--')

plt.xlabel('Actual Values', fontsize=14)
plt.ylabel('Predicted Values', fontsize=14)
plt.title('Actual vs. Predicted for Test Set', fontsize=16)
plt.grid()
```

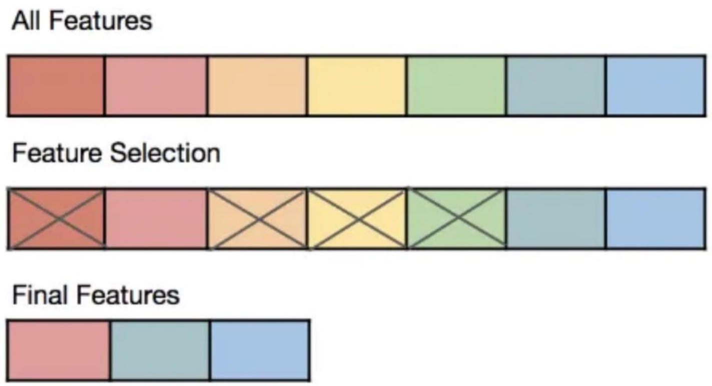
You coded it last session – TP3


10000
8000
6000
4000
2000
0
Predicted Values
0
2000
4000
6000
8000
10000
12000
Actual Values

Feature selection technique

- Feature selection is the process of reducing the number of input variables when developing a predictive

- **Why it matters:**
  - Improves accuracy
  - Decreases training time (Faster Training and In
  - Makes models easier to understand
  - ....

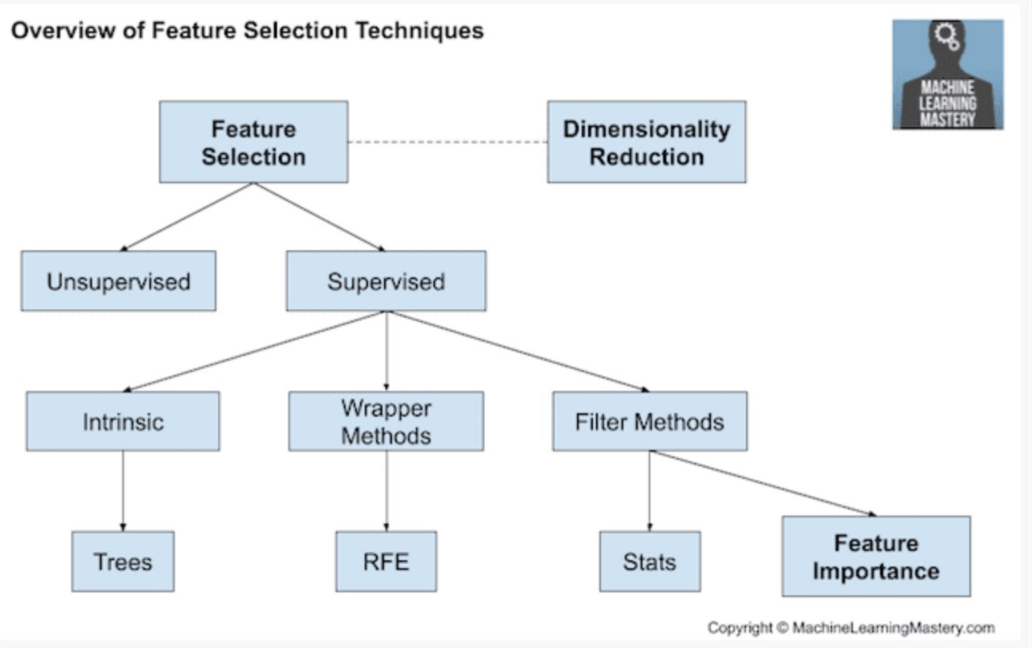


https://medium.com/@nirajan_DataAnalyst/understanding-feature-selection-techniques-in-machine-learning-02e2642ef63e

How to Choose a Feature Selection Method For Machine Learning

- **Types of Feature Selection Techniques**
  - **Wrapper**:
    - Search for well-performing subsets of features. RFE
  - **Filter**:
    - Select subsets of features based on their relationship with the target. Statistical Methods, Feature Importance Methods
  - **Intrinsic**:
    - Algorithms that perform automatic feature selection during training. Decision Trees
  - **Hybrid Methods**
    - ✓ combine aspects of filter, wrapper, and embedded methods.

How to Choose a Feature Selection Method For Machine Learning



Overview of Feature Selection Techniques

https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/

The scikit-learn library provides an implementation of most of the useful statistical measures.

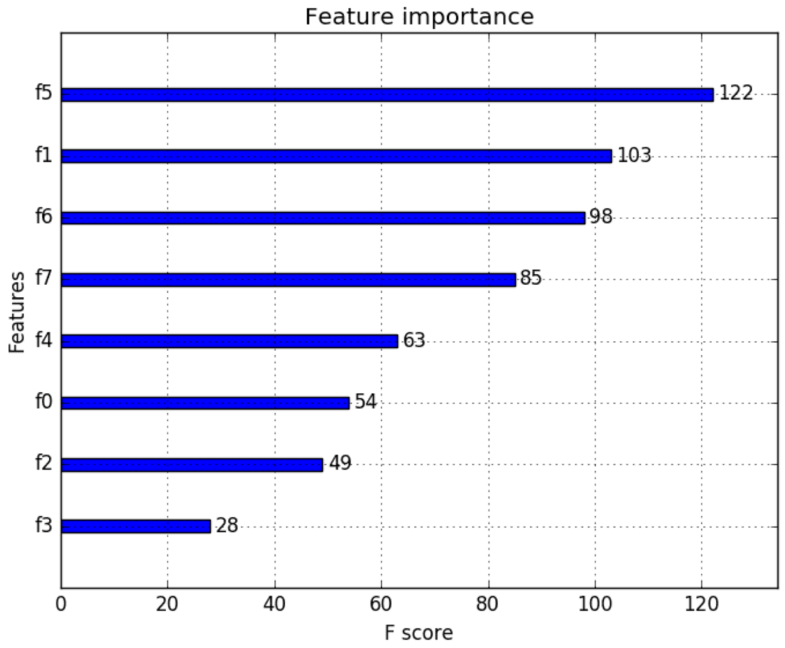
For example:

- Pearson's Correlation Coefficient
- ANOVA
- Chi-Squared

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Feature Importance

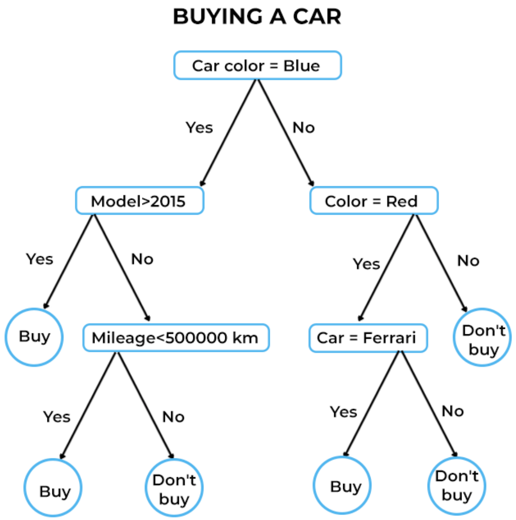
- Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction.



PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Decision Trees

- Desision tree are easy to learn as they are powerful
- A fundamental supervised learning algorithm
- How decision tree learn from data
  - They recursively splits data into the largest, **purest** groups(all examples have the same label)



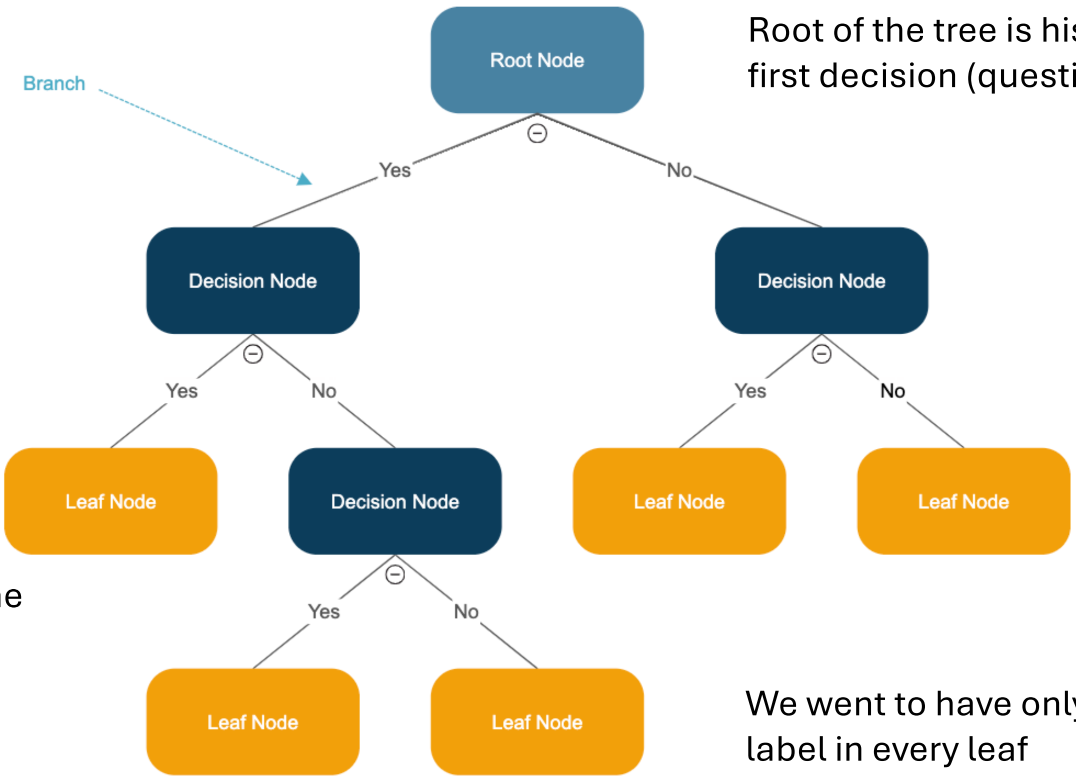
We ask a question, then based on the answer to the question , we move down to the tree.

https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-decision-tree/

PROGIS

ETUDES, MEDIAS, COMMUNICATION, MARKETING

Desision Tree



Root of the tree is his very first decision (question)

Leaf nodes are where the predictions are made.

We went to have only one label in every leaf

Desision Tree

- A leaf node is 100% pure :
  - every row of data that flows down the tree and ends up in a particular node have a same label
- decision tree models learn the data by partitioning the data points based on certain criteria.

- Using either of these measures, decision trees can grow until all of the nodes are pure or until the stopping criteria are met.

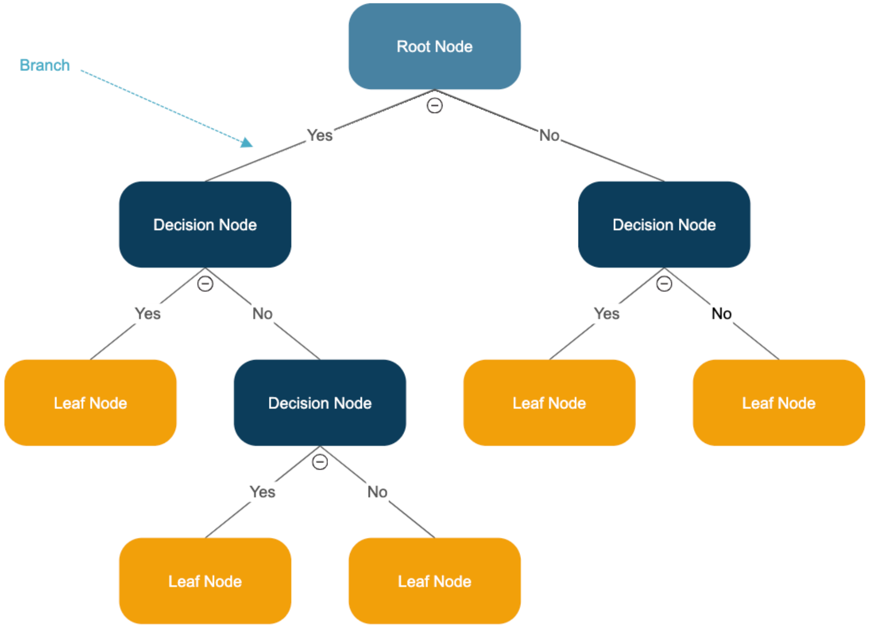
Desision Tree Regression

- The goal of using a decision tree as you create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data commonly referred to as the training data.

- we start from the root of the tree. We compare the values of the root attribute with the records attribute. On the basis of this comparison, we follow the branch corresponding to that value and jump to the next node.

Desision Tree- Parent and child node

- A node which is divided into sub nodes is called a **parent** node of the sub. Nodes whereas the sub nodes are the **child** node of that particular parent.

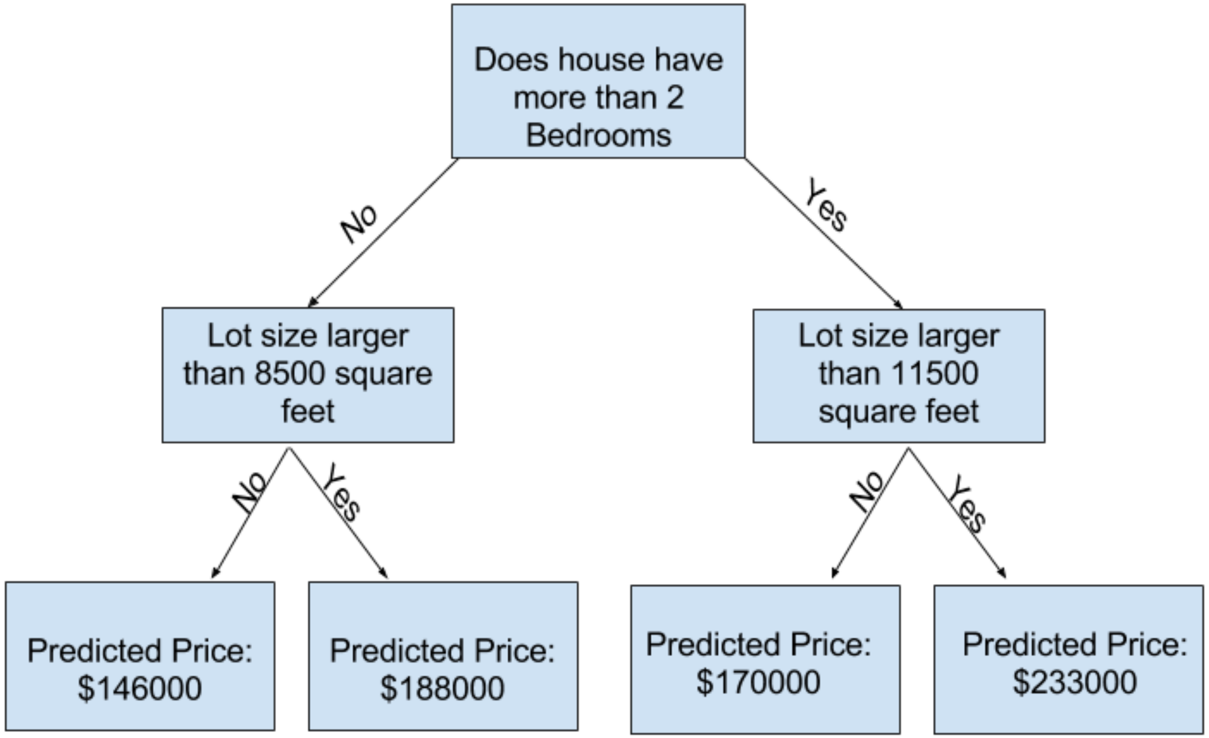


Desision Tree Regression

- Decision trees as the name suggests learn from the data by growing a tree.
- The main difference between the logistic regression and decision tree model is the fact that logistic regression algorithms search for a single best linear boundary in the feature set, whereas the decision tree algorithm **partitions** the data to **find the subgroups** of data that have **high likelihood** of an event occurring.
- Desision tree models perform better for **non-linear** datasets.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Example of Desision Tree



Working of the algorithm

- **Building the Tree:**
  - Imagine you have a dataset with various attributes (features) and their corresponding target values. The algorithm begins by creating a root node that represents the entire dataset.
- **Splitting the Data:**
  - The algorithm analyzes each feature to determine the best way to divide the data into distinct groups based on their target values. It does this by setting specific conditions or thresholds on the feature values.
- **Recursive Splitting:**
  - The algorithm then repeats this process for each child node, recursively splitting the data further based on the best features and conditions.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

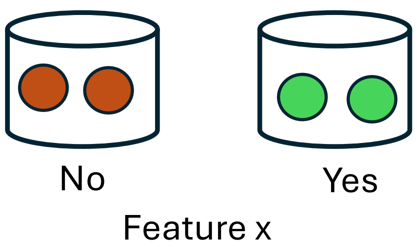
Example of Desision Tree- visual representation

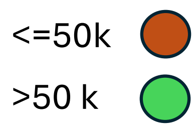
Target variable

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k

Which of these columns(features) best splits these labels into the largest purest buckets?

We have two rows less that 50k and two more than 50k



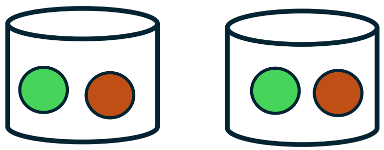


Example of Desision Tree-visual representation

Target variable

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50k
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k

First we look at age feature:



No Yes

Age>=50

We have impurity.
Age creates a 50/50 split.
We are completely uncertain of its effect on salary.
This feature does not really help us in making a prediction

<=50k

>50 k

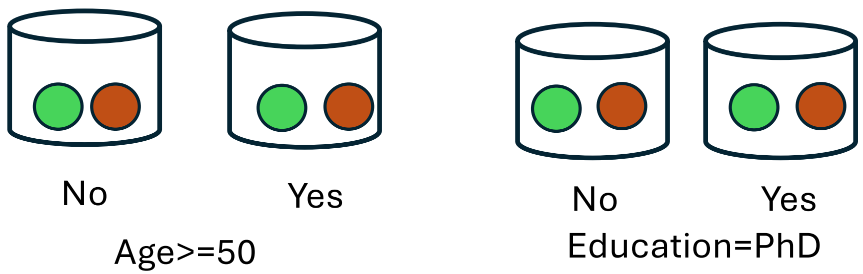
Example of Desision Tree-visual representation

Target variable

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50k
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k

move on the education columns

50/50. Education won ‘t help us make prediction.



<=50k

>50 k

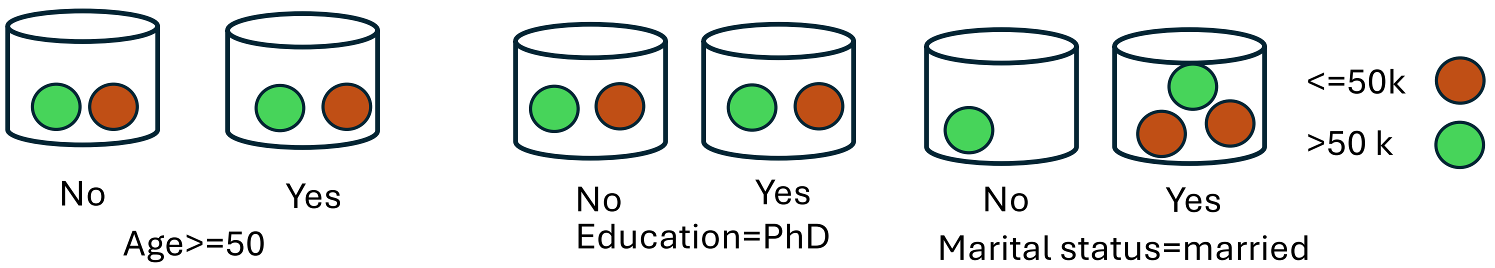
Example of Desision Tree-visual representation

Target variable

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50k
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k

move on the Marital status

At least one is pure. It does not offer a clean split either



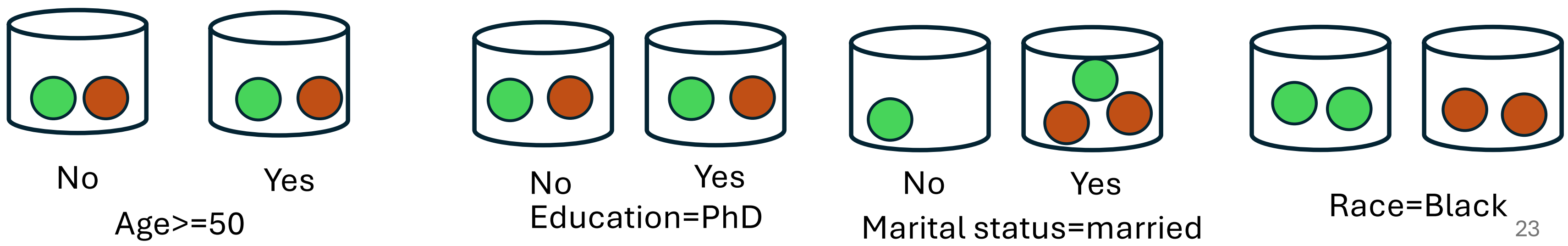
Example of Desision Tree-visual representation

Target variable

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50k
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k

move on Race

100% pure



Desision Tree

- We can continue to check other features.
- In this example you can see the feature hours per week is also 100% pure.
- But between race and hours per week, which one is the best one.
- Decision trees is known as a **greedy algorithm**. And it picks the very **first feature** that it finds that is the best.
- So, in this example, at the top of the tree should use race=black as a root node. (first decision in the tree)

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

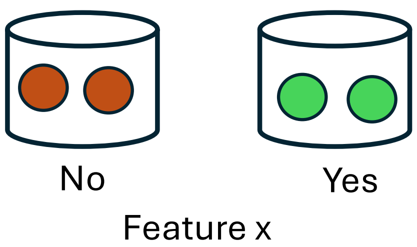
Example of Desision Tree-visual representation

Target variable

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k

Which of these columns(features) best splits these labels into the largest purest buckets?

We have two rows less that 50k and two more than 50k



<=50k

>50 k

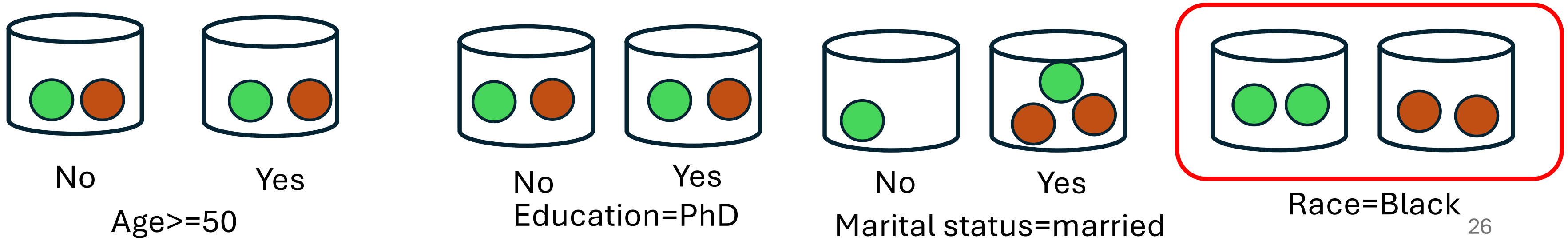
PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Example of Desision Tree-visual representation

Target variable

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50k
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k

Race is a best one 100% pure



LISTS IN PYTHON

Index	**0**	**1**	**2**	**3**	**4**
List Data	David	4.12	6	**[3,9]**	657

[David,4.12,6,[3,9],657]

- Lists are a popular data structure in Python.
- Each box has a numerical reference called **an index** that is used to refer to the individual data item.
- A list is represented with square brackets.
- Lists can contain strings, floats, integers. Also, we can nest other lists.
- Note that in Python the first element of the list shown here has an index of **zero**.

LIST OPERATIONS

- Lists are mutable; can be changed in-place
- Lists are dynamic; size may be changed

```
>>> r = [1, 2.0, 3, 5]
>>> r[3] = 'word'                # replace an item by index
>>> r
[1, 2.0, 3, 'word']
```

```
>>> len(r)                       # length of list; number of items
4
```

```
>>> 6 in r                       # membership test
True
```

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

LIST METHODS

- Lists have a set of built-in methods
- Some methods change the list in-place

```
>>> r = [1, 2.0, 3, 5]
>>> r.append('thing')              # add a single item to the end
>>> r
[1, 2.0, 3, 5, 'thing']
>>> r.append(['another', 'list'])  # list treated as a single item
>>> r
[1, 2.0, 3, 5, 'thing', ['another', 'list']]
```

```
>>> r = [2, 5, -1, 0, 20]
>>> r.sort()
>>> r
[-1, 0, 2, 5, 20]
```

```
>>> s = 'a few words'
>>> w = s.split()               # splits at white-space (blank, newline)
>>> w
['a', 'few', 'words']
```

End