

A futuristic AI robot head is shown in profile, facing left. It has a metallic, greyish-blue finish with a large, circular, glowing blue eye. The robot is positioned in front of a complex digital interface. The interface features various financial charts and data visualizations, including a bar chart in the top left, a line graph in the top right, a world map in the center, and several circular gauges and smaller charts. The overall color scheme is a cool blue and grey, with glowing blue highlights from the robot's eye and the interface elements.

# MGMT 675

# AI-ASSISTED FINANCIAL ANALYSIS



**RICE | BUSINESS**  
Jones Graduate School of Business



# TREES AND FORESTS

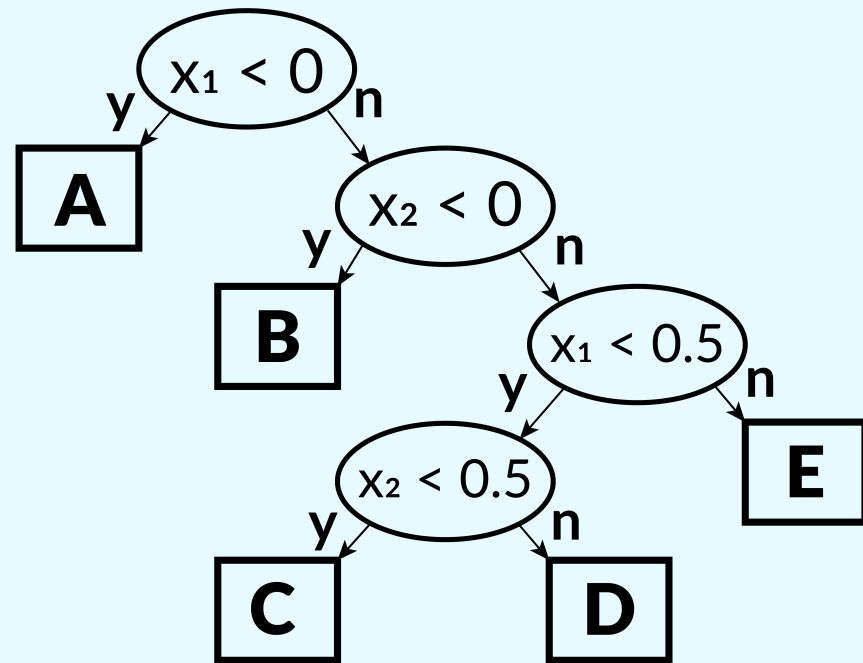
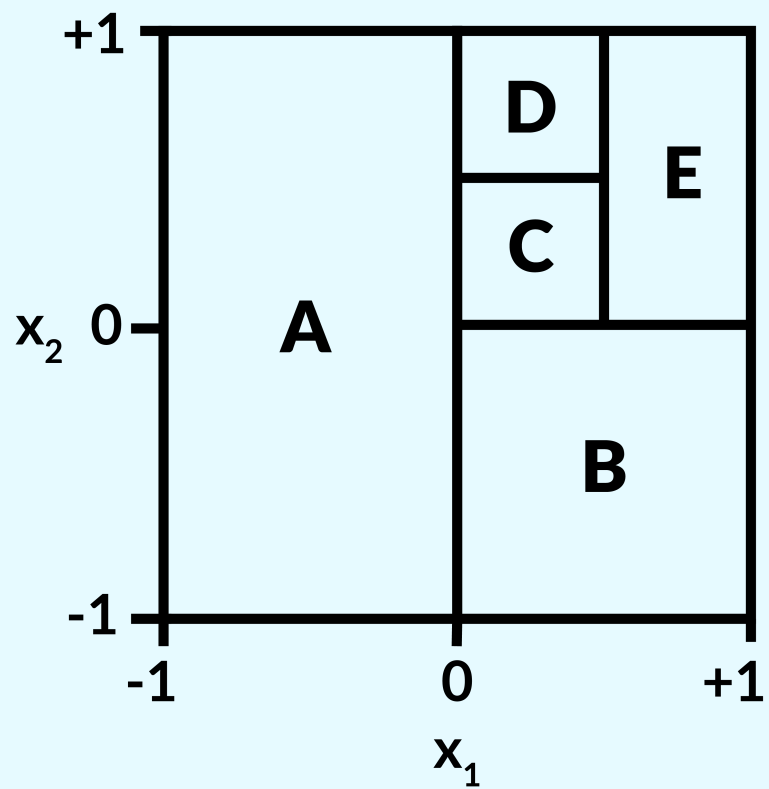
The background of the slide is a complex digital interface with a blue and white color scheme. In the center, a white humanoid robot with glowing blue eyes and joints is shown in profile, interacting with a floating digital globe. To the left, a large screen displays a world map with a jagged line graph below it. To the right, another screen shows a globe surrounded by various charts, graphs, and data tables. The overall aesthetic is high-tech and futuristic.

# OUTLINE

- Decision tree
- Random forest and gradient boosting
- Shapley values
- House price application
  - Missing values
  - Dummy variables
  - Scaling

# DECISION TREE

- Split dataset into subsets. Within each subset,  $\hat{y}$  = mean of subset. Calculate MSE.
- Split each subset into further subsets and continue.
- Use means of subsets to estimate. Choose splits to minimize MSE.



# EXAMPLE

- Ask Julius to read irrelevant\_features.xlsx.
- Ask Julius to fit a decision tree regressor with y1 as the target using all of the data as training data. Ask Julius to plot the tree.

# **RANDOM FOREST AND GRADIENT BOOSTING**



# RANDOM FOREST

- Generate random datasets of the same size as the original.
- Create the random datasets by randomly drawing rows from the original with replacement.
- Fit a decision tree to each random dataset.
- The prediction for any observation is the average of the predictions of the various trees.

- Randomization helps to avoid overfitting.
- Also control overfitting through:
  - `max_depth` = maximum number of times to split in each tree
  - `max_features` = number of features to look at when deciding how to split (a subset of features of that size is randomly chosen for each split)

# GRADIENT BOOSTING

- Fit a decision tree.
- Look at its errors. Fit a new decision tree to predict the errors.
- New prediction is original plus a fraction of the prediction of original's error (fraction = learning rate).
- Look at the errors of the new predictions. Fit a new decision to predict these errors.
- Continue ...

# EXAMPLES

Ask Julius to train and test

- a random forest regressor
- a gradient boosting regressor

to predict  $y_1$  in irrelevant\_features.xlsx.

Ask Julius to use GridSearchCV to

- find the best max\_depth for the random forest regressor in (5, 10, 15, 20, 25)
- find the best learning rate for the gradient boosting regressor in (0.001, 0.005, 0.01, 0.05, 0.1, 0.2)
- find the best max\_depth and best learning rate for the gradient boosting regressor in some sets

# SHAPLEY VALUES

- The Shapley value for a feature at an observation is a measure of how much that feature contributed to the prediction at that observation.
- A summary of Shapley values is a bar chart showing the mean absolute contribution of each feature (mean across observations).
- A Shapley scatter plot for a feature plots all of the observations with the feature's value on the x axis and the feature's contribution to the prediction on the y axis.



- Ask Julius to create a summary plot of the Shapley values for the random forest regressor with the best max\_depth.
- Ask Julius to create a scatter plot of the Shapley values for the x1 feature.
- Ask Julius to create a scatter plot of the Shapley values for another feature.

# VALUING HOUSES

# DATA

- Download house\_price.xlsx from the **course website**
- Upload the file to Julius.
- Ask Julius to read the data and describe it.
- Tell Julius that SalePrice is the target and the other columns are features.

# NEW TOPICS

- Missing values. Possible solutions:
  - Fill in missing values
  - Drop columns with missing values
  - Drop rows with missing values
- Categorical variables. Convert to dummies.
- Scaling features. It is important for some models that features be on the same scale.

# MISSING VALUES

Tell Julius to fill in missing values

- for categorical features with “None”
- for numeric features with 0.

# DUMMY VARIABLES

## Categorical

	Feature
Row1	Hi
Row2	Lo
Row3	Med
Row4	Med
Row5	Lo

## Dummies

	Lo	Med	Hi
Row1	0	0	1
Row2	1	0	0
Row3	0	1	0
Row4	0	1	0
Row5	1	0	0



# PIPELINE

Ask Julius to create a pipeline that

- transforms the qualitative features to dummy variables
- applies Standard Scaler to the numeric features
- applies a random forest regressor

# TRAIN AND TEST

Ask Julius to train and test the pipeline.

# FURTHER WORK

- Apply GridSearchCV to the pipeline to find best hyperparameters
- Replace random forest regressor with other models:
  - lasso regressor
  - ridge regressor
  - gradient boosting regressor