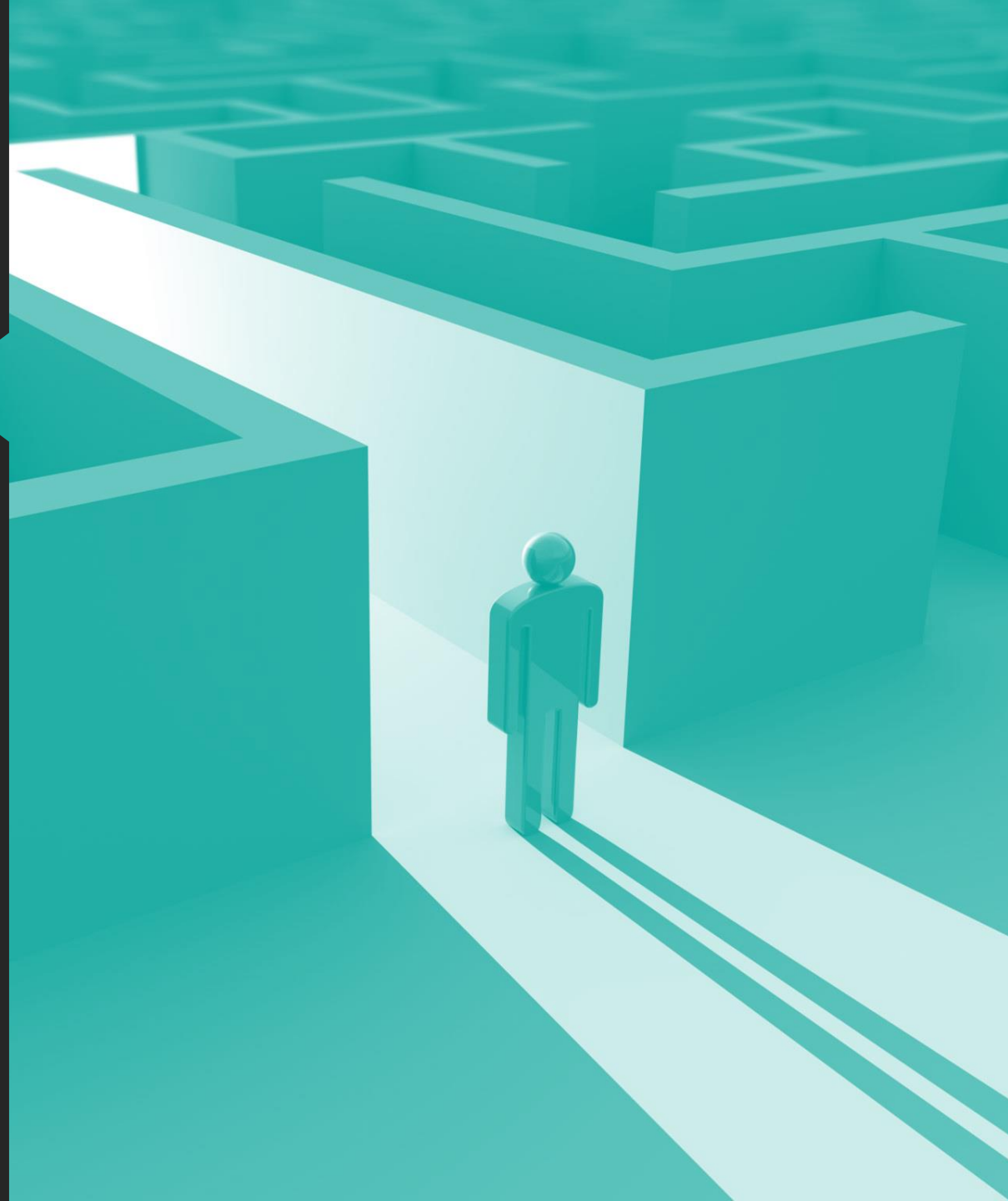
Indoor Locationing Predictions using WiFi Fingerprinting

IOT Analytics:
For Internal Use Only

Jennifer Brosnahan, MPH

Background

- Our client is developing a system to help people navigate a complex, unfamiliar interior space on a college campus.
- They would like us to investigate the feasibility of using wifi-fingerprinting to determine a person's indoor location.
- If a model meets or exceeds minimum specifications, it will be incorporated into a smart phone app for indoor locationing on a college campus.



Business Objectives (Goals)

1

Build multiple different models to predict a person's location in indoor college campus spaces using Wifi signals

2

Answer the business question, "Can a model be built that predicts indoor location on a college campus that meets the Client's minimum specifications?"

3

If a model can be built meeting minimum specifications, deploy algorithm into smart phone indoor locationing app.

Client Minimum Specifications

- Indoor location must be as precise as predicting within 10-15 feet of the indoor room, also defined as 'SpaceID' within source data. Relative position, or whether individual is outside or inside of room, is unnecessary.
- Performance metrics ideal for deployment:
  - Accuracy scores on test data reaches 80% or higher
  - Precision (accuracy of minority class) on test data reaches 80% or higher
  - Recall (coverage of minority class) on test data is 80% or higher
  - F1 Score for multi-class problem achieves 80% or higher

Data Description

- Data was collected by 20 individuals using mobile phone devices on a college campus in Velencia, Spain
- The data source is the UJIIndoorLoc WLAN database
  - 19937 observations of 529 variables
  - 520 of the variables (98% of dataset) are homogenous wifi-access points providing numeric values representing signal strength
  - 9 remaining variables are Longitude, Latitude, Floor, BuildingID, SpaceID, Relative Position, UserID, PhoneID, and Time
  - Contains no missing values

Data Management Methodologies

Feature Engineering

- A unique dependent variable ('location') was engineered in order to pinpoint room location on campus by concatenating 3 features from out-of-box (OOB) dataset:
  - 'BuildingID' (3 buildings, values = 0, 1, 2)
  - 'Floor' (5 floors, values = 0, 1, 2, 3, 4)
  - 'Space ID' (125 total space IDs)
  - Results in 735 total unique 'location' classes

Data Management Methodologies

Feature Selection

- As this problem requires predicting location using Wifi Access Points (WAPs), all non-WAP variables were removed
- Removing zero variance variables is utilized as a strategy to help reduce dimensionality from datasets during model training

Four primary datasets were developed from raw source data and utilized in model training process (outlined on following slide)

- Out-of-box (OOB) dataset (contains all campus locations to predict)
- Because OOB dataset is extremely large, smaller datasets filtered by building were created for model training as a strategy to see if models trained by individual building will have enhanced performance over a model trained on all campus location data.

Dataset Descriptions

Dataset 1: Out-of-box (OOB)

- 19,937 observations and 521 variables
- 520 WAP variables retained as independent variables, 'location' as dependent variable

Dataset 2: Building 0

- 5249 row observations and 201 variables after zero variance variable removed
- 200 WAP independent variables, 'location' as dependent variable

Dataset 3: Building 1

- 5196 row observations and 208 variables retained after zero variance variable removed
- 207 WAP independent variables, 'location' as dependent variable

Dataset 4: Building 2

- 9492 row observations and 204 variables retained after zero variance variable removed
- 203 WAP independent variables, 'location' as dependent variable

Algorithms

Four classification algorithms were chosen as follows:

- Decision Tree – automatically prunes ineffective nodes and branches, useful in high dimensional datasets
- Random Forest – handles large and high dimensional datasets well and performs automatic dimensionality reduction
- KNN – trains much quicker than tree-based models and new data can be added easily
- Support Vector Machines – can be effective in high dimensional spaces and when number of observations is greater than the number of features to train

Resampling Techniques

Resampling methods

- Each dataset was split into training and testing data at 75% and 25%, respectively.
- K-fold cross validation was utilized to minimize likelihood of overfitting model. Data was split into 3 k-fold subsets.

Cross Validation Results

- 3 K-fold subsets used to minimize likelihood of overfitting
- Random Forest consistently performed best in cross validation comparisons, followed by SVC.
- Both deemed worthy of further investigation through hyperparameter tuning and post-resampling.

Algorithm	oob2	Building 0	Building 1	Building 2
Decision Tree	.45323	.43574	.61586	.39475
Random Forest	**.67418**	**.67903**	**.78868**	**.61410**
Support Vector RBF	**.51297**	**.45574**	**.62721**	**.50390**
K Nearest Neighbor	.47455	.43422	.59199	.43605

Post-resample Performance Comparison of All Models

Random Forest algorithm is top performer across all post-resampling dataset results

	RF oob2	RF Building 0	RF Building 1	RF Building 2	SVC oob2	SVC Building 0	SVC Building 1	SVC Building 2
Accuracy	0.81384	0.76770	0.87914	0.82385	0.69067	0.61538	0.74288	0.72524
Precision (weighted)	0.85869	0.81528	0.89504	0.89055	0.74650	0.67022	0.76627	0.80129
Recall (weighted)	0.81926	0.76771	0.88527	0.82455	0.69527	0.62106	0.74748	0.72585
F1 (weighted)	0.81616	0.76631	0.87668	0.83257	0.69356	0.61416	0.73967	0.73484

Recall comparison of Random Forest

- All Random Forest algorithms meet the minimum specification for 80% accuracy except Building 0, which is slightly less.
- Whether to choose the algorithm trained on the full dataset (oob2) or algorithms trained by individual buildings is hard to decide based on average model metrics.
- Recall helps gauge how many spaces the models correctly classify (True Positives) out of all Actual Positives within each class. There is a high cost associated with a False Negative (incorrect room prediction) when considered for a smart phone app.

Recommendation: Compare recall metrics by quartile (absolute count for location) for all models in order to make a more informed decision on model recommendation.

Recall comparison of Random Forest

- Higher recall is better
- Total locations in 75-100% recall ranges:
  - Oob2: 456
  - Buildings 0, 1, 2: 483
- 27 more locations in high recall range for individual buildings versus full dataset

Recall quartiles	RF oob2	RF Building 0	RF Building 1	RF Building 2	Total Ind Buildings
0 – 25%	20	5	4	9	18
25 – 50%	88	40	13	36	89
50 – 75%	166	70	18	53	141
75 – 100%	456	141	125	217	483

Model Recommendation

Recommend deploying **Random Forest algorithms trained on individual buildings** for smart phone app

- All metrics for Buildings 1 and 2 well surpass 80% minimum goal and are higher than model trained on full dataset. Metrics for Building 0 are slightly below 80% goal, however, this is offset by greater accuracy on Buildings 1 and 2.
- Models trained on individual buildings predicted more overall room locations with high recall, True Positives out of Actual Positives (483 total) versus model trained on full dataset (456 total).
- Higher recall on 17 more locations to predict is important when considering for smart phone app deployment.

Alternative Solutions to Wi-Fi Fingerprinting

- Acuity Brands provides LED integrated indoor positioning light fixtures which send flickering patterns readable by phone receivers and link to indoor locationing maps. www.bytelight.com
- IndoorAtlas uses a variety of technologies for indoor locationing, including geomagnetic positioning, Wi-Fi signals, and Barometric pressure (vertical movement), all of which are read be a cell phone's sensors. www.indooratlas.com/positioning-technology
- iBeacon transmitters, or Bluetooth Low Energy (BLE) devices, project signals to nearby electronic devices which can be used to determine device's physical location. developer.apple.com/ibeacon
- Infrared (IR) systems use infrared light pulses to locate IR receiver signals installed in each room of a building. IR tag pulses emitted by receivers can be read by IR receiver device.
- Real-time fingerprinting apps rely on user collaboration to "check-in" and label fingerprints on their devices, which are then stored in a repository and pass through an algorithm that defines their current location. www.foursquare.com

Summary

- Random Forest algorithms overall met or exceeded client minimum specification metrics for predicting WiFi locations on a college campus.
- Recall for location predictions on individual building datasets was higher than OOB dataset.
- Recommend **deploying Random Forest algorithms by individual buildings** for Indoor Locationing smart phone app