



KNN Regression and Generalization

Applied Machine Learning
Derek Hoiem

Announcements

- HW0: practice & check knowledge of math as it relates to ML (optional, worth 5 XP)
- HW1: due Sept 14, will go over it in next class
- Planning to release two design problems, likely first in late Sept or early Oct, for some final project points
- Next week: I am traveling for ECCV, recorded lectures only (can still submit quizzes any time that day for credit)

Previous Lecture Recap

- **Data** is a set of numbers that contains information. Images, audio, signals, tabular data and everything else must be represented as a vector of numbers to be used in ML.
- **Information** is the power to predict something – a lot of the challenge in ML is in transforming the data to make the desired information more obvious
- In machine learning, we have
 - Sample**: a data point, such as a feature vector and label corresponding to the input and desired output of the model
 - Dataset**: a collection of samples
 - Training set**: a dataset used to train the model
 - Validation set**: a dataset used to select which model to use or compare variants and manually set parameters
 - Test set**: a dataset used to evaluate the final model
- In a **classification** problem, the goal is to map from features to a categorical label (or “class”)
- Nearest neighbor (or **K-NN**) algorithm can perform classification by retrieving the K nearest neighbors to the query features and assigning their most common label
- We can measure **error** and **confusion matrices** to show the fraction of mistakes and what kinds of mistakes are made

Machine learning model maps from features to prediction

$$f(x) \rightarrow y$$

Features Prediction

Examples

- Classification: predict label
 - Is this a dog or a cat?
 - Is this email spam or not?
- Regression: predict value
 - What will the stock price be tomorrow?
 - What will be the high temperature tomorrow?
- Structured prediction: predict a set of related values
 - What is the pose of this person?



Key principle of machine learning

Given feature/target pairs $(X_1, y_1), \dots, (X_n, y_n)$:

if X_i is similar to X_j , then y_i is probably similar to y_j

Fundamentally, learning depends on:

1. Representation of samples
2. Similarity function



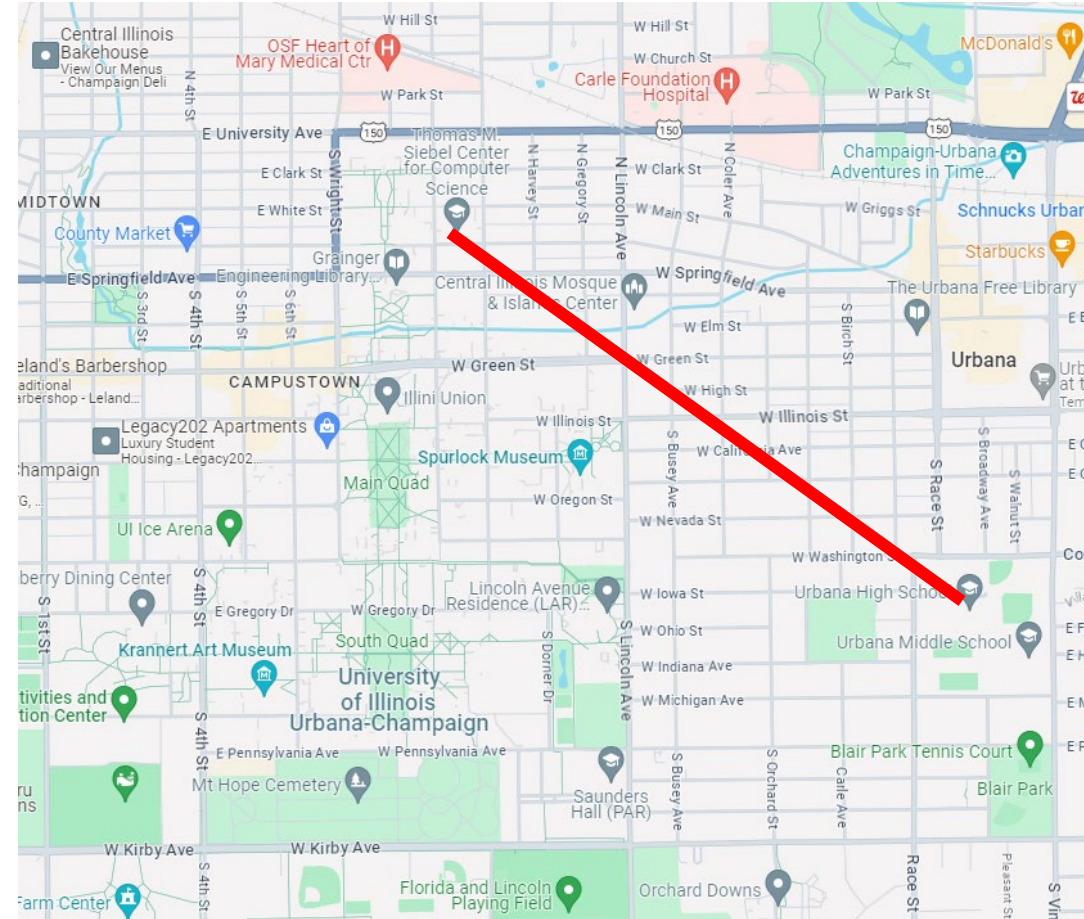
Today's lecture

- Similarity measures
- Regression
- Generalization

Common Distance/Similarity Measures

- L2: Euclidean

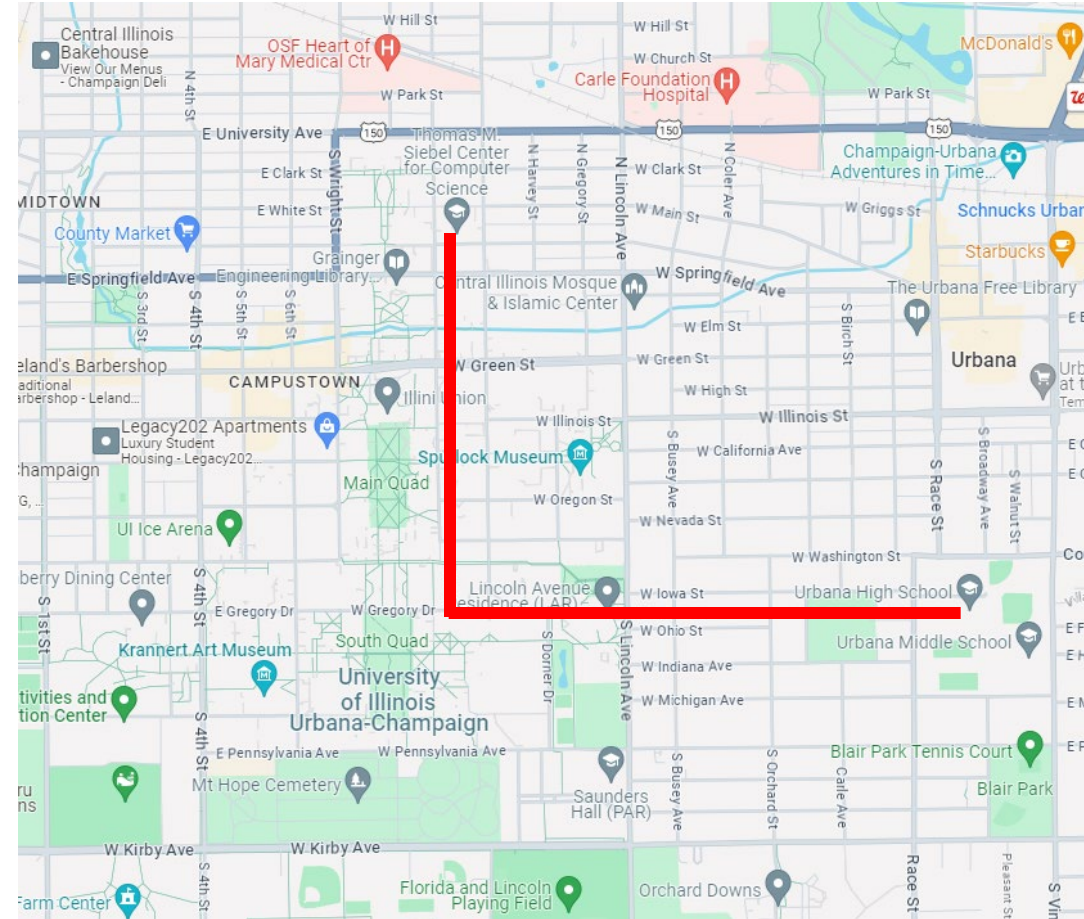
$$d_2(x, y) = \|x - y\|_2$$
$$= \sqrt{\sum_i (x_i - y_i)^2}$$



Common Distance/Similarity Measures

- L1: City-Block

$$d_1(x, y) = \|x - y\|_1 \\ = \sum_i |x_i - y_i|$$



Common Distance/Similarity Measures

- Dot product, Cosine

Dot product (or inner product)

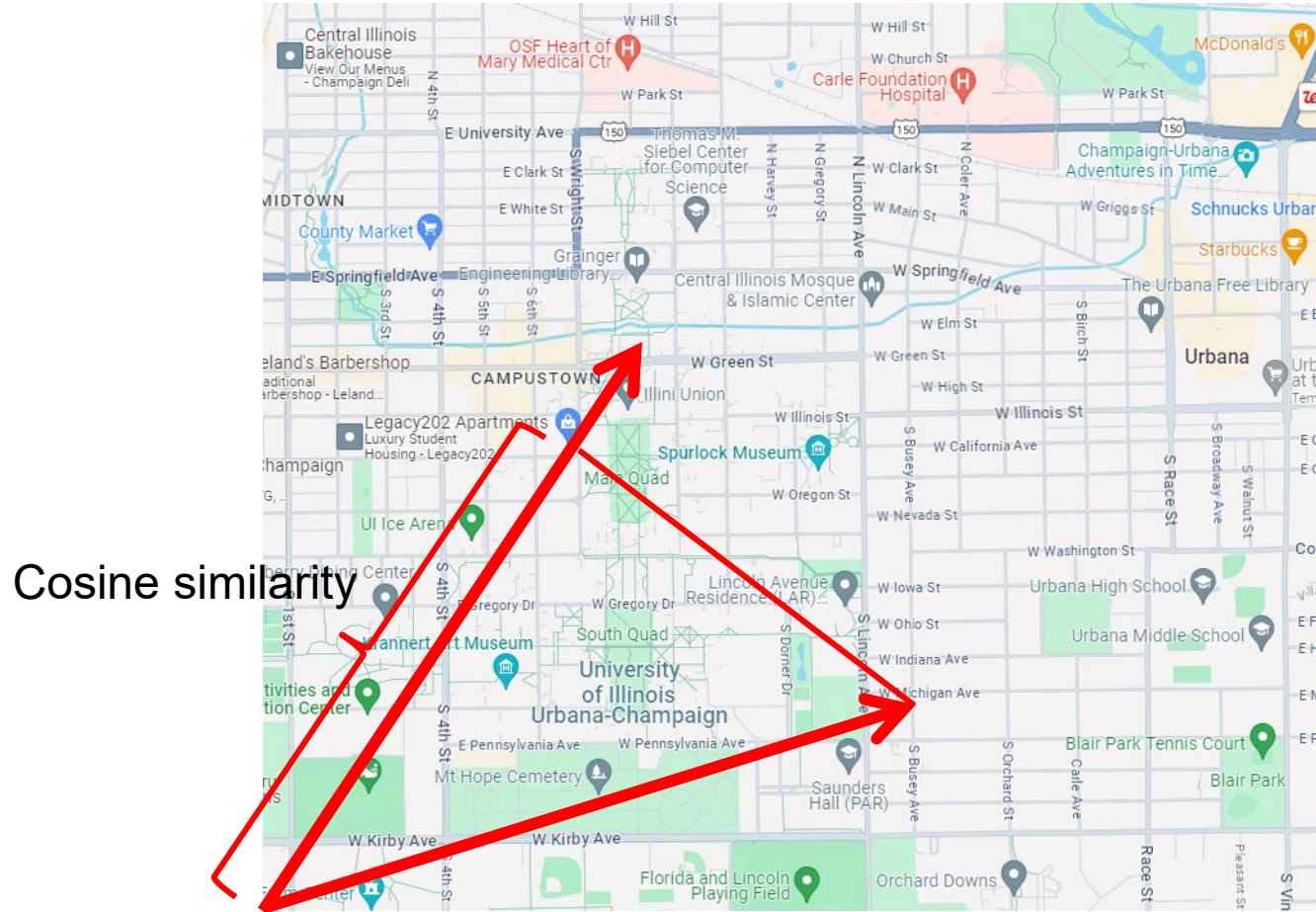
$$s_{dot}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y} = \sum_i x_i y_i$$

Cosine similarity

$$s_{cos}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}$$

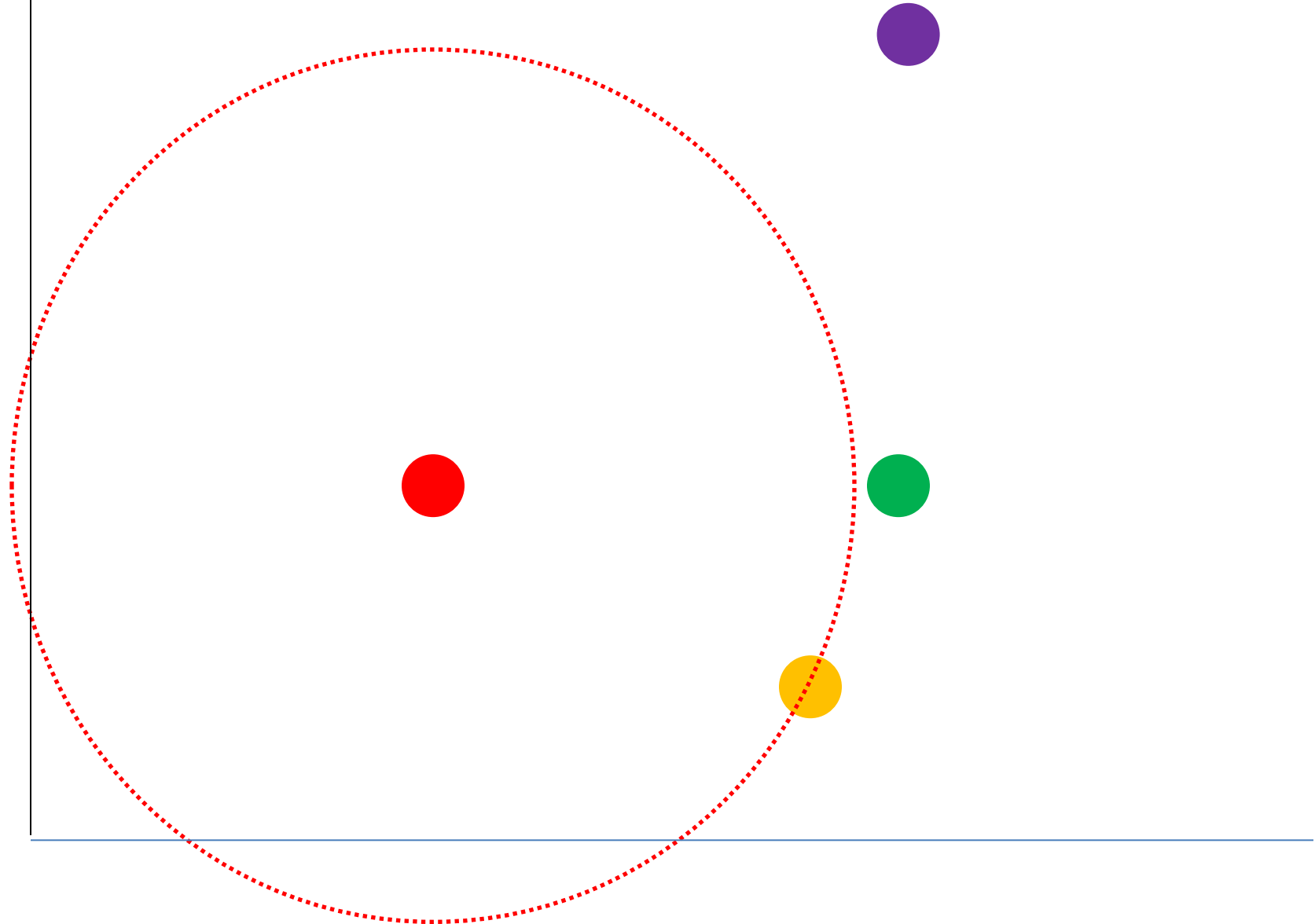
Dot product: how far does one vector go in the direction of the other vector

Cosine similarity: how similar are the two directions





Which is closest to the red circle under L1, L2, and cosine distance?



Comparing distance/similarity functions

- L2 depends much more heavily than L1 on the coordinates with the biggest differences

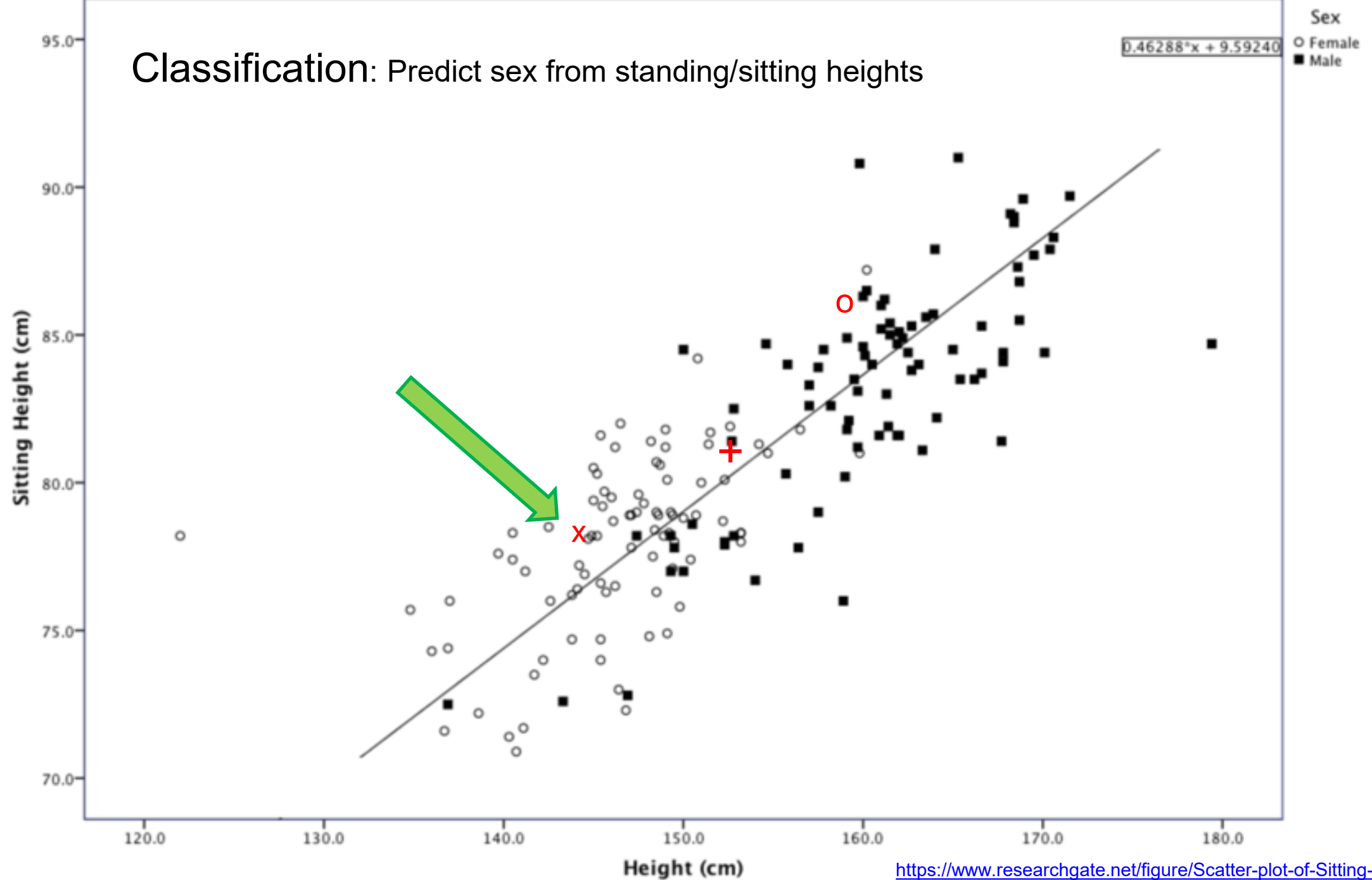
$$d_2([0 \ 100], [5 \ 0]) = 100.1$$

$$d_1([0 \ 100], [5 \ 0]) = 105$$

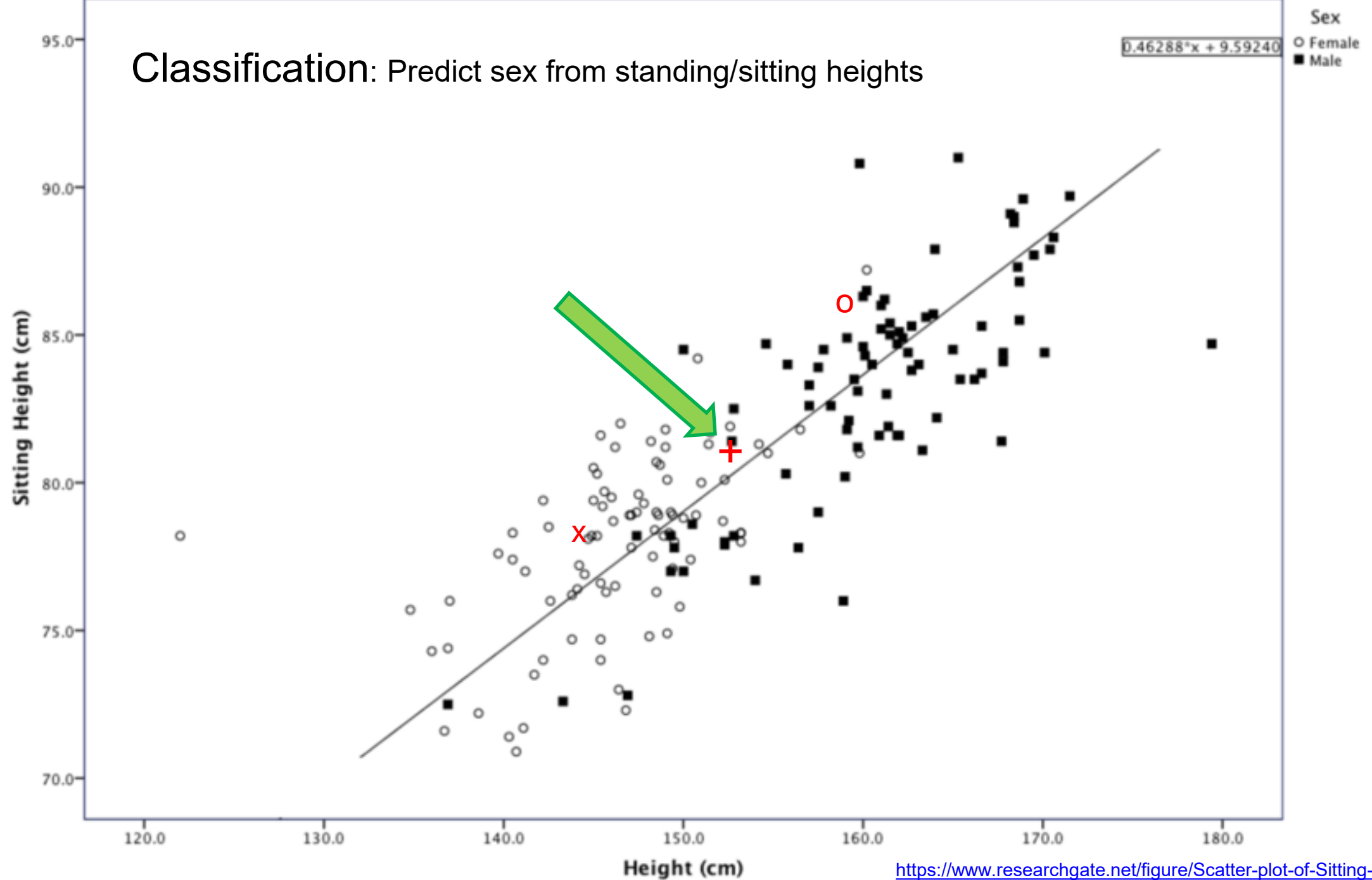
- Cosine and L2 are equivalent if the vectors are unit length

$$\|x - y\|_2^2 = \underset{1}{x^T x} - 2\underset{1}{x^T y} + \underset{1}{y^T y} = 2(1 - s_{cos}(x, y))$$

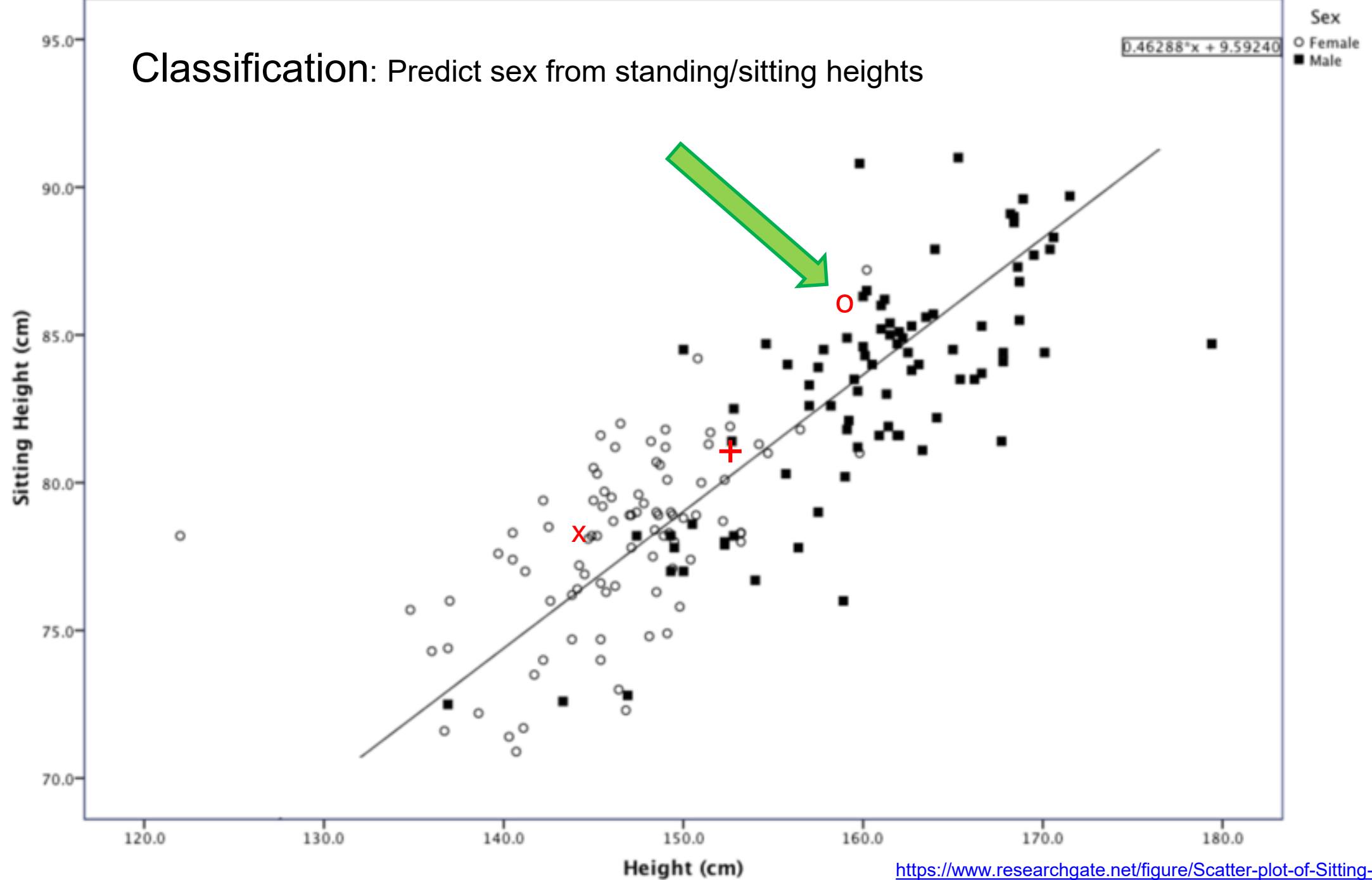
Classification: Predict sex from standing/sitting heights



Classification: Predict sex from standing/sitting heights



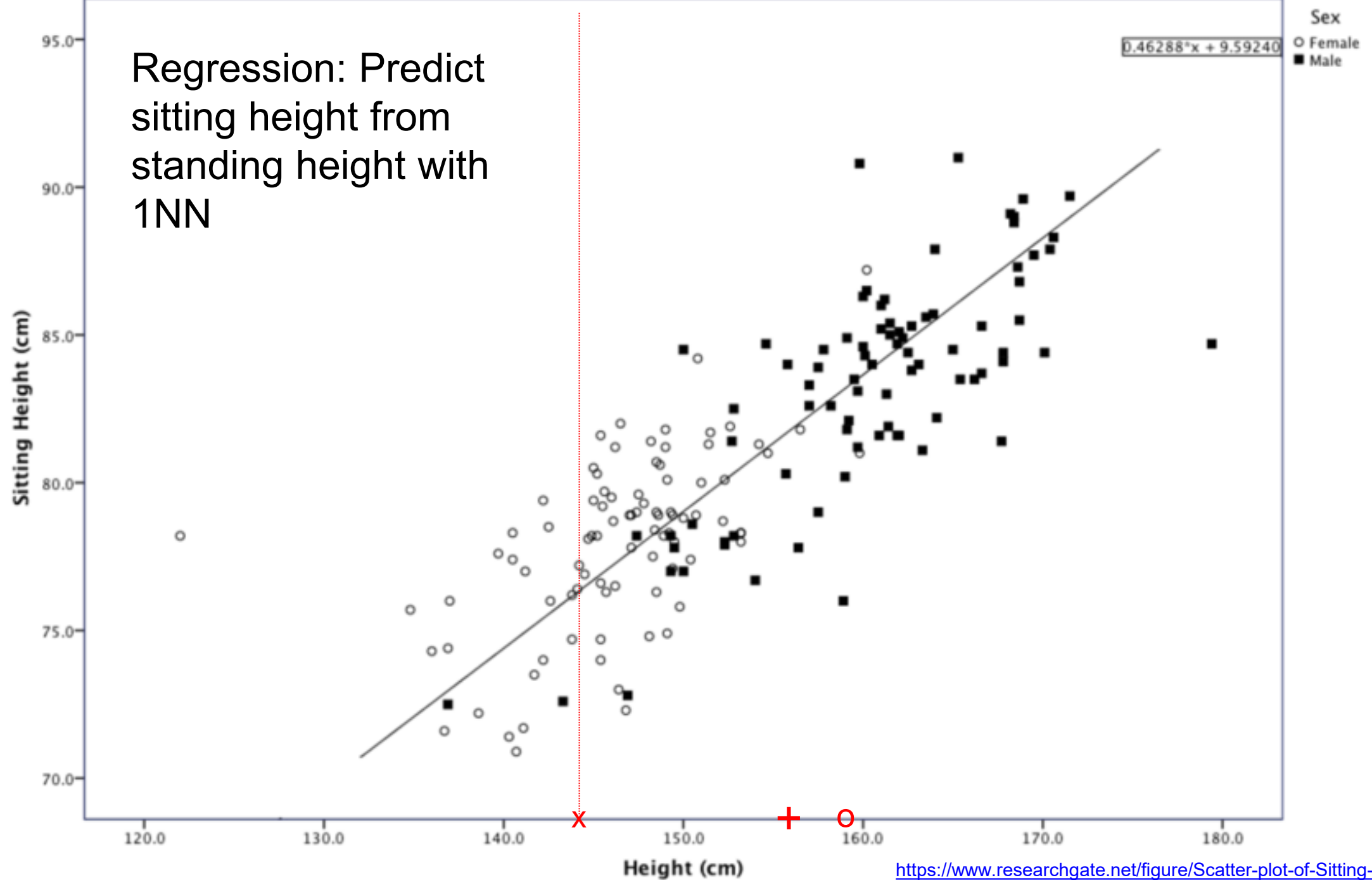
Classification: Predict sex from standing/sitting heights



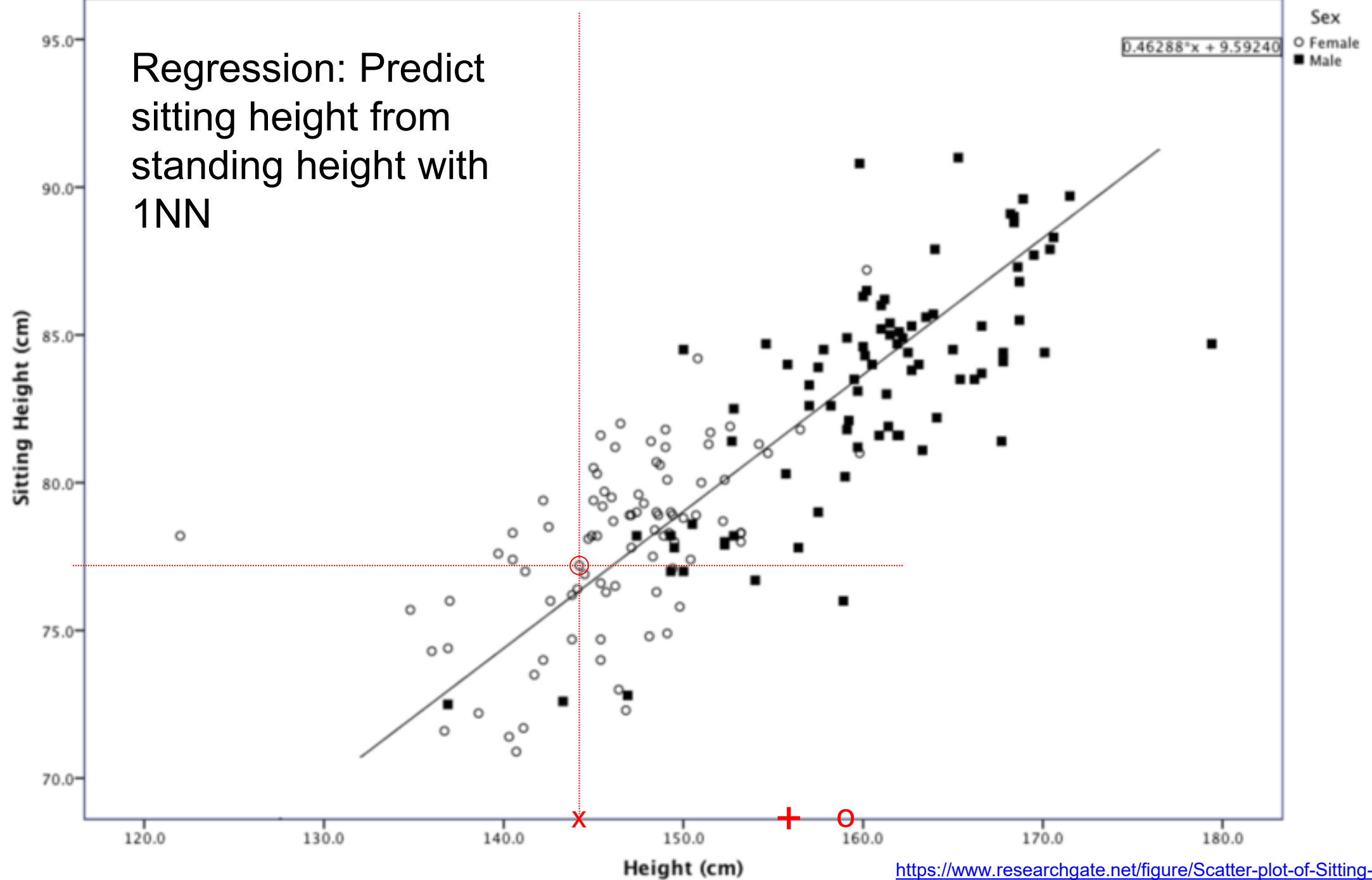
KNN Regression

- Also retrieve the K-nearest neighbors
- But, instead of predicting the most common retrieved label, predict the average of the returned values

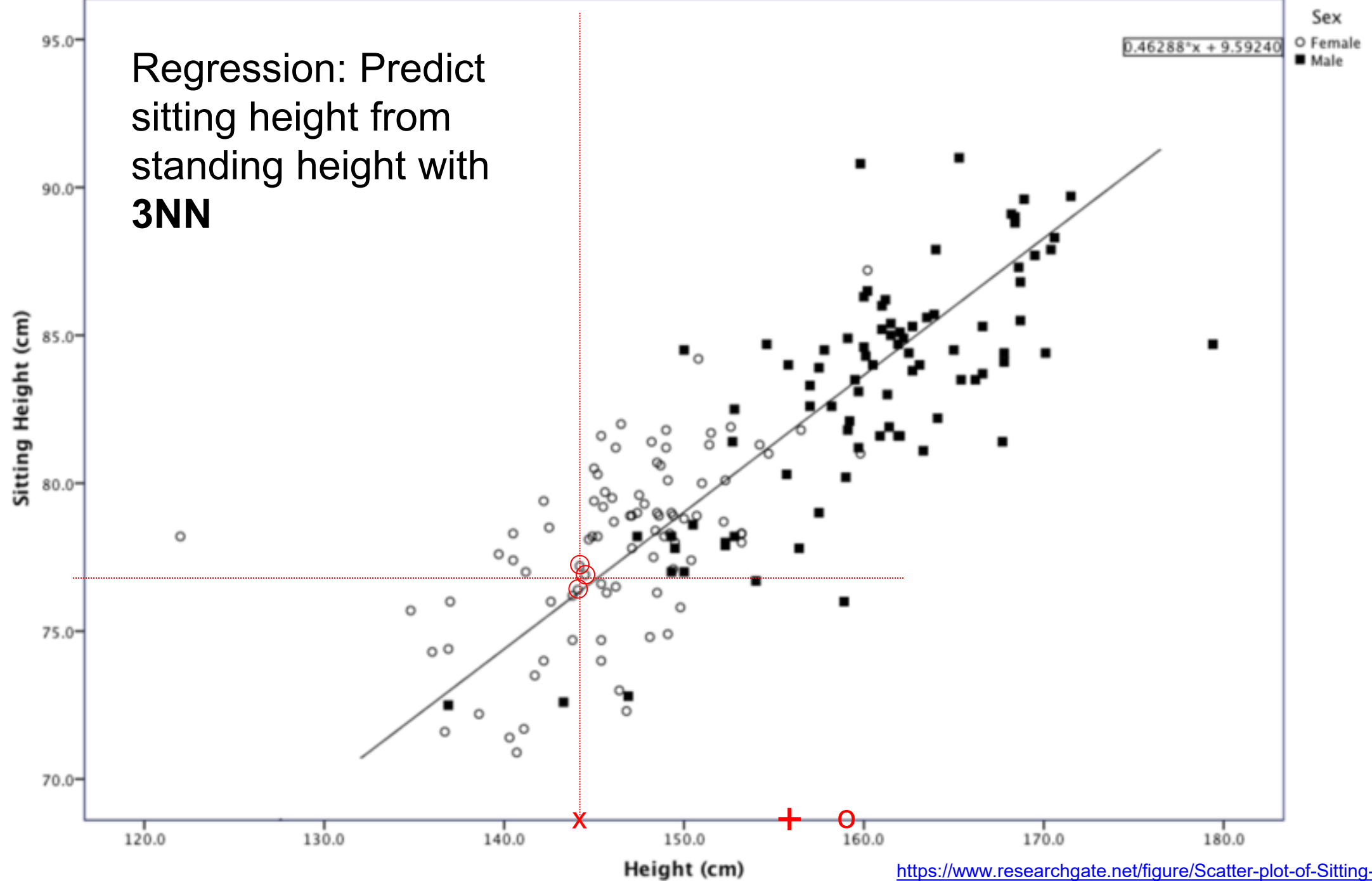
Regression: Predict
sitting height from
standing height with
1NN



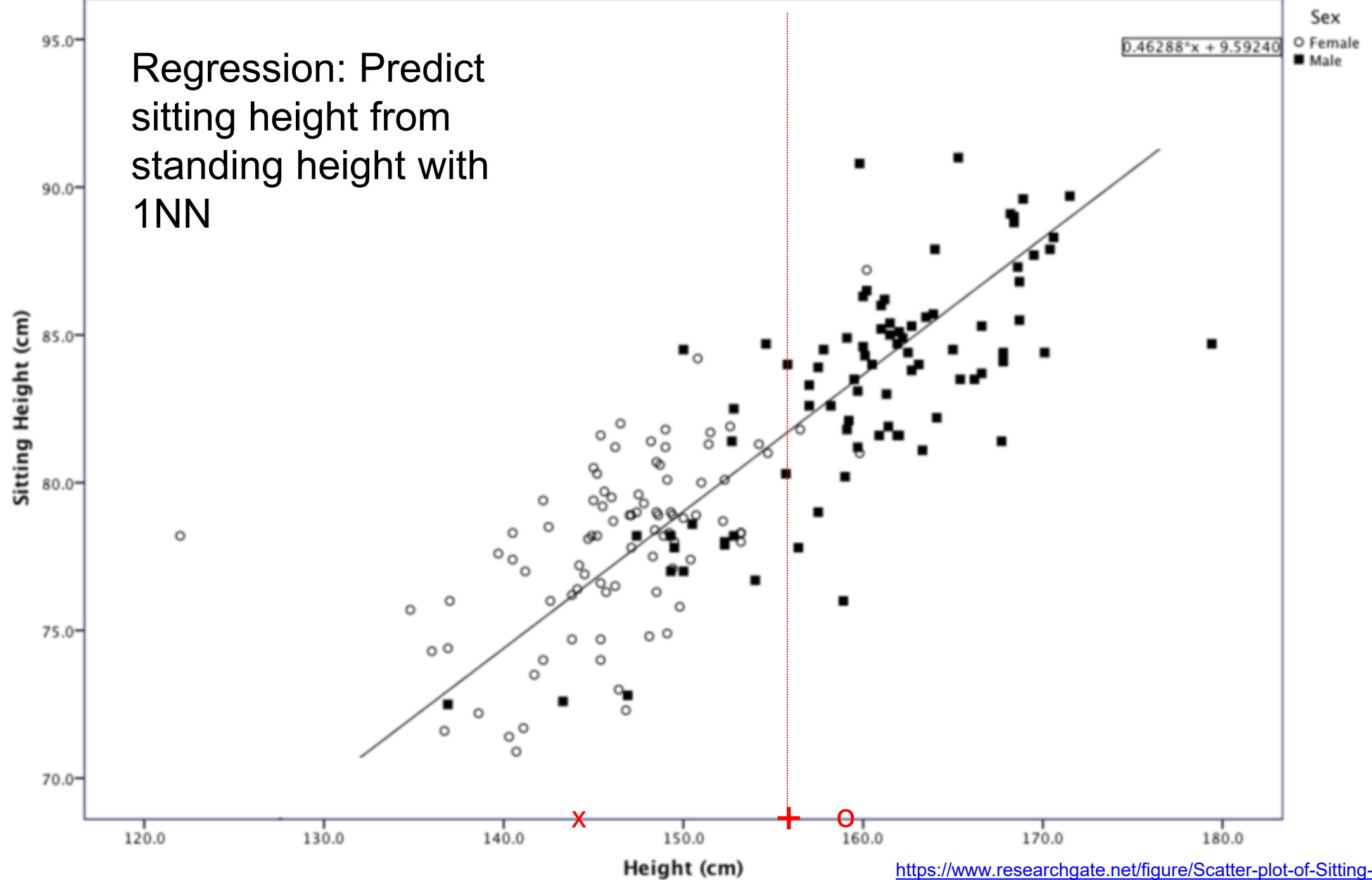
Regression: Predict
sitting height from
standing height with
1NN



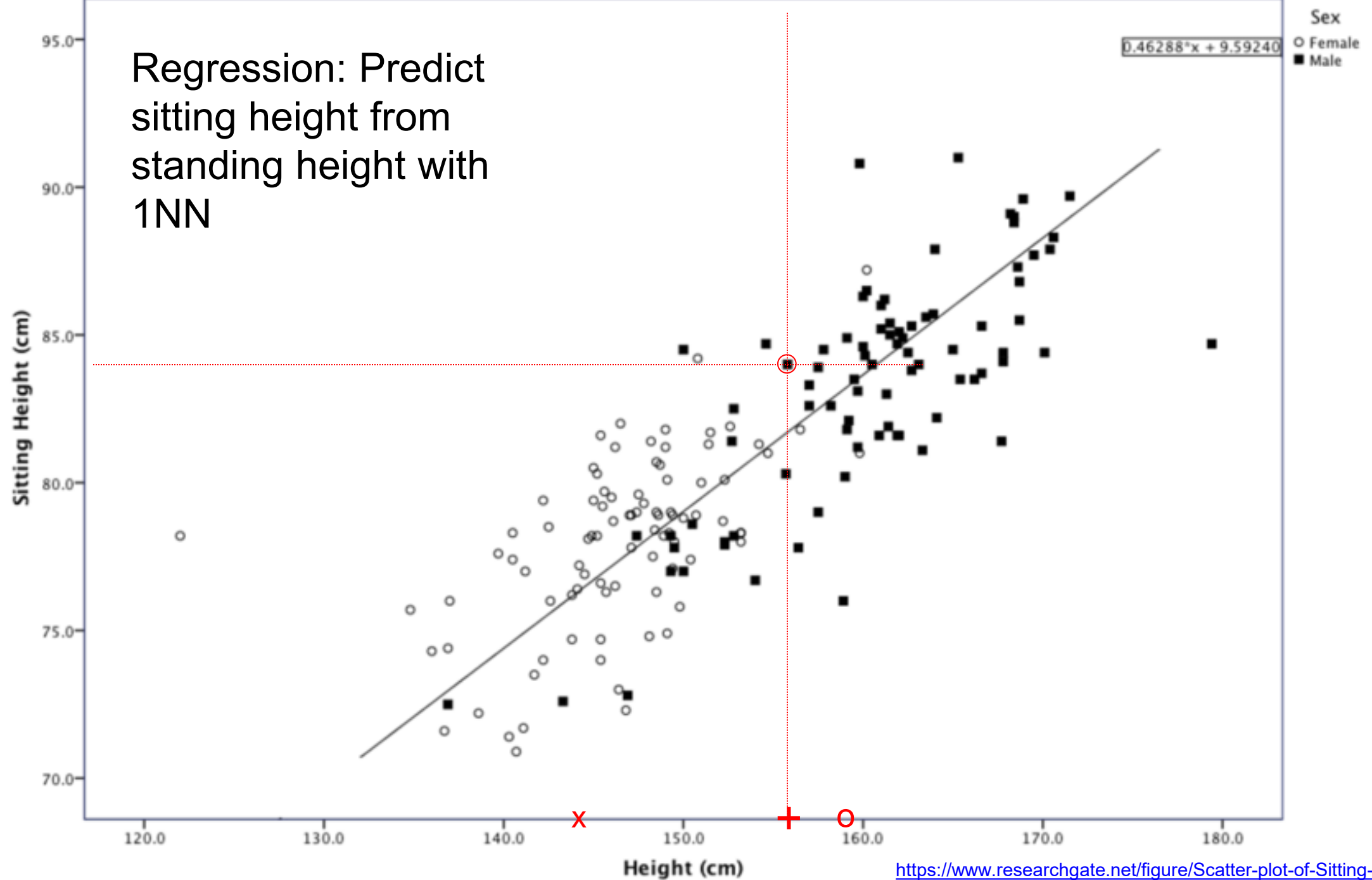
Regression: Predict
sitting height from
standing height with
3NN



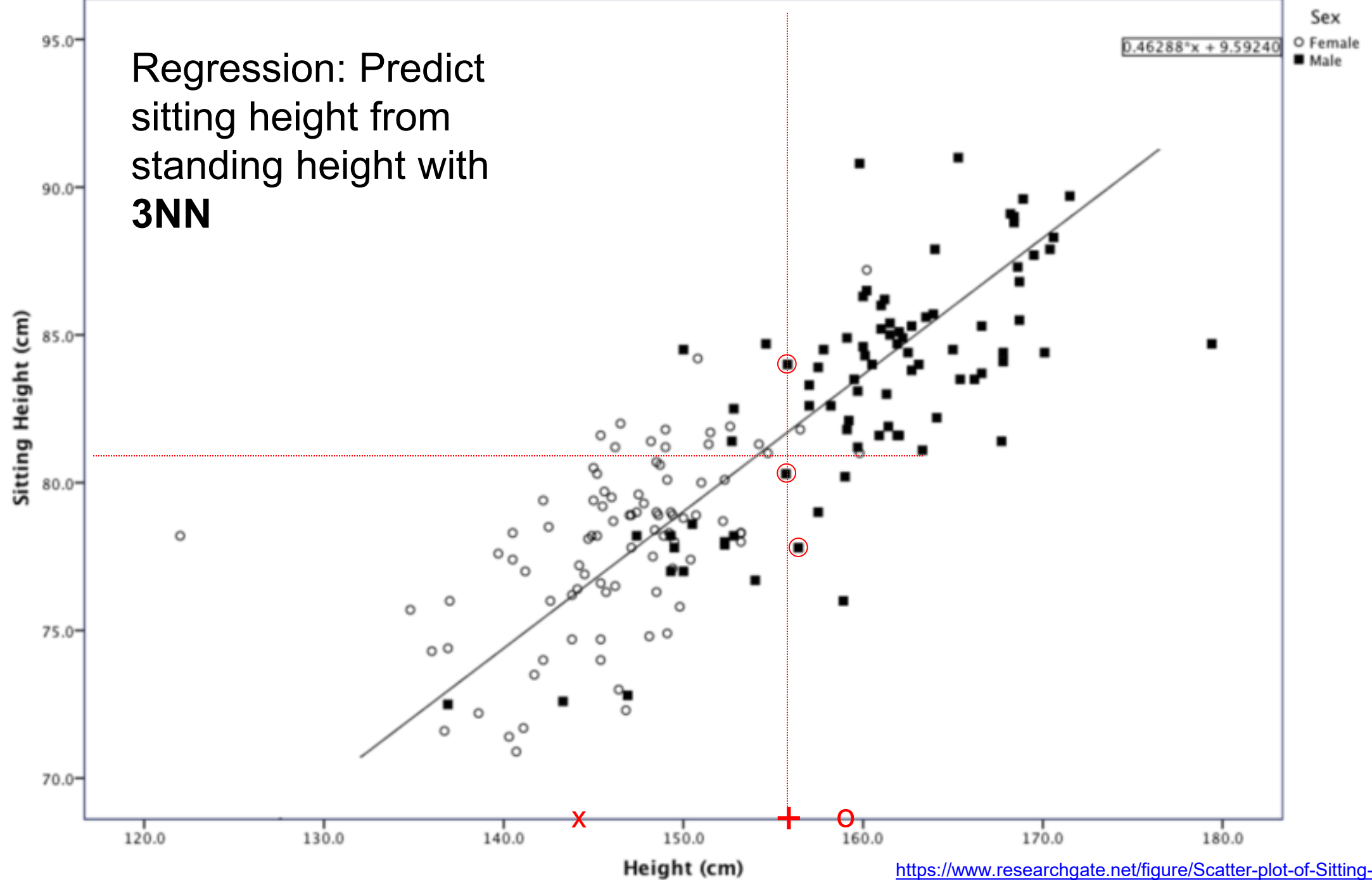
Regression: Predict
sitting height from
standing height with
1NN



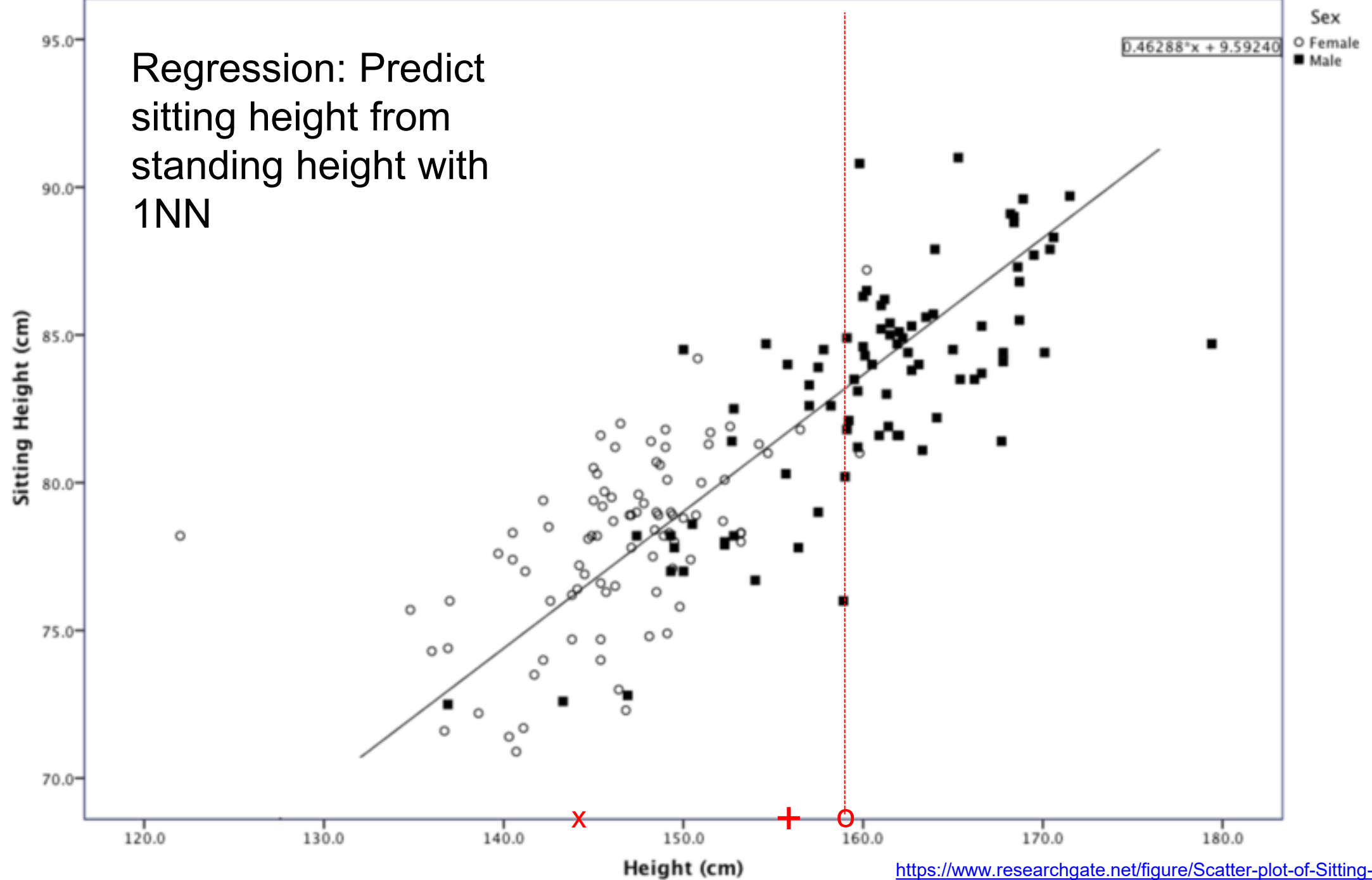
Regression: Predict
sitting height from
standing height with
1NN



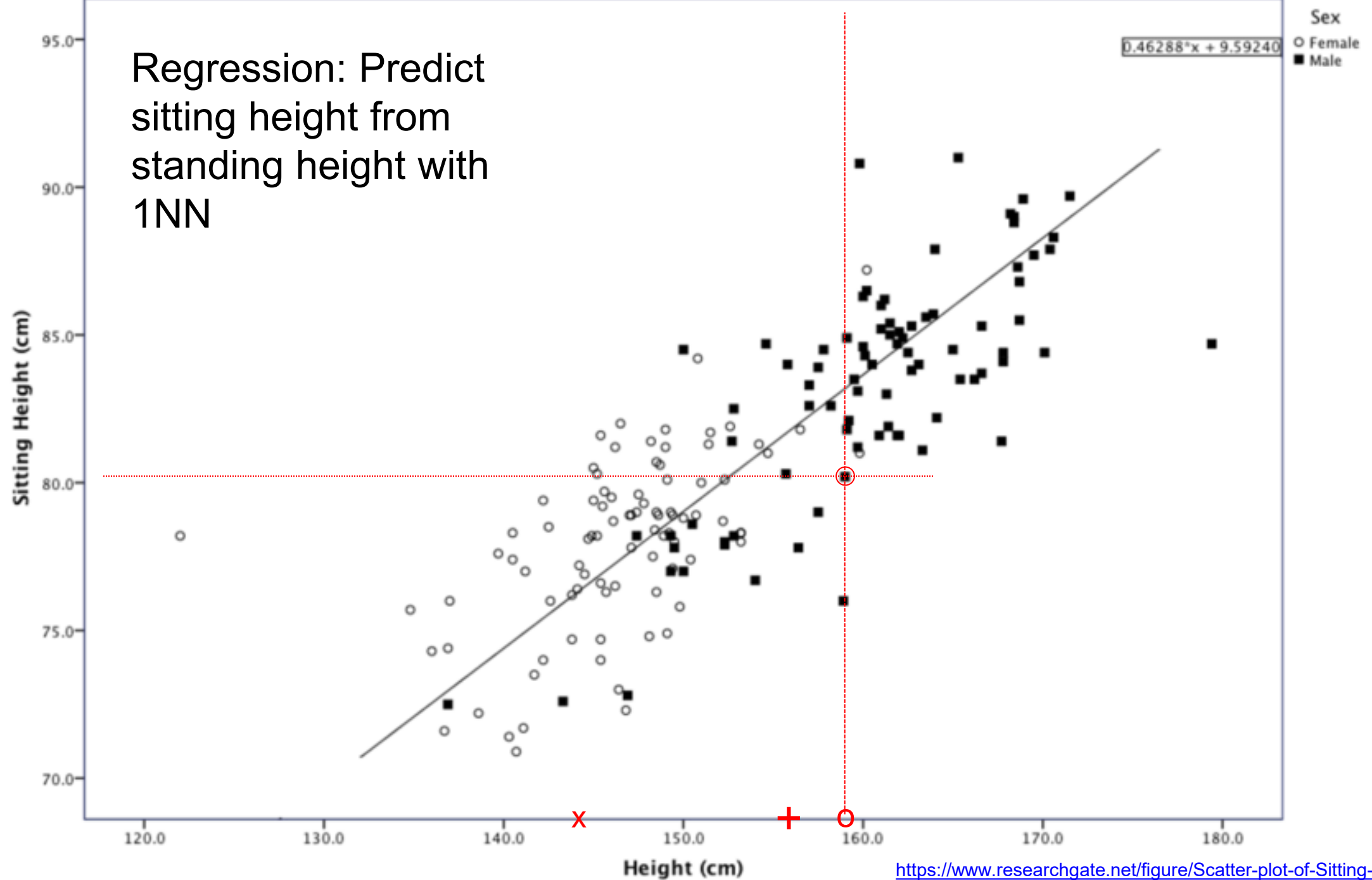
Regression: Predict
sitting height from
standing height with
3NN



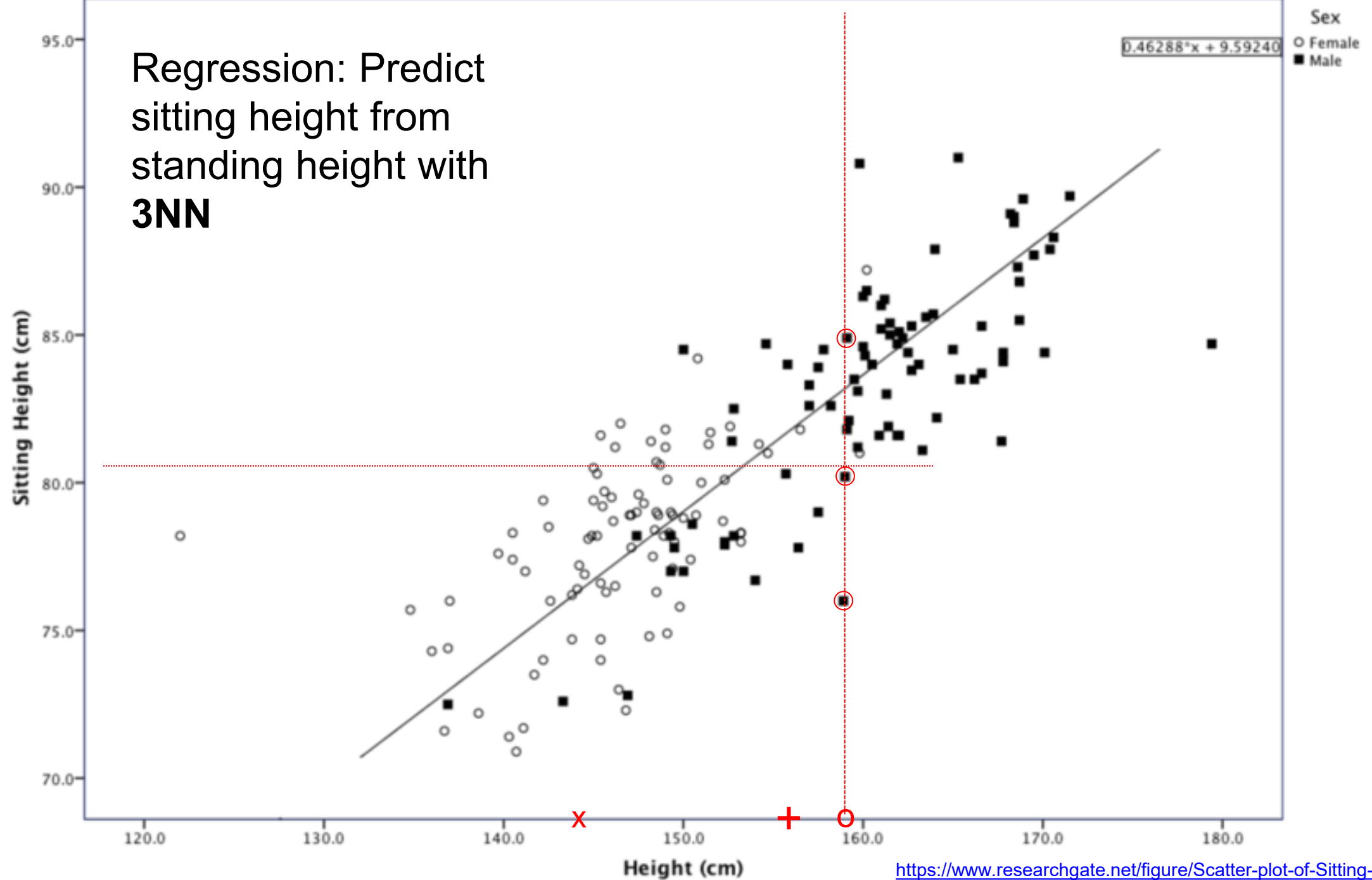
Regression: Predict
sitting height from
standing height with
1NN



Regression: Predict
sitting height from
standing height with
1NN



Regression: Predict
sitting height from
standing height with
3NN



How do we measure and analyze regression error?

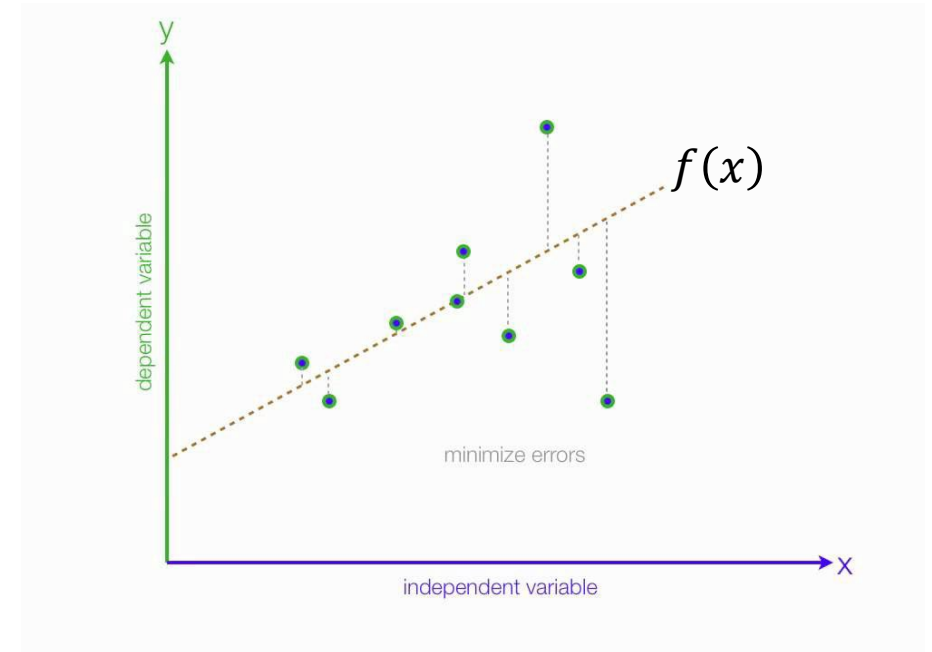
- Root mean squared error

$$\sqrt{\frac{1}{N} \sum_i (f(x_i) - y_i)^2}$$

- Mean absolute error $\frac{1}{N} \sum_i |f(x_i) - y_i|$

- $R^2: 1 - \frac{\sum_i (f(x_i) - y_i)^2}{\sum_i (y_i - \bar{y})^2}$ (unexplained variance)
(total variance)

- RMSE/MAE are unit-dependent measures of accuracy, while R^2 is a unitless measure of the fraction of explained variance



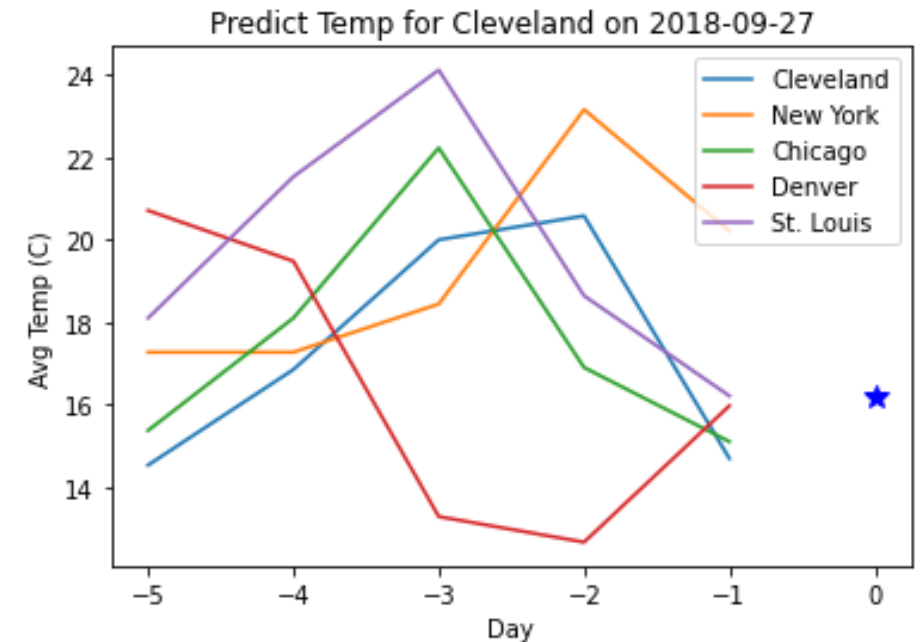
Q1-Q3

<https://tinyurl.com/AML441-L3>



Introducing the Temperature Regression Dataset

- Input: temperature (C) from 83 US cities for each of previous 5 days
 - Total of $415 = 83 \times 5$ features
- Target: temperature of Cleveland for next day
- Datasets
 - Train: 2555 samples (7 years of data, starting 2011-09-29)
 - Val: 365 samples (next 1 year of data)
 - Test: 365 samples (next 1 year of data)



KNN for Temperature Regression

```
def regress_KNN(X_query, X_train, y_train, K):  
  
    # (1) Compute distances between X_query and each  
    sample in X_train  
  
    # (2) Get the K_smallest_idx: K indices  
    corresponding to smallest distances (e.g. use  
    np.argsort)  
  
    # (3) Return the mean of y_train[K_smallest_idx]  
  
def RMSE(y_pred, y_true):  
    return np.sqrt(np.mean((y_pred-y_true)**2))
```

Testing procedure:

```
# Get y_pred[i] = regressKNN(X_test[i], X_train,  
y_train, K) for each ith sample in X_test  
  
# measure error: err = RMSE(y_pred, y_test)
```

Some things to consider

- The temperatures vary a lot over the year, which will reduce the number of examples with similar temperatures
 - What can we do?

Some things to consider

- The temperatures will vary a lot over the year, which will reduce the number of examples with similar temperatures
 - What can we do?
 - Reframe the problem by making all temperatures *relative* to previous day's temperature of Cleveland (the target city)
 - Samples around warm days can match those from cold days – only the shifts in temperature matter
- How do we choose K?

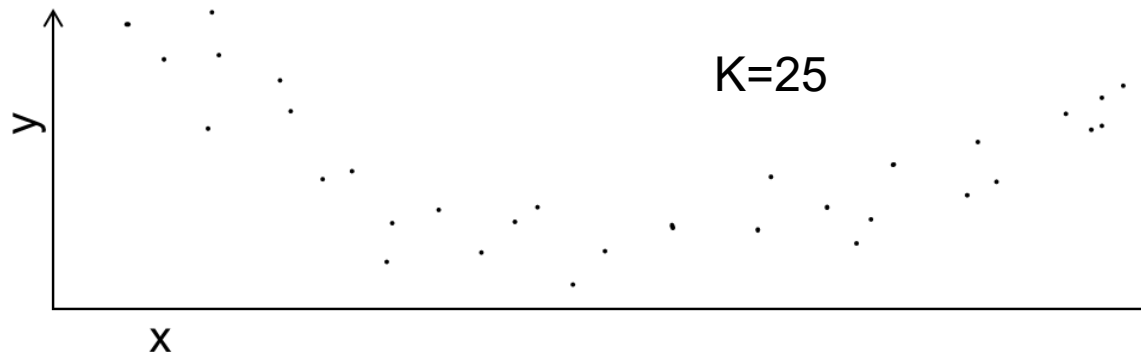
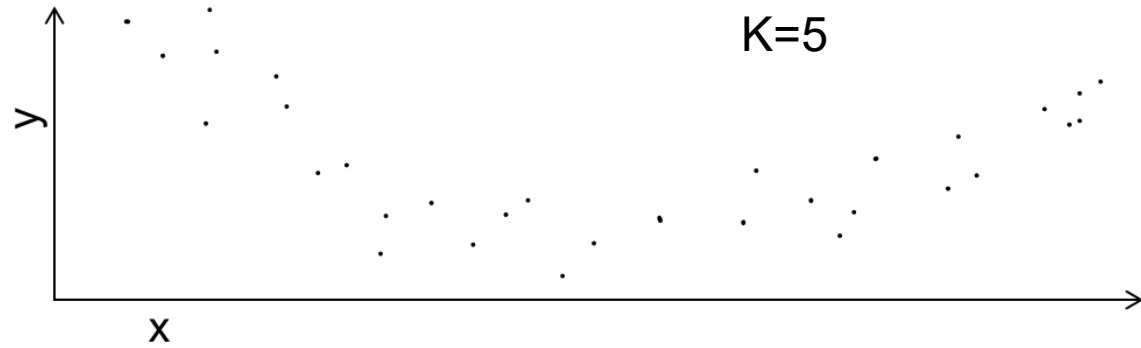
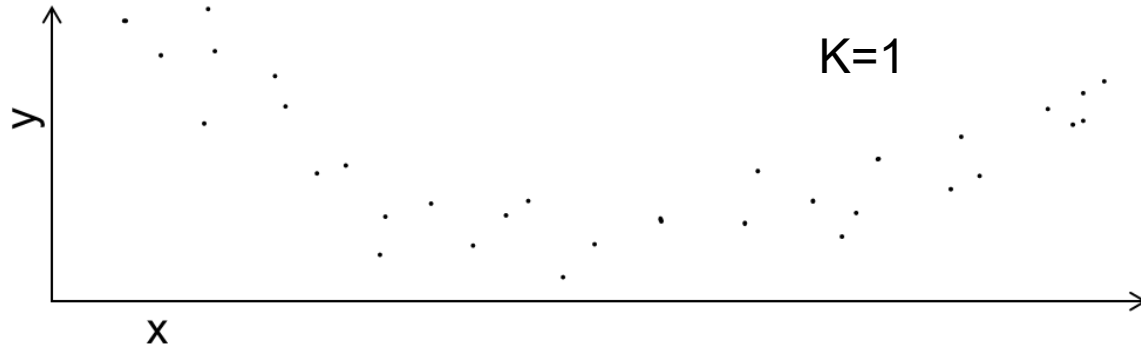
Choosing K Using a Validation Set

For each candidate K, e.g. $K=1, 3, 5, 9, 11, 25$:

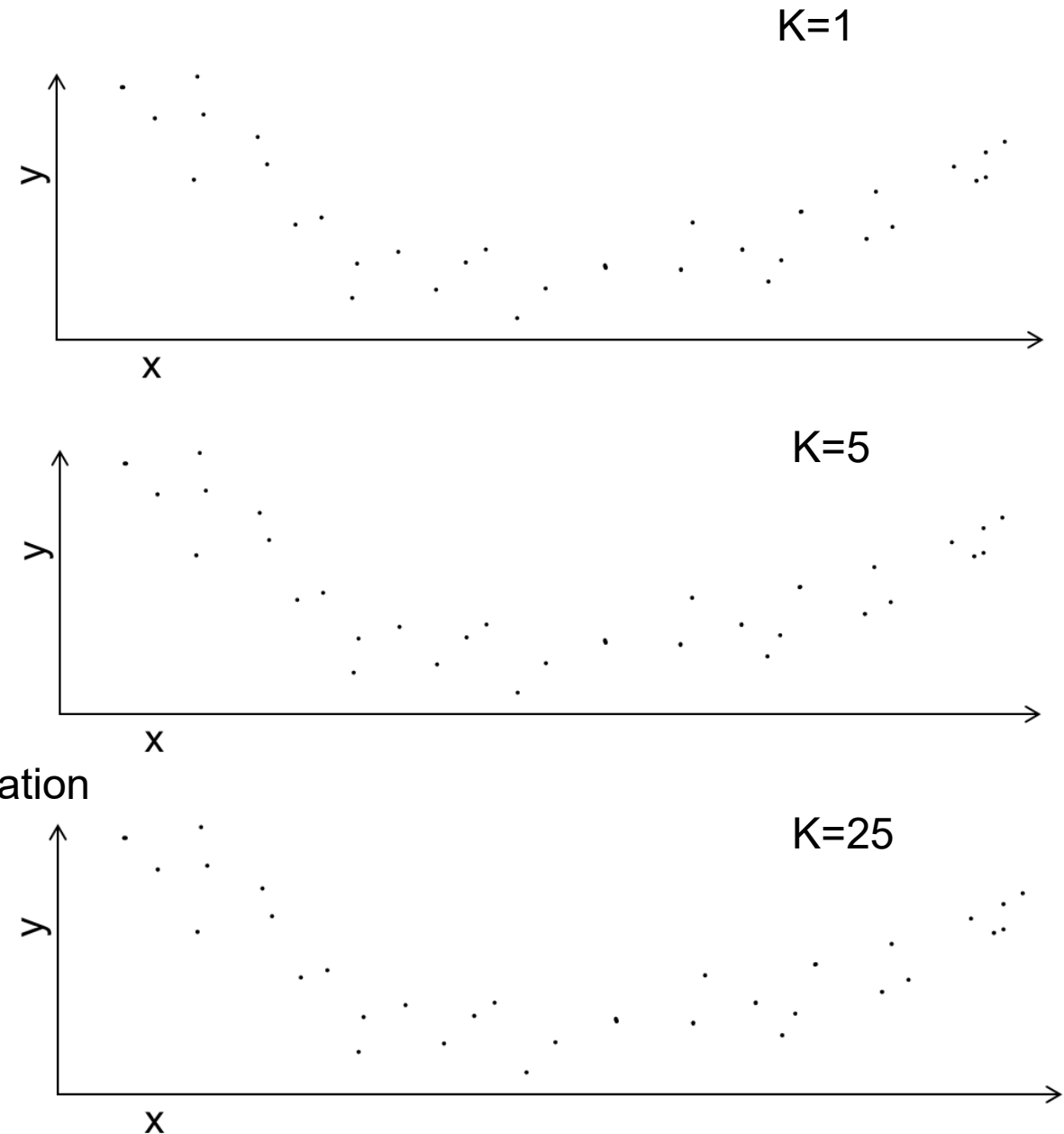
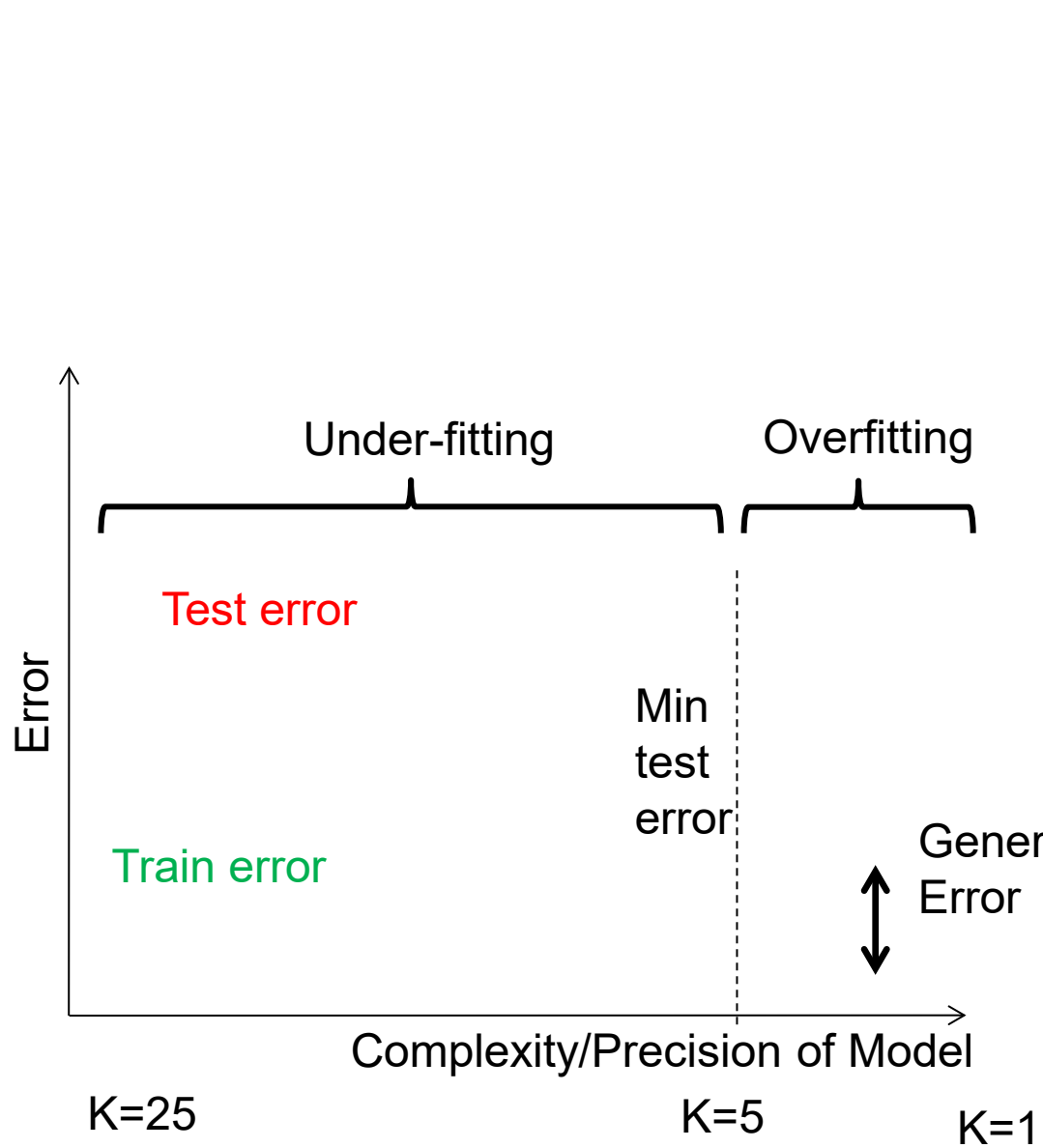
Evaluate error using the validation set

Select the K with the lowest validation error

Small K may “overfit” data, while large K may not be able to fit the true trend



Error and Bias Variance Trade-off



Error and Bias Variance Trade-off

When model parameters are fit to a *training set* and evaluated on a *test set*

- **Training error:** The error on the training set
- **Test error:** The error on the test set
- **Generalization error:** test error – training error *

Test error has three important sources in common ML settings:

- **Intrinsic:** sometimes it is not possible to achieve zero error given available features (e.g. handwriting, weather prediction)
 - Bayes optimal error: The error if the true function $P(y|x)$ is known
- **Model Bias:** the model is limited so that it can't fit perfectly to the true data distribution (e.g. there will be error, even if you have infinite training data)
- **Model Variance:** given finite training data, different parameters and predictions would result from different samplings of data

A more complex or specific model is expected to have

- Lower bias: better fit to training set
- Higher variance: more uncertainty in best parameters, so higher generalization error
- Could have higher or lower test error

* This is the course definition. Sometimes generalization error is defined as the test error.

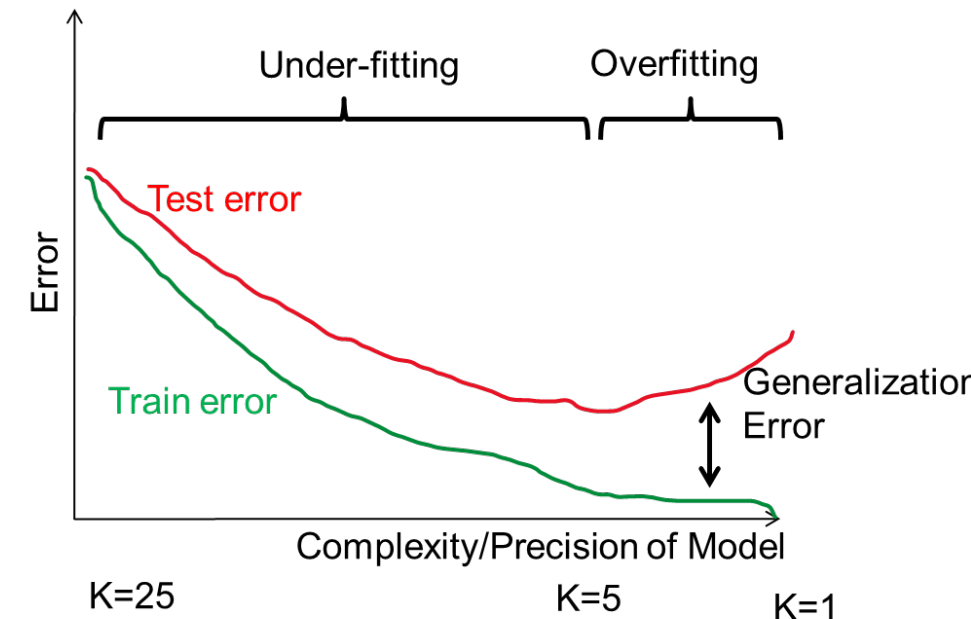
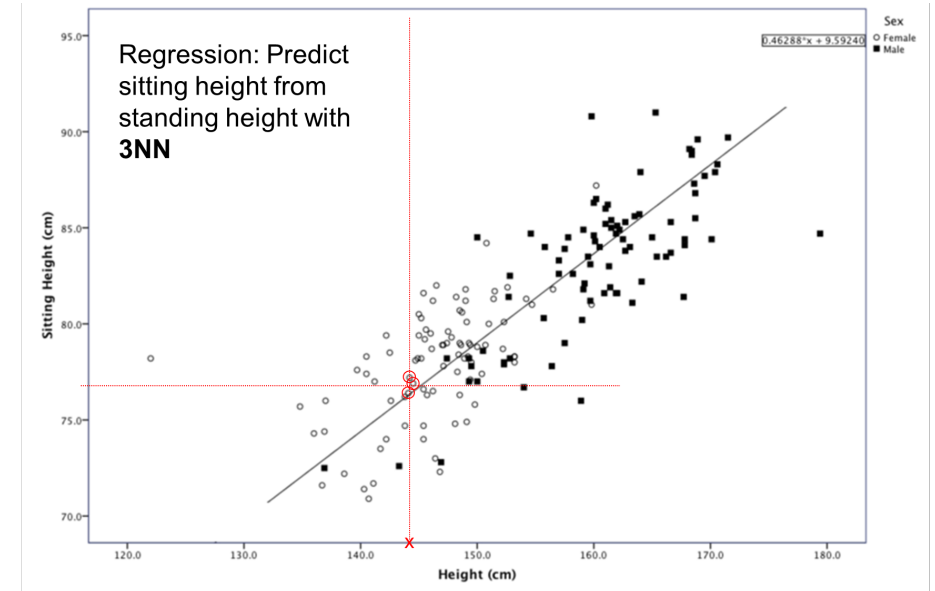
Q4-Q6

<https://tinyurl.com/AML441-L3>



Things to remember

- Similarity/distance measures: L1, L2, cosine
- KNN can be used for either classification (return most common label) or regression (return average target value)
- Test error is composed of
 - **Irreducible error** (perfect prediction not possible given features)
 - **Bias** (model cannot perfectly fit the true function)
 - **Variance** (parameters cannot be perfectly learned from training data)



Thursday

- Retrieval and clustering