Introduction to Data Science

(Lecture 9)

Dr. Mohammad Pourhomayoun

Assistant Professor

Computer Science Department

California State University, Los Angeles

Final Project!

Final Project!

- **Step 0:** Find your teammates! 2-5 students can team up (if you *really* have difficulties in finding teammates, let me know!).
- **Step 1:** You have the option to select your own project. Your project should be related to data science topics, and it should be challenging enough for this class (The TAs should approve your project before you want to start).
  - A great source with hundreds of interesting projects is: **www.kaggle.com/competitions** Feel free to visit this website and pick one of the projects. Make sure that you select a doable project w.r.t the time and due dates!
- **Step 2:** The team should see the TA (time/location is on CSNS) to describe the project details, project data and goals, and get your project confirmed **(Due Date: Fri, Oct. 11)**.
- **Step 3:** Design your own project plan and schedule, assign the tasks to team members, and start the project!

Next

Final Project!

- **Step 4:** Make sure that the team have meetings on timely basis (e.g. weekly meetings) to evaluate the project progress! Don't leave everything for last days!
- **Step 5:** By end of the semester, each team should **either** give a 10 min presentation to the class, **or** come to my office to present their results, describe their project, the developed methods and algorithms, and the RESULTS (As time permits!).
- **Step 6:** By end of the semester, Each team should submit a **project report** including project details, the developed methods, algorithms, and codes to address the projects requirements, and the responsibility of EACH TEAM MEMBER. A big part of your grade depends on the quality of your report!

Next

Final Project: Some Tips!

- **AN IMPORTANT RECOMMENDATION:** Select a project that sounds **interesting, exciting** and **fun** to you, NOT something that looks just easy to do! Let's enjoy this project and this class. I don't expect you to get "record breaking" results! I just care about your effort and how you deal with challenges! So, try to work on an exciting project that pleases you, try to have a lot of fun, and enjoy your work!

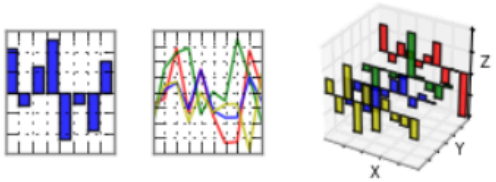
- The due dates for report and presentation will be announced later.
- Your work including algorithms, codes, and reports **must be your original work!** The teams are not allowed to use other people's work from kaggle website or any other resources!
- Some of the outstanding class projects can form the basis of future **paper publications**. Let me know if you are interested in such an opportunity!

Data Science with Python

IP[y]: IPython
Interactive Computing

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



scikits learn
machine learning in Python

NumPy

SciPy.org Sponsored By ENTHOUGHT

matplotlib

Scikit-Learn:
A Library for Data Science and Machine Learning

Scikit-Learn (sklearn)

- Scikit-learn is the Python Machine Learning Library.
- It includes optimal implementation of various **classification, regression** and **clustering** algorithms.
- It also includes hundreds of commands and functions for data preprocessing and processing along with a number of **default datasets** to work with.
- It is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
- Scikit-learn has an exceptional documentation.

IRIS Dataset

- **Recognizing flowers**
  - 150 sample flowers in three species (50 each).
  - Species of Iris (Labels): setosa, versicolor, virginica
  - Features: sepal length, sepal width, petal length, petal width



Important Hint about sklearn

- Sklearn only accept **NUMERICAL features**. Thus, we have to convert non-numerical (categorical) features into numerical values.
- **Note**: In converting features (and sometimes labels), we have to be cautious to avoid defining a confusing "ordering" between categorical values (we will talk about it later in this course).
- Depending on the classification algorithm, We usually use **LabelEncoding** to convert labels, and **OneHotCoding** to convert features.

5 Steps To Make Prediction In sklearn

- **Step1:** Importing the sklearn class (the machine learning algorithm) that you would like to use for prediction FROM sklearn library.
- **Step2:** Set up the Feature Matrix and Label Vector.
- **Step3:** Defining (instantiating) an "object" (instance) of the sklearn class as an initial predictive object.
- **Step4:** Training Stage: Train the above predictive model using the training dataset.
- **Step5:** Testing (Prediction) Stage: Making prediction on new observations (Testing Data) using the trained model.
- **Step6:** Evaluating the machine leanring model and results

Evaluating The Accuracy Of Our Predictive Model

Evaluating The Accuracy Of Our Predictive Model

Here is a simple way to evaluate the accuracy of our predictive model:

1- Let's split the dataset **RANDOMLY** into two new datasets: **Training Set** (e.g. 70% of the data samples) and **Testing Set** (30% of the data).

2- Let's **pretend** that we do **NOT** know the label of the Testing Set!

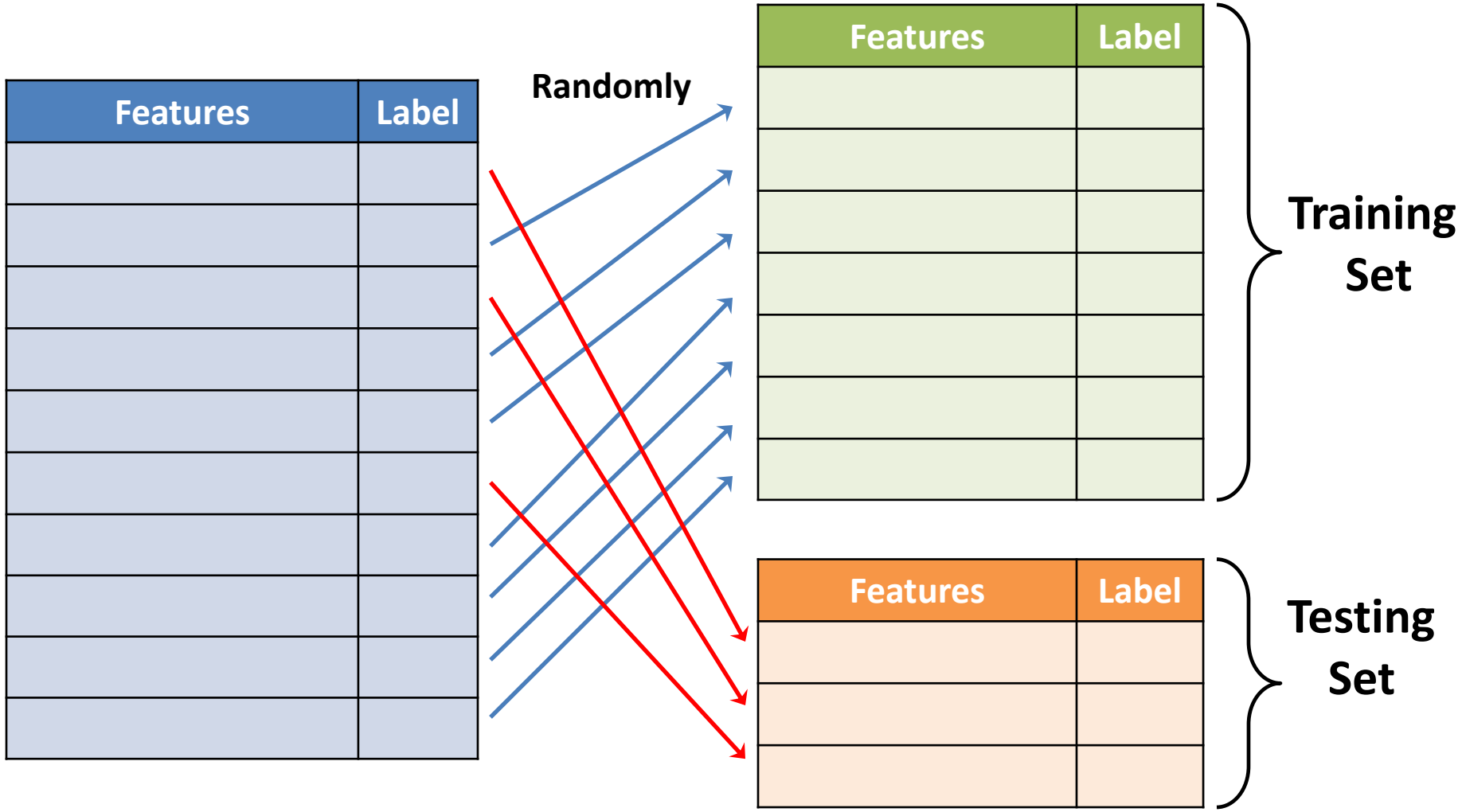
3- Let's Train the model **ONLY on Training Set**, and then Predict on the Testing Set!

4- After prediction, we can compare the **predicted labels** for the Testing Set with the **actual labels** of it to evaluate the accuracy of our prediction!

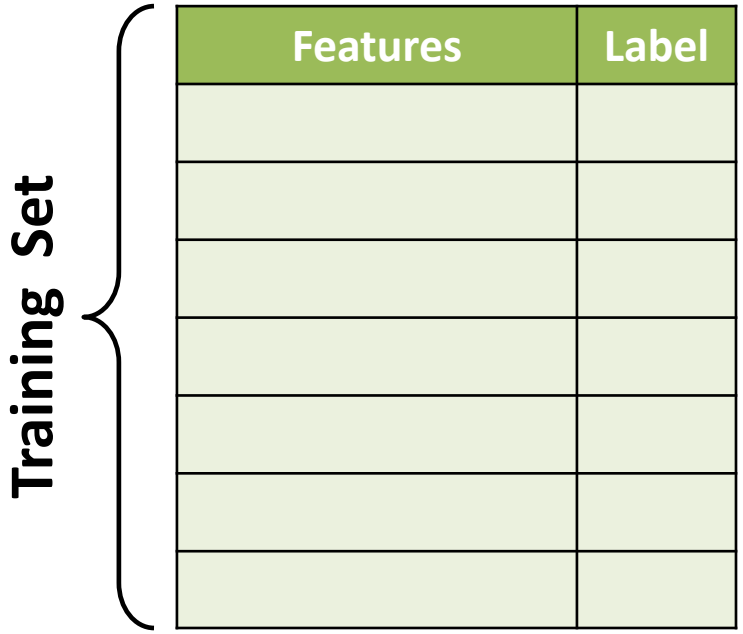
We will learn more techniques for model evaluation (e.g. **Cross Validation** method) later in this class!

Training and Testing Sets

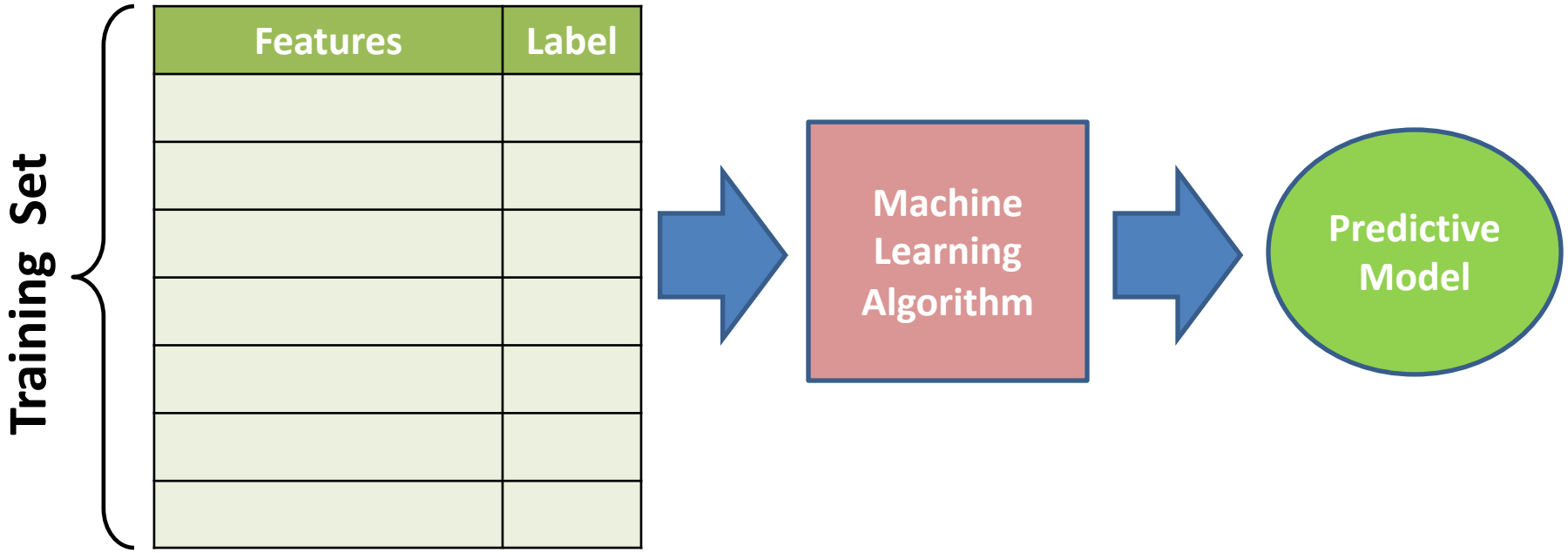
Training and Testing Sets



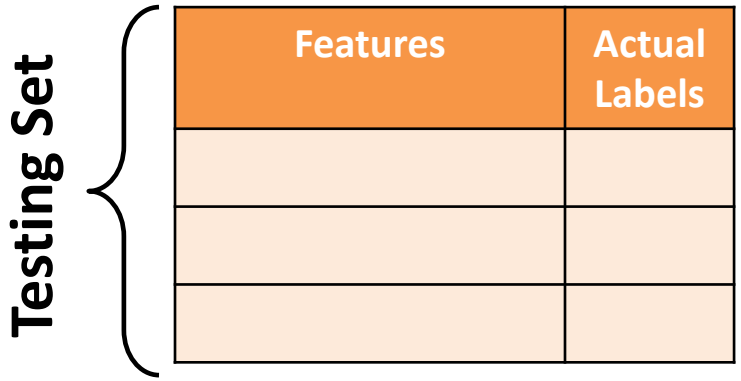
Training Stage



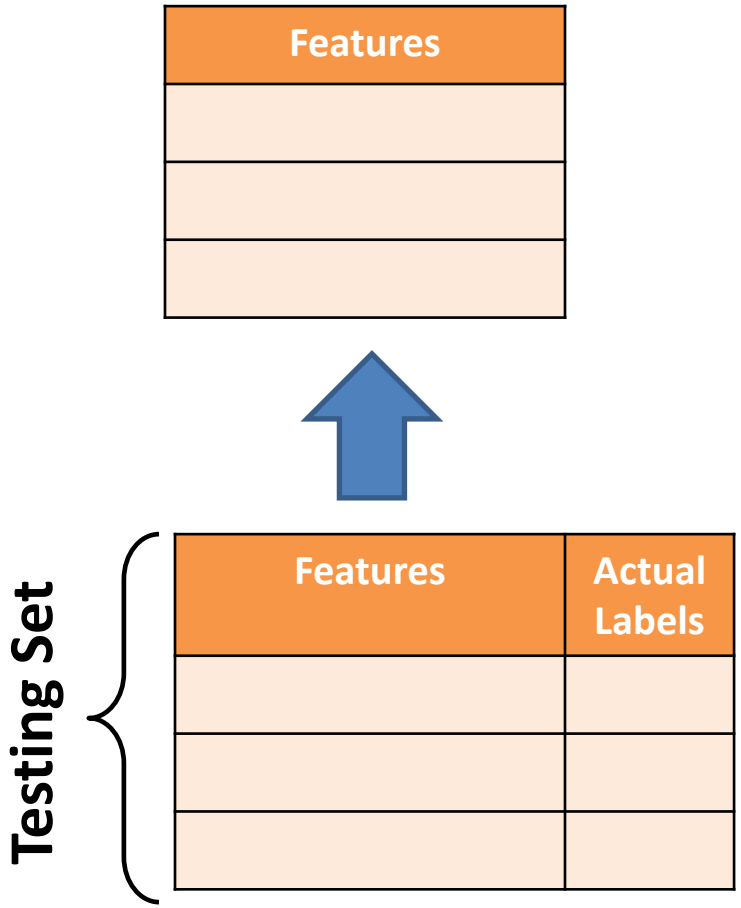
Training Stage



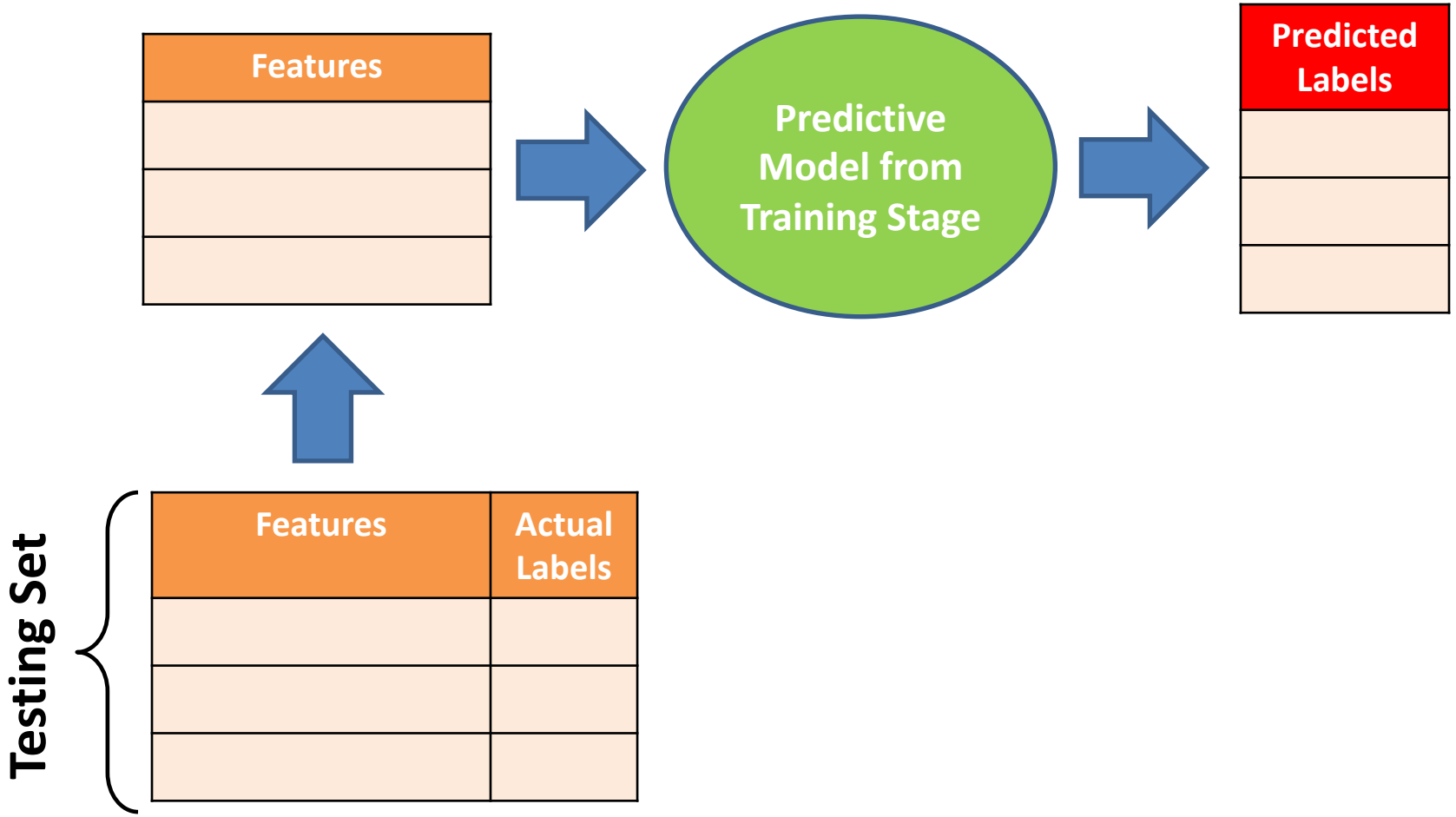
Testing Stage



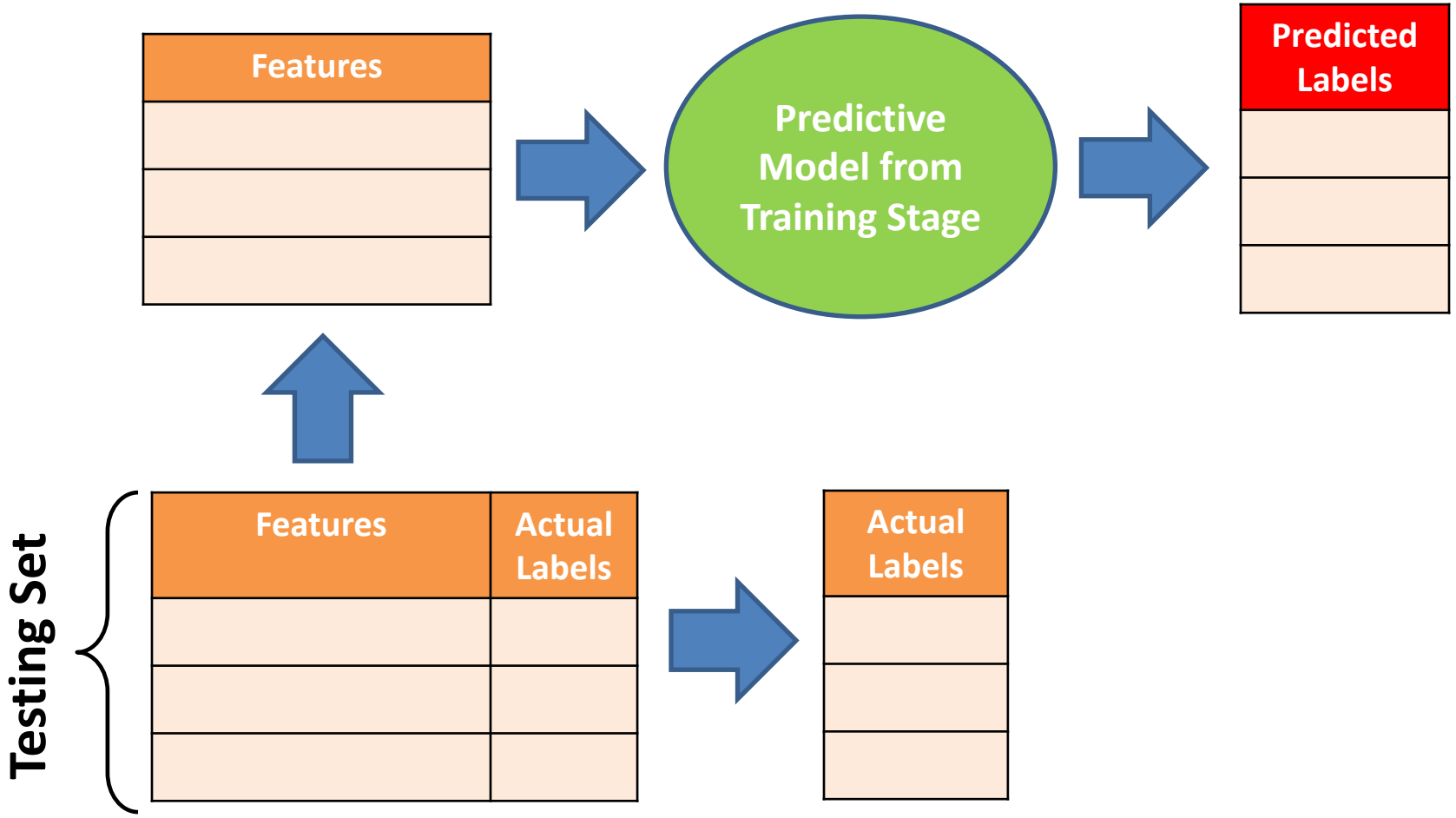
Testing Stage



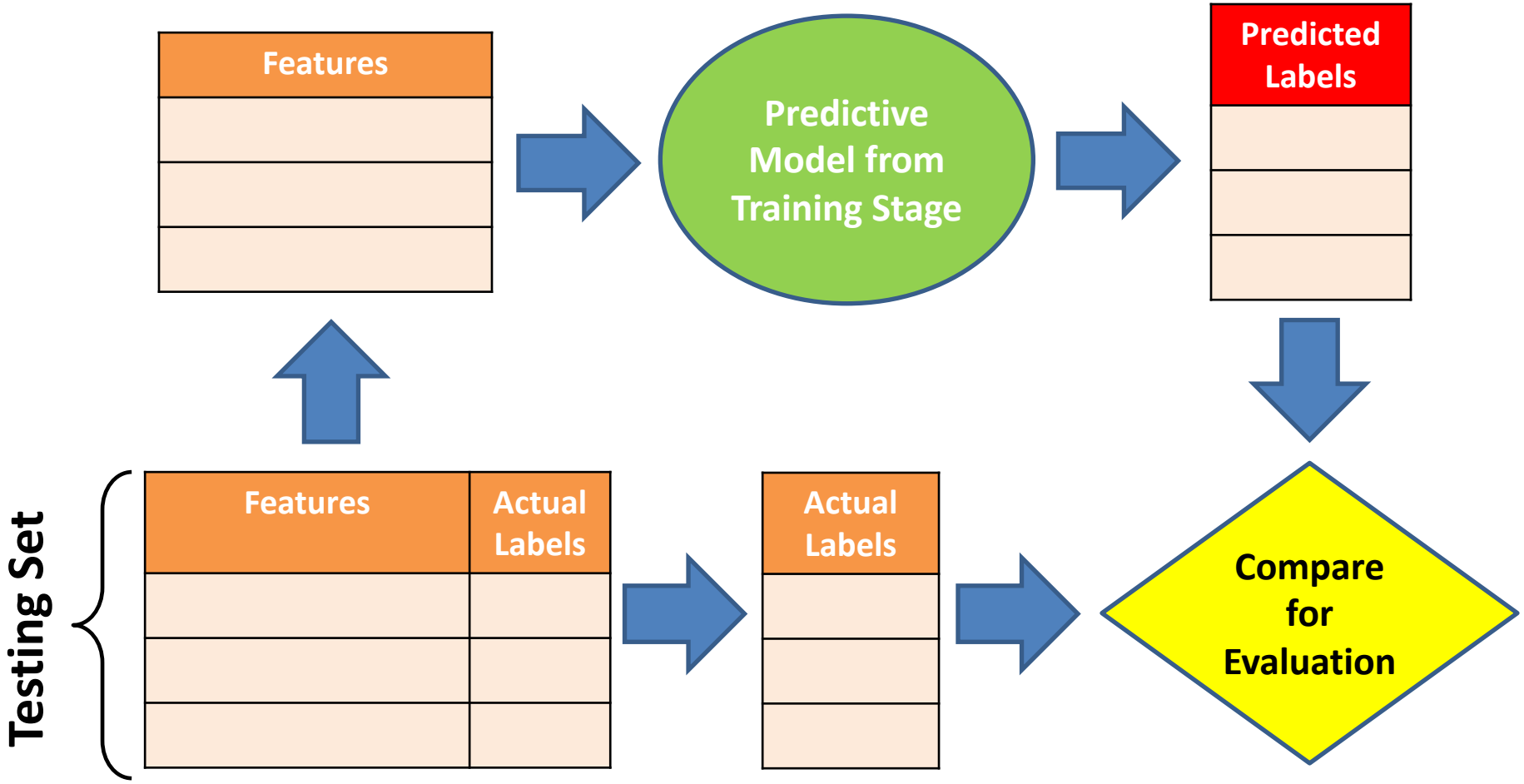
Testing Stage



Testing Stage



Testing Stage



Evaluating The Accuracy Of Our Predictive Model

VERY IMPORTANT: There must be NO OVERLAP between Training Set and Testing Set!

Evaluating The Accuracy Of Our Predictive Model

- **Note1:** Later, we will see that we can split the original dataset into 3 sets: **Training Set, Validation Set** , and **Testing Set**. In this case, We can use Validation set for adjusting the classifier parameters, and then use Testing Set for final evaluation.

- **Note2:** Later, we will also talk about **Cross-Validation** approach. In Cross-Validation, several rounds of partitioning will be applied to assure that all data samples are used both in training set and testing set but not simultaneously (NO OVERLAP!)

Data Science Practical Tutorial

- Let's open file ***CS4661-PythonDataScienceTutorial-Lab3.ipynb***
  in Jupyter notebook to continue the tutorial.