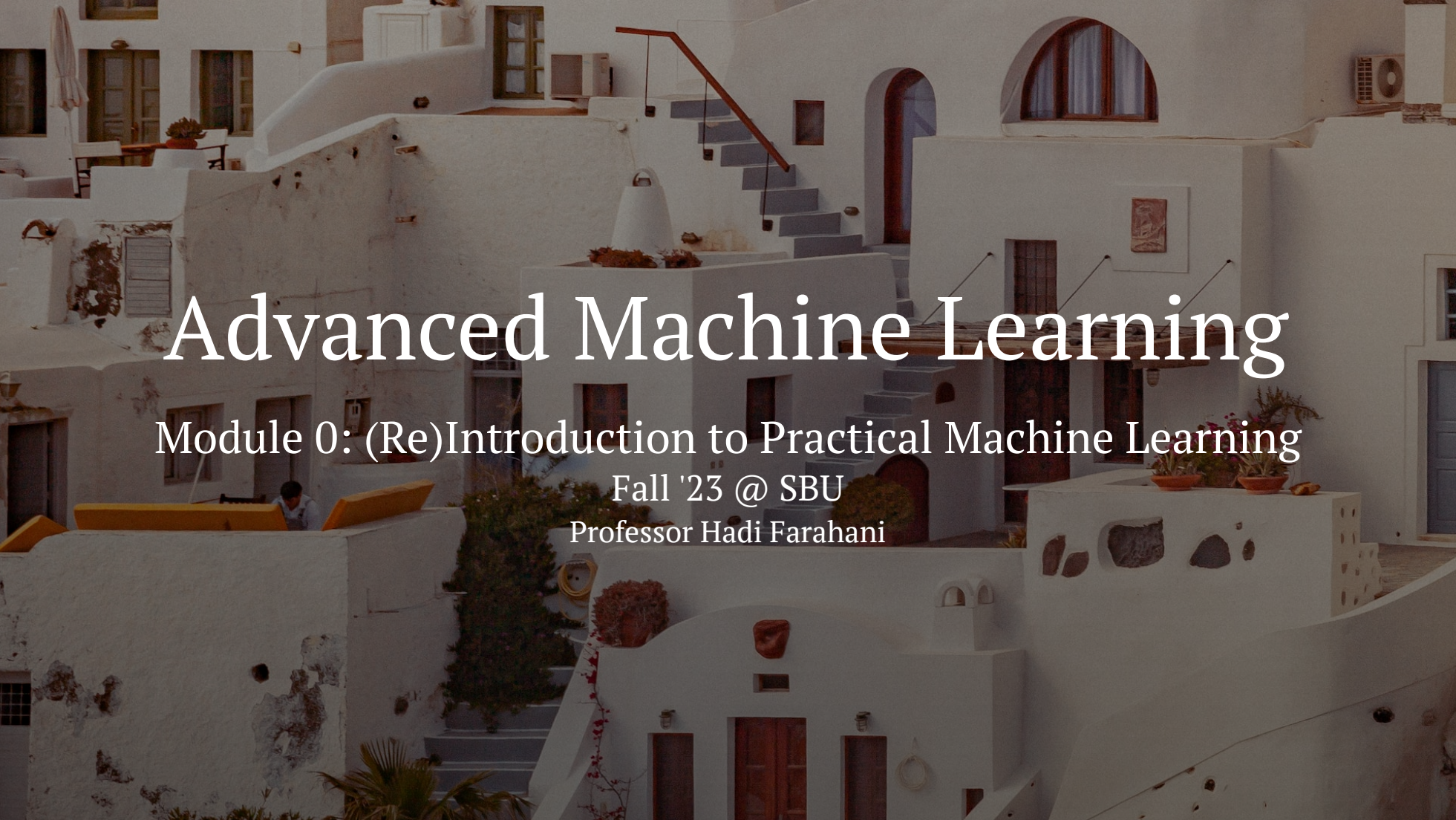


Advanced Machine Learning

Module 0: (Re)Introduction to Practical Machine Learning

Fall '23 @ SBU

Professor Hadi Farahani

Course Overview

- **Machine Learning Basics**: Building a model, data visualization and cleaning, feature engineering, working with popular data science and machine learning libraries such as Scikit-Learn, Pandas, Seaborn, etc.
- **Data Management**: Using data management techniques and methods popular in practical machine learning tasks, such as working with SQL, and MongoDB.
- **Deep Learning**: An introduction to deep learning and implementing artificial neural networks.
- **NLP**: An introduction to Natural Language Processing (NLP) and various concepts, such as machine translation, question answering, Large Language Models (LLMs) and various common mechanisms used in NLP, such as attention.
- **Image Processing**: An introduction to deep learning-based image processing and various common tasks, such as object detection, semantic segmentation and diffusion models.

Near the end of the course, you will be assigned a final project which will account for a large portion of your final grade.

> This course assumes that you have a basic understanding of machine learning, common models and a core understanding of artificial neural networks and how they work.

How Classes Will Work

1. **Core Concept**: The theory behind a certain concept is presented.
2. **Code Implementation**: A practical implementation of said concept using code is presented
3. **Homework**: Students are then expected to apply said concepts to a given problem

---

Telegram Group Chat:

(Link: https://t.me/+JAnc_FRwcDs0MTQ0)

Technical Details

Language used: **Python**

Some of the modules that we'll be using in this class include:

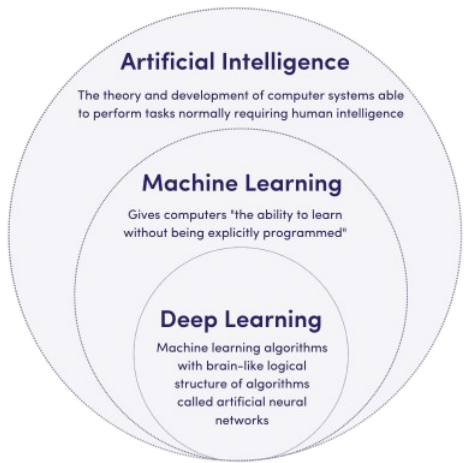
- PyTorch (for Deep Learning)
- Scikit-Learn (non-DL machine learning, data processing)
- Pandas (for data loading and management)
- Seaborn and Matplotlib (for data visualization and EDA)
- (and more!)

(Re)Introduction to ML

Machine learning is a subset of artificial intelligence that enables computers to learn from data and improve their performance on a task without being explicitly programmed.

Machine learning involves developing algorithms that allow computers to learn patterns and make predictions or decisions based on input data.

- It's different from statistics in the sense that we're more focused on building good autonomous agents rather than drawing conclusions from the data.



What ML is *not*:

One common misconception surrounding machine learning is that it possesses human-like intelligence and can fully understand complex concepts.

- In reality, it relies on algorithms that can detect patterns in data and make predictions, but it lacks true understanding.
- Machine learning models can be very biased if we're not careful, and it's up to us to make sure that does not happen.

> In this course, we'll be exploring practical machine learning, and state-of-the-art algorithms and how we might be able to apply them to real-world scenarios.

ML in Practice

While machine learning in *theory* is relatively simple — applying a statistical model to a predetermined data set with various features — but in practice there's a lot of things that need to be taken into account.

Over the next couple of weeks we'll be focusing on implementing a practical ML models and applying various algorithms.

After that, you'll be asked to do the same (homework).

ML in practice:

Machine learning has two main aspects. Without one, you cannot have the other:

1. **Data**
2. **Model**

> *(Why?)*

General Flow of Building a Machine Learning Model

The process and flow of building a machine learning model typically involves the following steps:

1. **Data collection**: Gather relevant and representative data for the problem you are attempting to solve. This data should include both input features and corresponding output labels or target values.

2. **Data preprocessing**: Cleanse and prepare the collected data for analysis. This step may involve handling missing values, handling outliers, scaling or normalizing features, and splitting the data into training and testing sets.

3. **Feature engineering**: Derive new features or select relevant features from the available data that can improve the model's performance. This step might involve techniques like dimensionality reduction, feature extraction, or the creation of synthetic features.

4. **Model selection**: Choose an appropriate machine learning algorithm or model that suits the problem at hand. The selection may depend on the type of problem, available data, and desired outcome. Some common types of models include linear regression, decision trees, support vector machines, and neural networks.

5. **Model training**: Use the training data to train the selected model. During this step, the model learns the patterns and relationships in the data to make predictions or classifications. The training process usually involves finding optimal values for the model's parameters or weights.

General Flow of Building a Machine Learning Model (Cont.)

6. **Model evaluation**: Assess the performance of the trained model using the testing data set. This evaluation helps measure how well the model generalizes to unseen data. Various metrics like accuracy, precision, recall, and F1 score can be used to evaluate the model's performance.

7. **Model optimization**: Fine-tune the model to optimize its performance. This step may involve hyperparameter tuning, regularization, cross-validation, or using more advanced techniques like ensembling to improve the model's accuracy.

8. **Deployment and prediction**: Once satisfied with the model's performance, deploy it to make predictions on new, unseen data. This can involve integrating the model into a larger software system or creating an API for real-time predictions.

9. **Monitoring and maintenance**: Continuously monitor the model's performance and re-evaluate periodically. As new data becomes available, update the model to ensure it remains accurate and relevant.

> It's important to note that the exact flow and steps may vary depending on the specific problem, data, and requirements of the project.

Data in Machine Learning

> Data in machine learning refers to a collection of observations or examples that are used to train or build a model.

It consists of features or attributes that describe the characteristics of the observations, and labels or target variables that define the desired outcome or prediction for each observation.

The role of data in model creation is crucial as it serves as the foundation for training and evaluating machine learning models.

By analyzing and learning patterns from the data, models are able to make predictions or decisions on new, unseen data.

Data Importance

A deep understanding of our data is *vital*.

1. **Data quality**: Understanding our data helps us to identify and address any issues related to data quality. This may include missing values, outliers, or inconsistencies in the data.

2. **Feature selection**: A deep understanding of the data allows us to select the most relevant and informative features for the model. Features that have a strong relationship with the target variable can significantly improve the model's accuracy and generalization capabilities.

3. **Bias and fairness**: Data can exhibit certain biases that might be inadvertently learned and reinforced by the model.

Data Importance (Cont.)

4. **Model performance and interpretation**: Understanding the data enables us to select and configure the appropriate machine learning algorithms and models that can best capture the underlying patterns and relationships in the data. Moreover, it helps us interpret the model's predictions and understand its limitations, ensuring that the model aligns with the real-world context and expectations.

5. **Generalization and robustness**: A deep understanding of the data aids in building a model that can generalize well to unseen data. By knowing the characteristics, distribution, and potential variations of the data, we can build a more robust and reliable model that performs well beyond the training data.

> Overall, having a deep understanding of our data allows us to make informed decisions throughout the machine learning process, resulting in more accurate, fair, and impactful models.

Jumping into Code

So, how do we work with data in Python?

During the duration of this course, we'll be using the *very* popular Pandas library for handling our data.

Why Pandas?

Pandas is preferred in machine learning projects for its efficient data manipulation and analysis capabilities, allowing users to easily clean, explore, and transform large datasets. Additionally, it seamlessly integrates with other popular libraries like NumPy and Scikit-learn, providing a comprehensive ecosystem for data preprocessing, modeling, and evaluation.

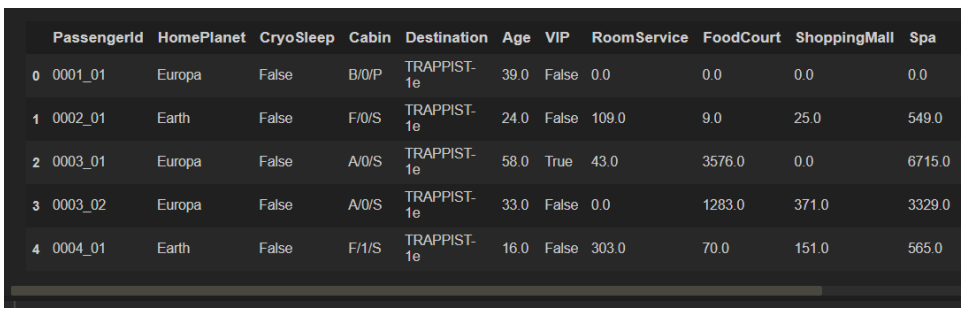
Loading in Our Data

```
import pandas as pd

df = pd.read_csv('filename.csv')

data.head()
```

Our output would look a bit like this:



	PassengerId	HomePlanet	CryoSleep	Cabin	Destination	Age	VIP	RoomService	FoodCourt	ShoppingMall	Spa
0	0001_01	Europa	False	B/0/P	TRAPPIST-1e	39.0	False	0.0	0.0	0.0	0.0
1	0002_01	Earth	False	F/0/S	TRAPPIST-1e	24.0	False	109.0	9.0	25.0	549.0
2	0003_01	Europa	False	A/0/S	TRAPPIST-1e	58.0	True	43.0	3576.0	0.0	6715.0
3	0003_02	Europa	False	A/0/S	TRAPPIST-1e	33.0	False	0.0	1283.0	371.0	3329.0
4	0004_01	Earth	False	F/1/S	TRAPPIST-1e	16.0	False	303.0	70.0	151.0	565.0

> If we're confused about what different features mean, we can always check the site that we downloaded the dataset from. (Spaceship Titanic Link)

(Onto code)

Data Visualization and EDA

EDA (Exploratory Data Analysis) is a crucial step in machine learning that involves analyzing and visualizing the datasets to gain insights and understanding of the data's characteristics.

- It helps identify patterns, uncover relationships, detect anomalies, and determine the quality of the data.

EDA allows data scientists to make informed decisions about data preprocessing, feature engineering, and model selection, ultimately improving the performance and accuracy of machine learning models.

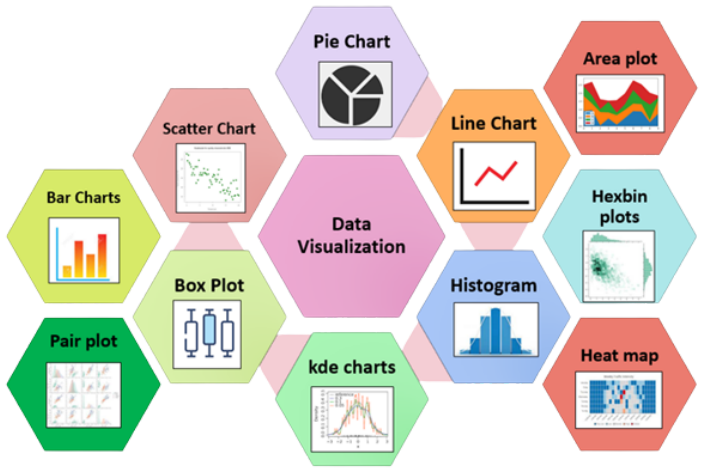
Data visualization, on the other hand, is the graphical representation of the data, aiding in interpreting complex information and presenting it in an easily understandable format.

> TL;DR: Very simply put, data visualization is a *subset* of EDA, and EDA focuses on statistical analysis of the data.

Data Visualization

Data visualization helps give us "hints" about our data and any trends that might exist.

- In many cases, data can be overwhelming and difficult to comprehend in its raw form. Data visualization simplifies complex data sets by representing them visually, making it easier for people to grasp and interpret the information.



It might seem like a waste of time now, but in practice when we're at a loss as to how we might want to visualize our data, data visualization may be invaluable.

Depending on whether or not our data is categorical or numerical, and what fields we're plotting, the type of graph we'll use will be different.

> A good source for picking the right kind of graph is from Data to Viz

Practical Data Visualization

Some things to look for when picking what to visualize include:

1. **Missing Values**: Visualize the presence and distribution of missing values in your dataset. This can help determine the extent of missingness and guide imputation or deletion strategies.

2. **Outliers**: Identify outliers in your data through visualizations. These extreme values may impact your analysis and can be handled either by correcting errors or applying appropriate transformations.

3. **Distributions**: Visualize the distributions of variables to assess their shapes, central tendency, and spread. This can aid in identifying skewness, multimodality, or potential issues with data quality.

4. **Relationships**: Explore the relationships between variables using scatter plots, heatmaps, or correlation matrices. This can reveal patterns, associations, or dependencies that can assist in feature engineering or identifying redundant variables.

Practical Data Visualization (Cont.)

5. **Time Series Patterns**: If dealing with temporal data, visualize time series patterns to inspect trends, seasonality, or anomalies. This can guide feature engineering by capturing time-related information.

6. **Class Imbalance**: If working with classification problems, visualize the distribution of target classes to assess class imbalance. This can help determine if resampling techniques like oversampling or undersampling are necessary.

7. **Feature Importance**: Use visualization techniques (e.g., bar plots, heatmaps, or trees) to evaluate the importance of features or variables. This can guide feature selection or extraction strategies.

> In summary, we're looking for hints for any trends in our data that might help us out during model creation.

Practical Data Visualization (Cont.)

For data visualization, we'll be using seaborn and matplotlib.

> Both matplotlib and seaborn have very good documentation.

What insights have we learned from this?

Now that we've gained insight, can we use this in a machine learning model **as-is**? If so, what model would be the best fit?

Data Preprocessing = Necessity

While in theory, our data is always ready to be used in a machine learning model and needs little to none preprocessing (unless the topic of study is feature engineering, such as PCA), in practice, we almost *always* need to do some data preprocessing.

> Even if we did not run into explicit errors when feeding the data to the model, we still might not have been able to obtain good results. *(Why?)*

“Why can’t we just make a machine learning model complex enough to make up for these short comings?” (Why not?)

Data Cleaning

An important concept here is data *cleanliness*.

Data cleanliness refers to the quality and integrity of the data used in data science and machine learning projects. It implies that the data is accurate, complete, consistent, and free from errors, anomalies, duplicates, and irrelevant information.

- **Accurate insights**: Clean data ensures that the models and algorithms can generate reliable and accurate insights and predictions. If the data is incorrect or contains errors, the results of analysis or predictions can be misleading or inaccurate.
- **Model performance**: The quality of the input data greatly impacts the performance of machine learning models. Clean, well-prepared data helps in training robust models that can generalize well to unseen data, leading to better predictive power and model performance.
- **Trust and credibility**: Clean data enhances the trust and credibility in the results and models generated by data scientists.
- **Time and cost efficiency**: Working with clean data reduces the time and effort required for data cleaning and preprocessing.

> Cleaning up data is normally a part of data preprocessing. It never hurts to check if our data is clean.

Data Preprocessing

To clean our data and make it fit for feeding to a machine learning model, we move onto the next step: data preprocessing.

Usually, data proprocessing consists of the following steps:

1. **Data Cleaning**: This involves handling missing values, removing duplicates, and dealing with outliers in the data.
2. **Data Transformation**: This step involves transforming the data into a suitable format for modeling. This may include scaling or normalizing numerical features, encoding categorical features, or converting text or images into numerical representations.
3. **Feature Selection**: In this step, features that are irrelevant or redundant for the model can be removed to improve efficiency and performance.
4. **Feature Engineering**: This step involves creating new derived features from the existing ones to capture more meaningful information for the model.

Data Preprocessing (cont.)

5. **Splitting Data**: The data is usually split into training and testing sets. The training set is used for training the model, while the testing set is used for evaluating its performance.
6. **Handling Imbalanced Data**: If the data is imbalanced, meaning that one class is significantly more prevalent than the others, techniques like oversampling, undersampling, or synthetic data generation can be applied to balance the data.
7. **Data Normalization**: This step involves scaling the features to have a consistent range to ensure that no feature dominates the model training process.

Back to Jupyter notebook

So does EDA come first or data cleaning?

It depends (on what?)

> Hint: Data cleaning is a type of EDA.

So now that data is clean, we can finally feed it to the model.

(Back to code)