

Advanced Python and Machine Learning

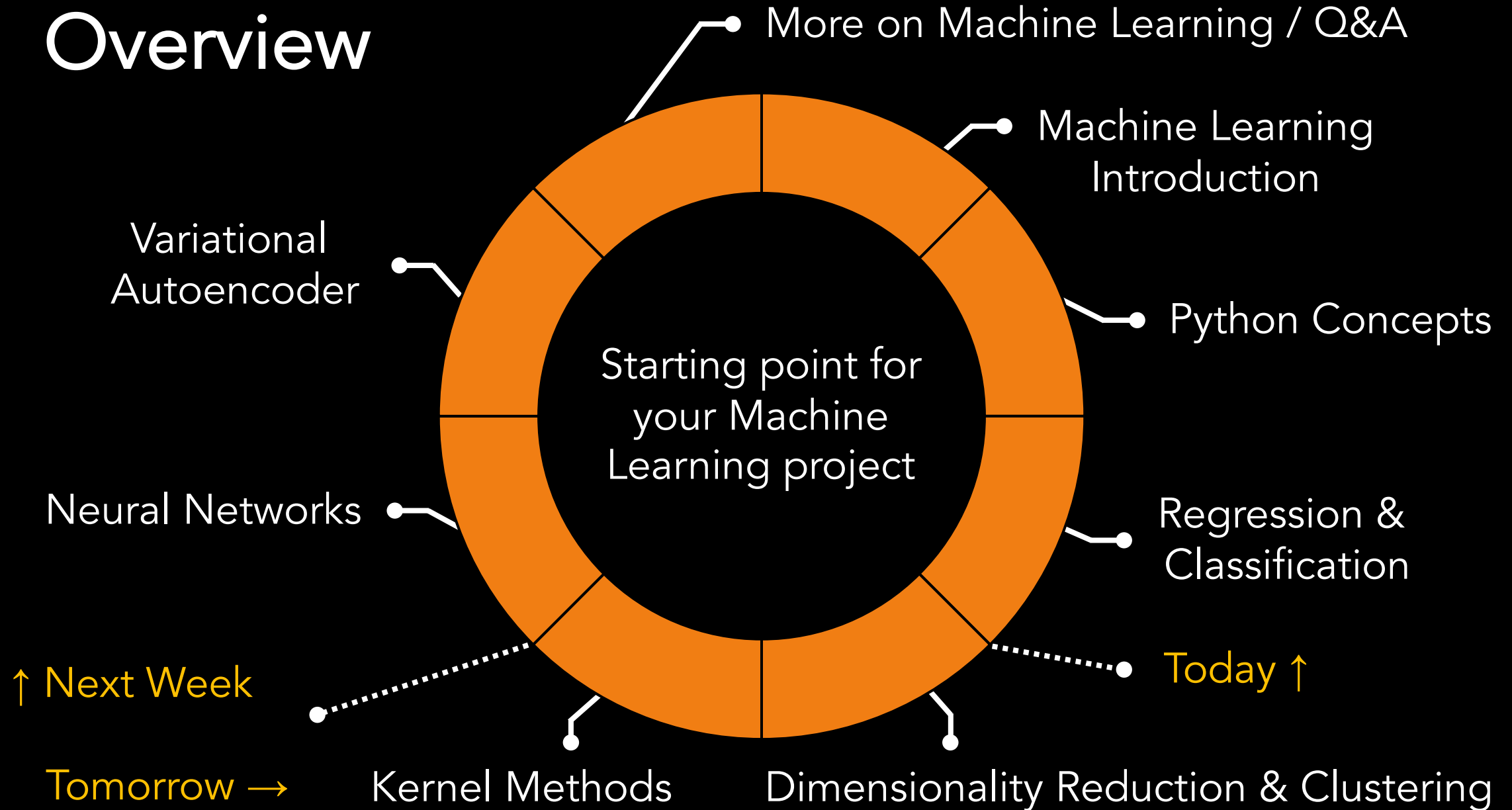
Dr. Maxim Samarin

Vitali Nesterov

16th / 17th / 23rd / 24th March 2023

Please:  +  for class;  +  for questions

Overview





- B.Sc. and M.Sc. in Physics, Ph.D. in Machine Learning
- Research in Generative AI, Computer Vision, Deep Learning theory and various applications
- Lecturing Python classes



- B.Sc. and M.Sc. in Computer Science
- Research in Deep Learning and Generative AI in Physics applications
- Quantitative Finance at BKB



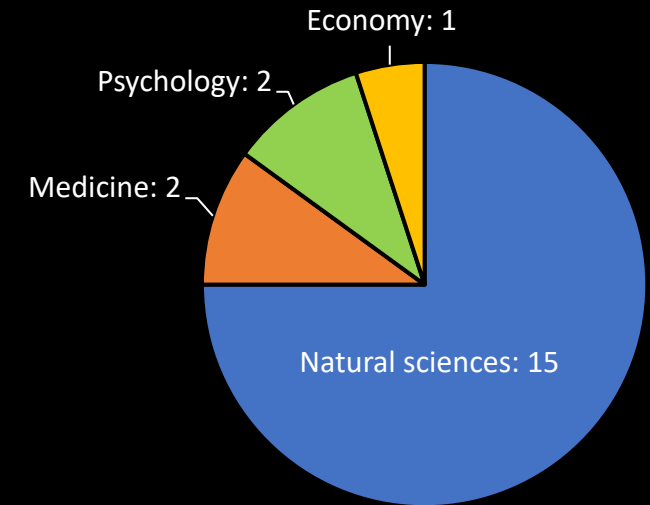
- B.Sc. and M.Sc. in Physics, Ph.D. in Machine Learning
- Research in Generative AI, Computer Vision, Deep Learning theory and various applications
- Lecturing Python classes



- B.Sc. and M.Sc. in Computer Science
- Research in Deep Learning and Generative AI in Physics applications
- Quantitative Finance at BKB

Your Experience and Goal

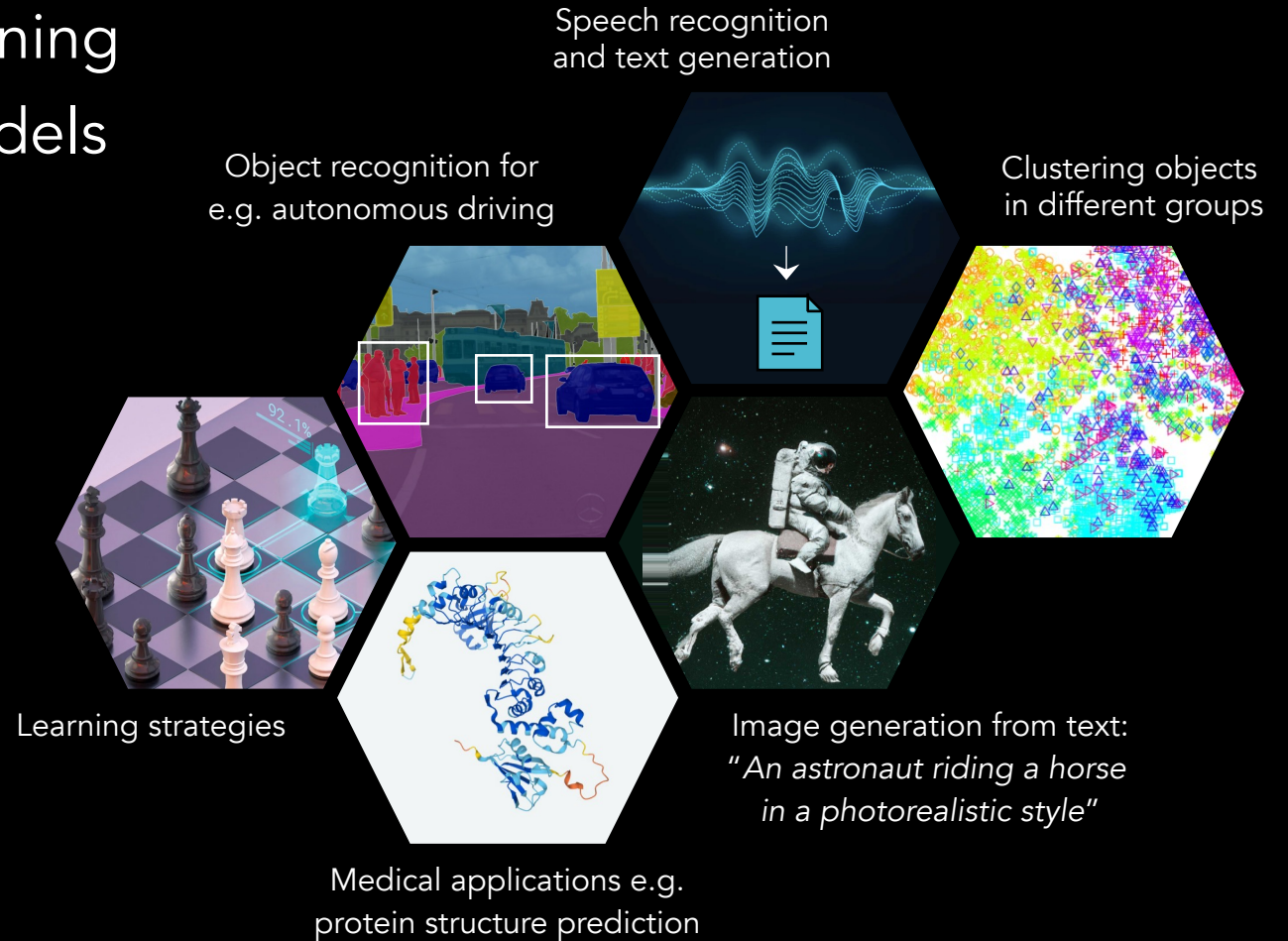
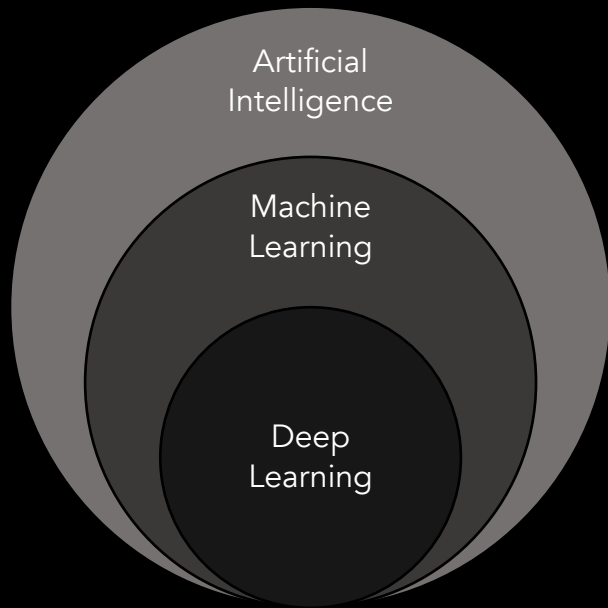
- What is your **programming experience** so far and an **intermediate goal** for you?



Alexander → Gina → Maria → Caner → Pius →
→ Frederick → Lukas → Yunrui → Sebastian L. → Claire →
→ Patrick → Felix → Fabio → Maria → Leonidas-Dimitrios →
→ Jana → Dietger → Larissa → Sebastian Z. → Han

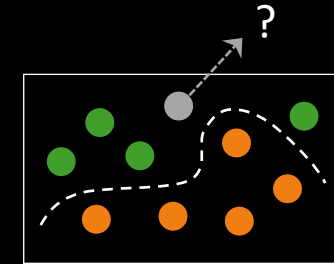
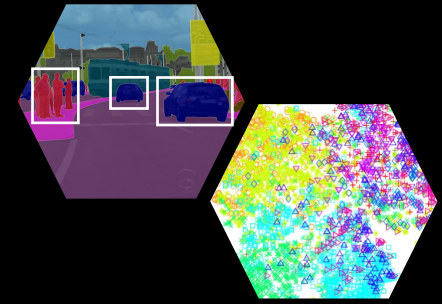
Machine Learning Landscape

- Supervised and unsupervised learning
- Discriminative and generative models
- Reinforcement learning

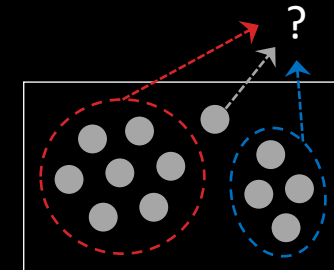


Supervised vs. Unsupervised

- **Supervised learning**: Labeled data guides learning
 - **Classification**: Class labels separate groups of data
 - **Regression**: Functional relationship between inputs (predictors) x and responses $y \rightarrow f(x) = y$
- **Unsupervised learning**: No labels available
 - **Clustering**: Identifying clusters inherent to the data
 - **Dimensionality reduction**: Identifying relevant predictors / dimensions or factors of variation in data

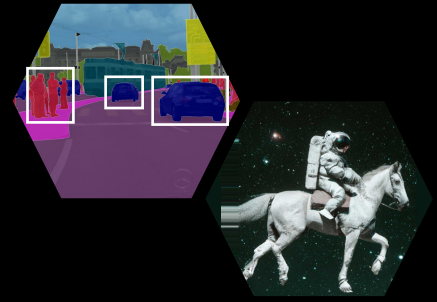


Supervised

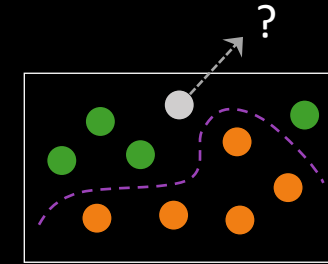


Unsupervised

Discriminative vs. Generative

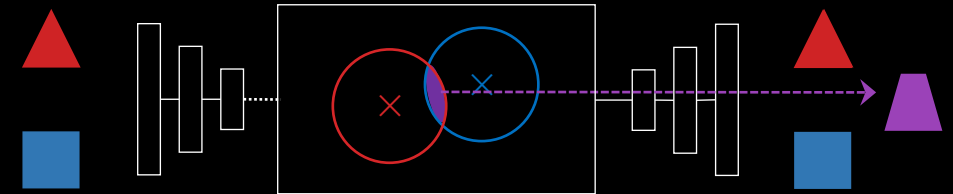


- **Discriminative models:** Learn **decision boundary** to differentiate data and provide accurate predictions to new, unseen data



Discriminative

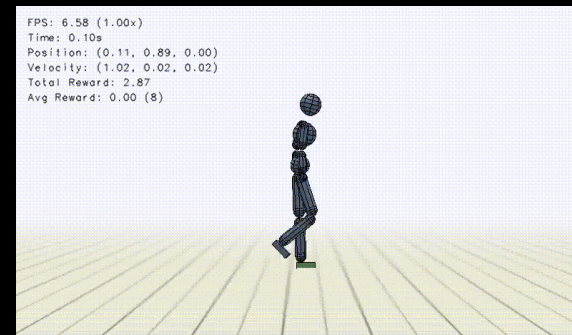
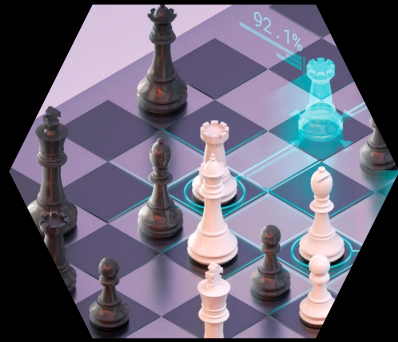
- **Generative models:** Approximate data distribution to provide accurate predictions but also allow **generating artificial data** similar to training data



Generative




Reinforcement Learning

- Agents explore environment, select actions, receive rewards, and form strategies / policies → **exploration** and **exploitation**
- Modelling (Markov) decision processes
- Challenging in (large) uncertain environments
- Application in games, robotic, and finance

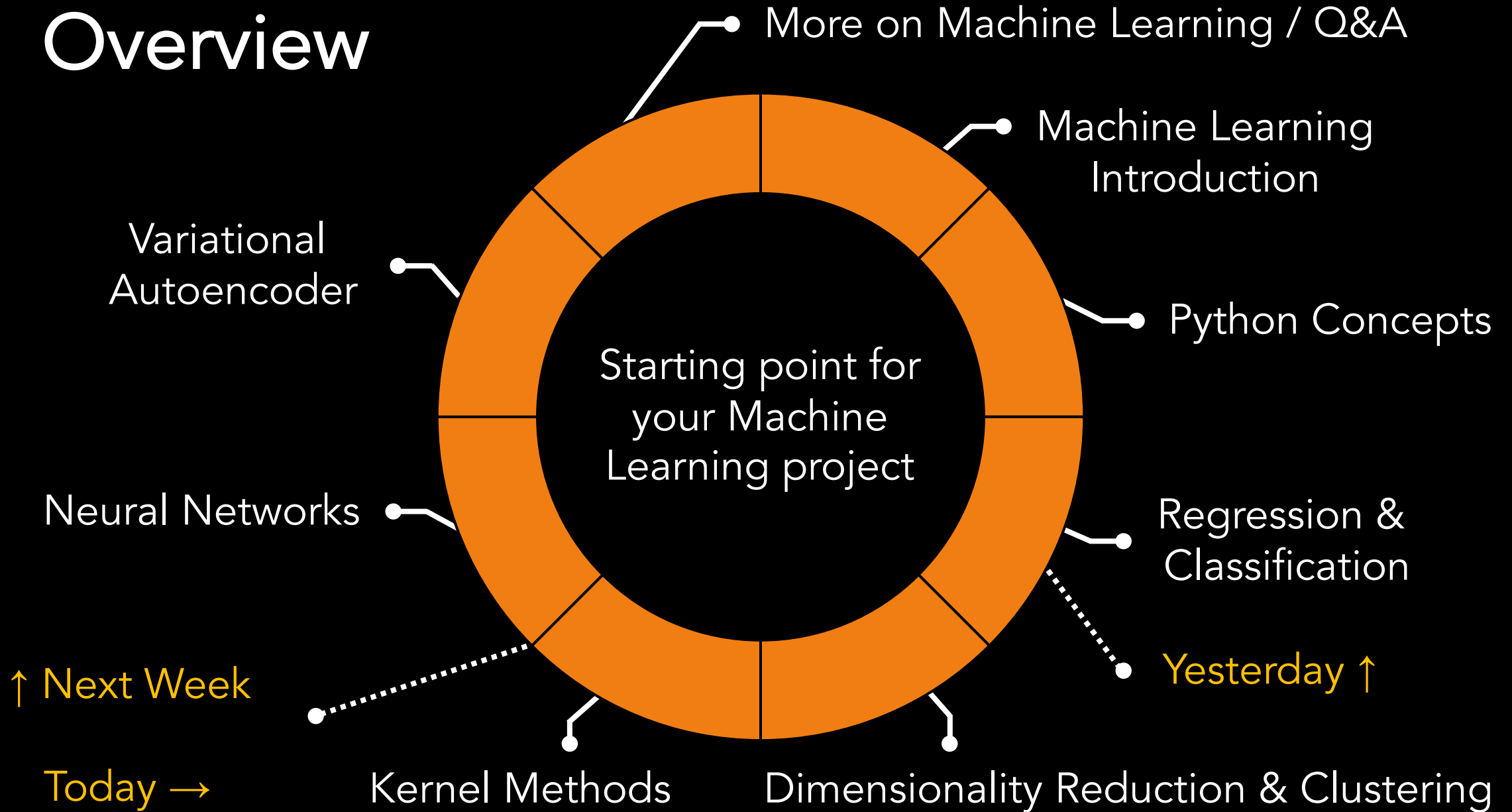


Source: [G. Berseth and C. Pal](#)

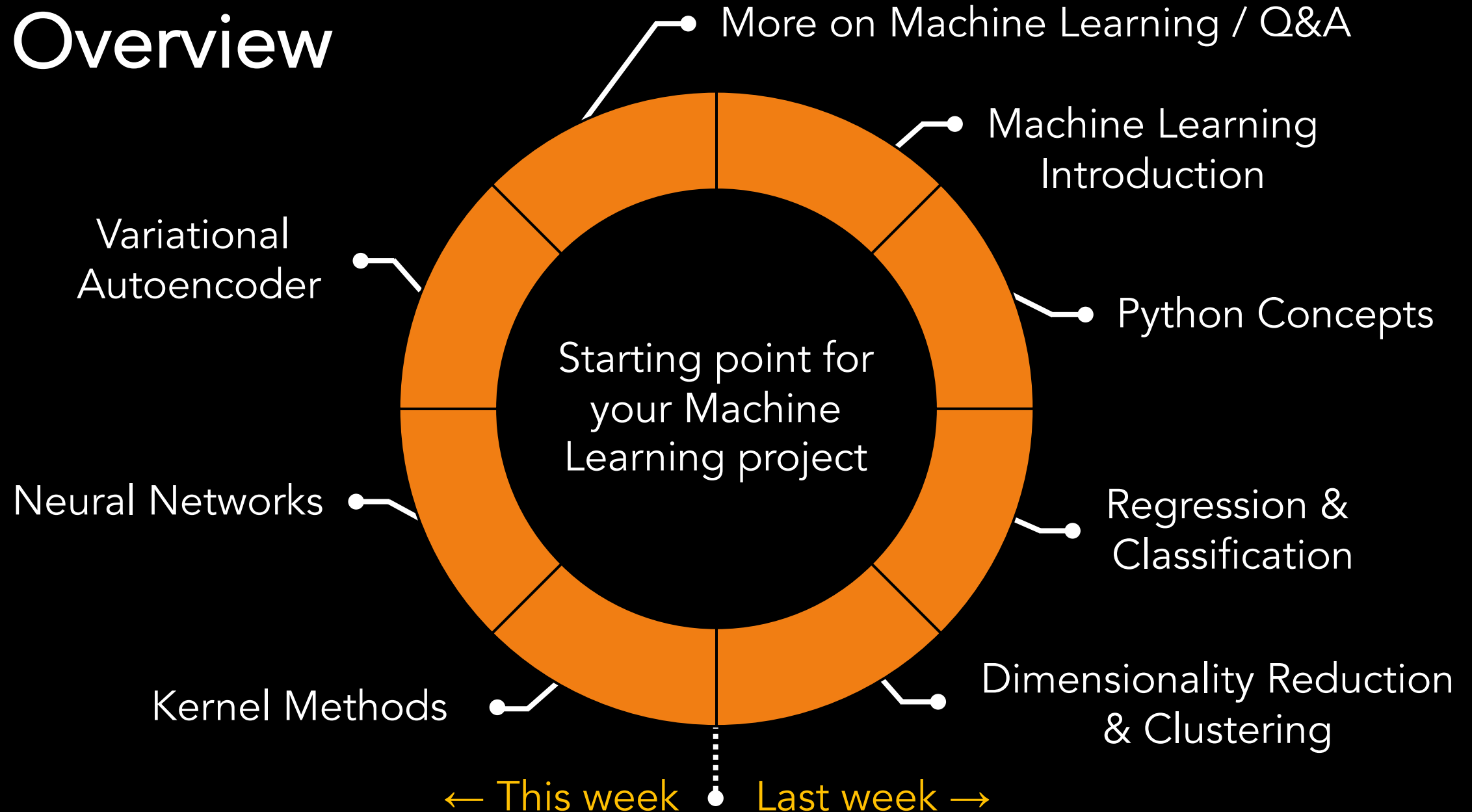
Start your Working Environment

1. Access the course environment via **Noto link** provided in E-Mail and sign in with your SWITCH edu-ID ("**Use your Switch AAI login**")
2. **Or:** Download new material, start Jupyter Lab and open notebooks
 - Suggestions on how to work in this course:
 -  +  : Follow presentation, while executing scripts yourself, making adjustments and notes in your own notebook
 -  : Follow presentation, switch to programming environment for exercises

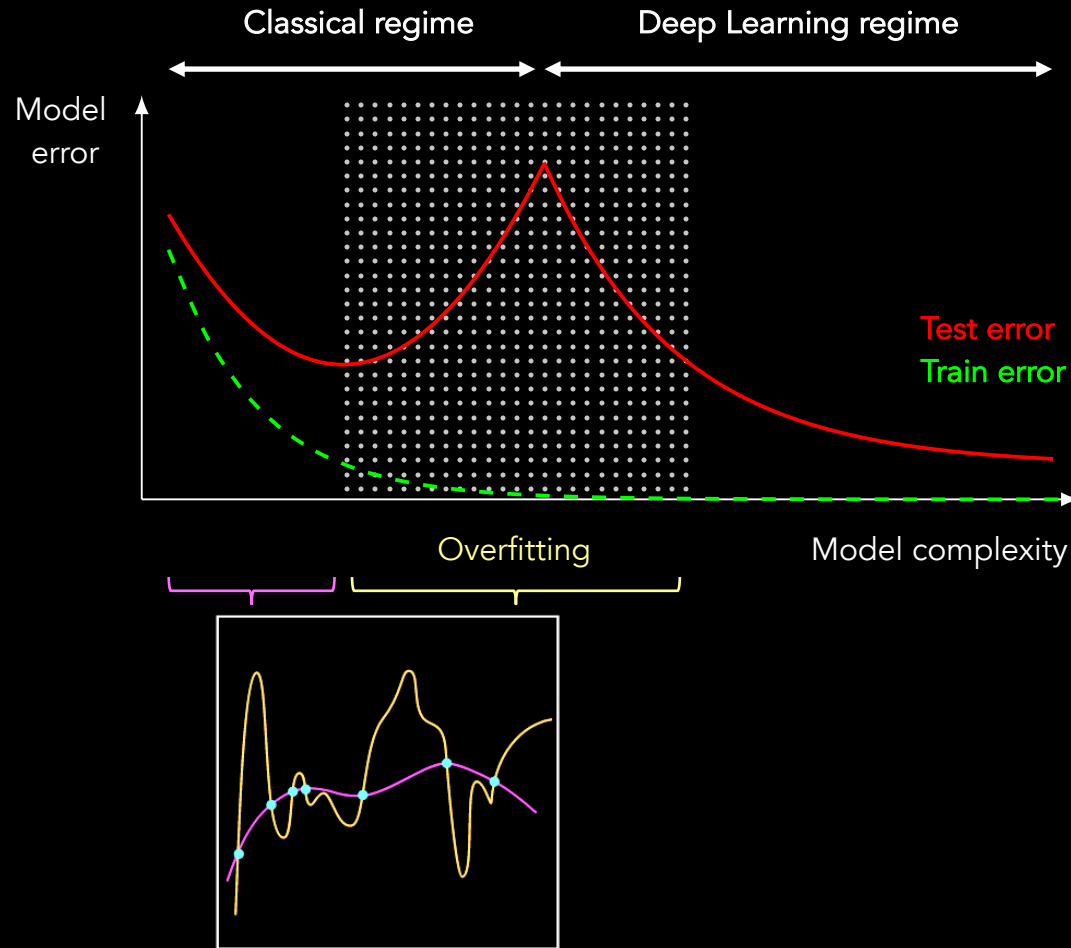
Overview



Overview



Bias-Variance Trade-off



- Flexible models are prone to overfitting
- **Training / validation / test** splits to ensure generalisation to new data
- Watch out for imbalance in training data

Machine Learning Topics

Regression & Classification	Dim. Reduction & Clustering	Kernel Methods	Neural Networks
Linear / Ridge / Logistic Regression	Principal Component Analysis	Kernel Ridge Regression	Feed-Forward Neural Networks
Decision Trees & Random Forests	k-Means	Gaussian Process	Convolutional Neural Networks
k-Nearest Neighbors	Gaussian Mixture Models	Kernel Density Estimation	Variational Autoencoder
Support Vector Machines	Factor Analysis	Kernel PCA	Transformer
Naïve Bayes	Independent Component Analysis	Kernel SVM	Generative Adversarial Networks
...	...	Kernel

How to Continue with ML & Python

- Machine Learning **scikit-learn** tutorials
<https://scikit-learn.org/stable/tutorial/index.html>
- **TensorFlow** tutorials
<https://www.tensorflow.org/resources/learn-ml>
- **Quick overview** with interactive tutorials on some basic topics and more advanced concepts:
<https://www.learnpython.org/>
- **Great tutorials** on specific (advanced) topics, easy-to-read books:
<https://www.realpython.com/>

How to Continue Programming

- Use an integrated development environment (IDE) like
 - **PyCharm**:
<https://www.jetbrains.com/pycharm/>
 - **Visual Studio Code**:
<https://code.visualstudio.com/>
- Use high-performance computing (GPUs, TBs of RAM) cluster **sciCORE**
 - **Getting started**:
<https://scicore.unibas.ch/using-scicore/getting-started/>
 - **Courses**:
<https://scicore.unibas.ch/events/teaching-training/>

Suggestions for the Feedback



- **Preparation information** / YouTube videos adequate?
- What did / didn't you like about **Noto**?
- Content appropriate: **anything missing** for you (Python and Machine Learning)?
- Too fast or slow, shallow or deep?
- **Insightful exercises** and enough time?

Thank you and good luck on your Machine Learning journey! 😊