## **Machine Learning Capstone Project**

In this capstone project, you will pick a problem of your choice that can be solved by applying machine learning. Be prepared to leverage algorithms and techniques you've learned throughout the Nanodegree program.

### **Define your problem**

Think about a problem domain that you are passionate about, e.g. healthcare, engineering, finance, robotics, marketing, bio-informatics, etc., something that excites you. Then choose an existing problem within that domain which you could solve by applying machine learning.

Look for challenges and datasets on platforms like [Kaggle](https://www.kaggle.com/), [Devpost](http://devpost.com/), etc. If you do find a problem you'd like to solve on one of these platforms, make sure to cite the source in your project report.

Once you have picked a problem, define it in clearly your report.

* *What is the problem you're trying to solve, why is it important to solve it?*
* *How does it affect people? How does it affect you?*

### **Describe a solution**

In order to solve your problem, you should know what a solution to it looks like. Try to describe such a solution as precisely as you can. For instance, if it is a prediction problem, how accurate do your results need to be? How quickly does your algorithm need to produce an answer?

Pick a suitable set of metrics to measure how well your approach works. For example, in a regression problem, you may choose an error metric such as mean squared error; in case of a clustering problem, you may measure the goodness of a solution in terms of intra-cluster and inter-cluster distances.

*In your report, explain briefly why you think your chosen metric(s) is/are applicable.*

Don't worry if you don't know enough to address this at the beginning - as you proceed with the project, you will be able to refine your expectations and estimate better. But starting with a goal in mind is infinitely better than starting with none!

### **Analyze the problem**

As first step towards solving your problem, you need to understand it better. If you have a dataset to work with, apply Exploratory Data Analysis techniques to gain some insight.

* *What is the size of your dataset?*
* *How many features are present?*

Compute some basic statistics on the different data elements, e.g. mean, variance, min/max.

* *Which features seem most promising?*
* *Are there any categorical variables that may need to be converted?*

Now try to think which machine learning algorithms would be most applicable for solving the problem.

* *Which ones would you expect to perform well?*
* *How easy is it to convert the available data into a suitable form?*

### **Implement a solution**

Now that you understand the problem better, you are ready to apply your chosen machine learning technique(s) and come up with a solution.

Create your solution by using an existing library (such as scikit-learn) or rolling your own implementation (code must be submitted using Python 2.7). Keep in mind that for a real-world challenge, an existing implementation may not work nicely out of the box. You will likely have to tweak parameters, and transform the input to get better results.

* *What pre-processing operations do you have to carry out on the features? (e.g. scaling, normalization, selection, transformation)*
* *Are there any incomplete data points or outliers that you have to work around?*

Once you have a first-pass solution, try to gauge how well it performs.

* *Using the metric(s) you defined earlier, measure your current performance. Is it close to what you expected?*
* *Are there any better metrics you can come up with?*

Feel free to revise your problem and solution descriptions according to what you have learned so far.

### **Refine your solution**

Iterate on your solution to make it as good as you can.

* *For each version of your solution, track what changes you make and how they affect performance.*
* *Does it ever become worse? If so, note down and figure out why.*

Your final solution may or may not be exactly like you initially planned. But that is okay!

* *Report how your project evolved, and what changes you made to your specifications (if any).*
* *What was your experience like working on this project? Do you feel more confident taking on open-ended projects like these in the future?*

### **Complete your report**

Make sure you have addressed all the rubric components. You can use the questions in *italics* as guidelines.

Turn in your project code and a PDF report (3-5 pages), together as a .zip file or as a link to an online repository.

### **Evaluation**

Your project will be reviewed by a Udacity reviewer against [**this rubric**](https://docs.google.com/document/d/1fC82dy85yskqNeMbB1Xz4wsOZGW--BIZXS1IWhJvrGM/pub). Be sure to review it thoroughly before you submit. All criteria must "meet specifications" in order to pass.

### **Submission**

When you're ready to submit your project go back to your [Udacity Home](https://www.udacity.com/me), click on Project 5, and we'll walk you through the rest of the submission process.

If you are having any problems submitting your project or wish to check on the status of your submission, please email us at **machine-support@udacity.com** or visit us in the [discussion forums](http://discussions.udacity.com/).

### **What's Next?**

You will get an email as soon as your reviewer has feedback for you. In the meantime, review your next project and feel free to get started on it or the courses supporting it!

## **Capstone Project Suggestions**

If you haven't identified your own problem yet, we have compiled several application areas and/or project suggestions to inspire you. Links to related courses are also included:

* [Deep Learning](https://docs.google.com/document/d/1L11EjK0uObqjaBhHNcVPxeyIripGHSUaoEWGypuuVtk/pub)
* Artificial Intelligence for Robotics
  + [Plotting and Navigating a Virtual Maze](https://docs.google.com/document/d/1ZFCH6jS3A5At7_v5IUM5OpAXJYiutFuSIjTzV_E-vdE/pub)
* [Healthcare](https://docs.google.com/document/d/1WzurKKa9AX2DnOH7KiB38mvozdOSemfkGpex8hdTy8c/pub)
* [Education](https://docs.google.com/document/d/1vjerjRQnWs1kLbZagDYT6rNqiwAG23Yj45oUY88IAxI/pub)
* [Computer Vision](https://docs.google.com/document/d/1y-XfjkPFgUQxFIQ9bBncUSjs4HOf5E-45FrLYNBsZb4/pub)
* [Investment and Trading](https://docs.google.com/document/d/1ycGeb1QYKATG6jvz74SAMqxrlek9Ed4RYrzWNhWS-0Q/pub)