- There are a number of machine learning specific sources of risk and dangerous design patterns.
- Technical debt is a metaphor for fiscal debt where decisions made to improve the system in the present become greater hindrances to the system in the future.
- Machine Learning code have all the complexity issues of normal code where debt can be incurred but also present a higher level complexity in the way that they are coupled closely with other systems.
- Example risk factors are re-using of input signals which cause tight coupling between disjoint systems, creating large quantities of glue code to adapt the machine learning aspect to its related systems, changes in these related systems that have knock on effects to the machine learning model, etc.
- The software engineering paradigms of boundaries and encapsulation are difficult to apply to machine learning because of their dependence on external data
- Machine Learning suffers from problems of entanglement where the mixing of data makes the isolating of factors impossible
- One strategy is decomposing the problem into sub problems and modelling them individually, however this is not scalable.
- Another strategy is to use visualisation tools to gain a deeper understanding of the behaviour of the model and identify the effects of altering parameters across dimensions.
- A third strategy involves complex regularization methods on hyperparameters, but this may introduce more debt via system complexity.
- Hidden feedback loops are a concern where the model learns from data it is influencing over an extended period of time causing gradual changes in the system that are difficult to analyze.
- Undeclared consumers are a concern where other parts of the system utilise data from the prediction model without proper consideration for how changes in the model will then affect them.
- Data input to the model from other systems and data output from the model to other systems cause data dependencies that must be well thought out and managed lest a chain is created between what should be independent systems.
- Experimental code and alternative configurations should be evaluated and implemented within the system only if they prove useful.
- Thresholds for decisions should be updated using validation data
- Care should be taken to identify when effects correlate but are not causal.
- Models should be monitored that the distributions they produce resemble the distributions of your validation data and that there are limits set on exceptional data

http://www.eecs.tufts.edu/~dsculley/papers/technical-debt.pdf