Machine Learning Checklist

#	Task	?
1	**Define the problem**	
1.1	**What is the problem?**	
1.1.1	Define the problem informally	□
1.1.2	Define the problem formally	□
1.1.3	List the assumptions about the problem	□
1.1.4	List problems that are similar	□
1.2	**Why does the problem need to be solved?**	
1.2.1	Describe the motivation for solving the problem	□
1.2.2	Describe the benefits of the solution (model predictions)	□
1.2.3	Describe how the solution will be used	□
1.3	**How could the problem be solved manually?**	
1.3.1	Describe how the problem is currently solved (if at all)	□
1.3.2	Describe how a subject matter expert would make manual predictions	□
1.3.3	Describe how a programmer might hand code a solution	□
2	**Prepare The Data**	
2.1	**Data Description**	
2.1.1	Describe the extent of the data that is available	□
2.1.2	Describe data that is not available but is desirable	□
2.1.3	Describe the data that is available that you don't need	□
2.2	**Data Processing**	
2.2.1	Format data so that it is in a form that you can work with	□
2.2.2	Clean the data so that it is uniform and consistent	□
	* Impute missing values	
	* Identify and remove outliers	
2.2.3	Sample the data in order to best trade-off redundancy and fidelity	□
	* Sample instances	
	** Randomly sample	
	** Rebalance classes	
	* Sample attributes	
	** Randomly sample	
	** Remove highly-correlated attributes	
	** Apply dimensionality reduction	
2.3	**Data Transformation**	
2.3.1	Create linear and nonlinear transformations of all attributes	□
	* Square	
	* Square Root	
	* Standardize	
	* Normalize	
	* Discretize	
2.3.2	Decompose complex attributes into their constituent parts	□

#	Task	?
	* Decompose date-times into components	
	* Decompose categorical into binary attributes	
2.3.3	Aggregate denormalized attributes into higher-order quantities	□
	* Roll-up events by entity into aggregate values, if relevant (min, max, count, avg)	
2.5	**Data Summarization**	
	Create univariate plots of each attribute	□
	Create bivariate plots of all pairwise combinations of attributes	□
	Create bivariate plots of each attribute with the output attribute	□
3	**Spot Check Algorithms**	
3.1	**Create a Test Harness**	
	Create a hold-out validation dataset for later use	□
	Evaluate and select an appropriate test option	□
	* Train and test sets	
	* k-fold cross validation	
	Select a performance measure used to evaluate models	□
3.2	**Evaluate Candidate Algorithms**	
	Select a diverse set of algorithms to evaluate (10-20)	□
	* k-nearest neighbors	
	* learning vector quantization	
	* naive bayes	
	* logistic regression	
	* linear discriminant analysis	
	* CART	
	* C4.5/5.0	
	* Backpropagation	
	* Support Vector Machines	
	* Random Forest	
	* Gradient Boosted Machines	
	Use common or standard algorithm parameter configurations	□
	* From literature	
	* From winning competition entries	
	Evaluate each algorithm on each prepared view of the data	□
	* i algorithm+configs by j data views	
4	**Improve Results**	
4.1	**Algorithm Tuning**	
	Use historically effective model parameters	□
	Search the space of model parameters	□
	Optimize well performing model parameters	□
4.2	**Ensemble Methods**	
	Use bagging on well performing models	□

#	Task	?
	Use Boosting on well performing models	□
	Blend the results of well performing models	□
4.3	**Model Selection**	
	Select a diverse subset of well performing models (5-10)	□
	Evaluate well performing models on a hold out validation dataset	□
	Select a small pool of well performing models (1-3)	□
5	**Finalize Project**	
5.1	**Present Results**	
	Write up the project in a short report (1-5 pages)	□
	Convert write-up to a slide deck to share findings with others	□
	Share code and results with interested parties	□
5.2	**Operationalize Results**	
	Adapt the discovered procedure from raw data to results to an operational setting	
	Deliver and make use of the predictions (if intended)	□
	Deliver and make use of the predictive model (if intended)	□