Things I learned with the Allstate Competition:

1. Preprocessing data:
   1. If the data is very skewed, you can log transform the data and add some number to make the distribution normal – you can do this even if it’s just one feature – which in this case is the target – loss
      1. Log(y+n)
      2. Log(y)
      3. Sqrt(y)
      4. Boxcox(y)
      5. Y => [0,1]for logistic regression
   2. For the categorical variables, you can:
      1. Factorize -by this you sort the data before labelencoding them which takes into account the order like A,B,C is 1,2,3
      2. Normal label encode of the variables which does not sort them
      3. Dummy variables or one hot encoded variables
      4. Lexical encoding is when A is 1 and AA is 27 because if you just do factorize then A becomes 1 and AA becomes 2 – seems to be good for tree based models
      5. TFidf encoding
      6. Removal of rare values
      7. Entity embedding – enables to express and learn the complex relations of different categories in a multi-dimensional vector space
         1. <https://github.com/entron/entity-embedding-rossmann>
         2. https://arxiv.org/abs/1604.06737
   3. For the numerical variables
      1. Box cox which can be used to make the data normal and by identifying a lambda and using that lambda to raise all the values to that power – this method is not a guarantee for normality. Box cox can improve linearity
      2. Minmaxscaler – transforms features by scaling each feature to a given range
      3. Standard scaler – 0 mean unit variance
      4. Quantile-based binning
      5. Percentile transformation
      6. Count of values
2. Feature Engineering/Interactions:
   1. 2,3, or even up to 4-way interactions between features
   2. sklearn.metrics.mutual\_info\_score – this is a measure of the similarity between two labels of the same data – take the ones with high MI
      1. <http://scikit-learn.org/stable/modules/generated/sklearn.metrics.mutual_info_score.html>
   3. Statistical features (not so useful)
   4. Select interactions with xgbfi
      1. <https://github.com/Far0n/xgbfi>
      2. <http://projects.rajivshah.com/blog/2016/08/01/xgbfi/>
      3. https://github.com/limexp/xgbfir
3. Feature Selection
   1. PCA – principal component analysis
   2. SVD – singular value decomposition
      1. Not used to normalize data but to get rid of redundant data
4. Models used:
   1. XGBoost
   2. Keras
   3. LightGBM
   4. GBR
   5. Regularized Greedy Forest
   6. Vowpal Wabbit
   7. LibFM
   8. LibFFM
   9. Linear Regression
   10. Support Vector Regression
   11. KNN
5. Tuning
   1. Bayesian Hyperoptimization - <https://github.com/fmfn/BayesianOptimization/>
   2. GridSearch
   3. Randomized GridSearch
6. Stacking/Blending
   1. Scipy Minimize package (BFGS, SLSQP)
   2. Markov Chain Monte Carlo (MCMC)
   3. Quantile Regression -<http://statsmodels.sourceforge.net/devel/examples/notebooks/generated/quantile_regression.html>
   4. When stacking you can predict the test data for each of the folds and then average it or you can retrain the entire model and predict the test data