# Data Challenge

Tejas Kulkarni

*tejasvijaykulkarni@gmail.com*

February 7, 2022

# Problem statement.

1. Due to the huge variance in treatment response, characterizing a recently diagnosed patient into high/low risk group can help allocate resources efficiently.
2. **High risk:** Very likely that patient will not respond to treatment, or relapse quickly.
3. **Low risk:** Not high risk, cancer is not likely worsen quickly.
4. Can be treated as a binary classification problem.

# First thoughts.

1. Merging clinical notes and gene expression data leads to a high dimensional data set ($583 \times 24172$).
2. Classification with lower *false negative* rate is more important than just accuracy related metrics.
3. Providing some measure of *uncertainty quantification* is also crucial.
4. Knowing the most important top-k features will be useful.

# Exploration.

1. There are no high risk patients with D_OS or D_PFS values $\geq 18$, or low risk patients with values $< 18$.
2. It seems that these labels completely determine the patient risk class, and all patients with CENSORED flag are high risk patients.
3. A small number of patients have disease stage (D_ISS) as nan.
4. Several gene ids have zero-rows for all patients.
5. Fortunately, we have enough training examples for both classes, hence no class imbalance.

# Essential pre-processing.

1. Gene expression file.
   1. Indexed the file with *Entrez id*.
   2. Deleted the gene expression records with zero-rows for all patients.
   3. Applied min-max scaling to deal with varying scales across gene ids.
2. Clinical annotation file.
   1. Deleted features with no information content (e.g. same value through out the column).
   2. Replacing rare nans in column D_ISS with 0 (not sure about the implications though).
   3. Converted days in the columns D_OS and D_PFS to months.
   4. Removed one feature from the most correlated feature pairs.
   5. Reduced the number of labels from 3 to 2 using the hint.
   6. **Removed D_OS and D_PFS to avoid model leakage.**
   7. Applied min-max scaling to deal with varying scales across columns.

# Model choices.

1. We preferred simpler models due to familiarity/ scalability/interpretability reasons.
2. We treat the classifier probabilities as a measure of uncertainty.
3. Models used.
    1. Logistic regression (easily scalable to high dimensional data but sensitive to outliers.)
    2. Support vector machines with RBF kernel (more robust to outliers).
    3. Ensamble decision tree with bagging (reducing model variance with data set bootstrapping).
    4. Random forest (additional randomness with feature sub-sampling).
    5. Multi-layer perceptron with Relu activation (universal approximators).
    6. K-nearest neighbors (easily overfits for higher dimensionality.)

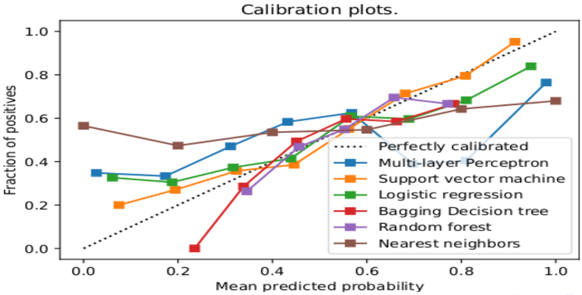
# Model performance.

1. We decided to retain 90% of the variance in the original data set, and used PCA to reduce the dimensionality.
2. The dimensionality is reduced from 23119 to 266.
3. Average model performances after **stratified 10-fold cross-validation** are presented below.
4. We weigh classifiers by avg. recall (fraction of true high risk patients correctly classified high risk), and AUC scores.
5. Poor performance of ensamble classifiers, 5-NN, and MLP could be due to bad hyper-parameters, curse of dimensionality.

| Metric<br>Classifier | test accuracy | test precision | test recall | f1 | auc |
|---|---|---|---|---|---|
| Multi-layer perceptron | 0.65 ± 0.07 | 0.69 ± 0.06 | 0.69 ± 0.09 | 0.69 ± 0.06 | 0.65 ± 0.07 |
| **Support vector machine** | **0.69 ± 0.06** | **0.72 ± 0.06** | **0.72 ± 0.08** | **0.72 ± 0.06** | **0.68 ± 0.06** |
| **Logistic regression** | **0.68 ± 0.05** | **0.71 ± 0.06** | **0.7 ± 0.06** | **0.71 ± 0.04** | **0.67 ± 0.06** |
| Decision trees with bagging | 0.58 ± 0.05 | 0.6 ± 0.03 | 0.74 ± 0.07 | 0.66 ± 0.04 | 0.56 ± 0.05 |
| Random forest | 0.58 ± 0.08 | 0.59 ± 0.05 | 0.8 ± 0.1 | 0.68 ± 0.06 | 0.56 ± 0.08 |
| 5-Nearest neighbors | 0.53 ± 0.05 | 0.59 ± 0.06 | 0.51 ± 0.06 | 0.55 ± 0.06 | 0.54 ± 0.05 |

# Uncertainty measure.

1. For stratified 10-fold cross-validation, we split and average classifier prediction probabilities on test data into 8 bins.
2. For each average predicted probability bin on the x-axis, we plot the fraction of positively predicted test points on the y-axis.
3. Ideally, we want classifiers to predict higher number of positives at higher probabilities than at lower probabilities, and vice versa.
4. SVM and LR once again stand out as better calibrated models.



# Differentially private prediction.

1. In a very preliminary exploration using IBM's *diffprivlib* library, we tried to fit the logistic regression for several $\epsilon$ values.
2. Surprisingly, performance was bad, nearly invariant of $\epsilon$, and $L_2$ norm of the each input vector.
3. It seems the library uses an old objective function perturbation method [1].
4. Several studies including [2] confirm that summary statistics and gradient perturbation based methods are more accurate for linear models.

# Next steps for further explorations.

1. Understand if we can treat this problem as a survival analysis task.
2. Understand more about data/domain to make more educated pre-processing/modeling decisions.
3. Measure performance without dimensionality reduction (currently prohibitively time consuming on our machine), and estimate the top-k most important features.
4. Try to check the possibility of improving on the metrics.
5. Perform hyper-parameter tuning.
6. Implement DP summary statistics or a gradient based (e.g. DP-SGD) methods for binary classification.
7. Simulate this study in a federated environment.

# Bibliography.

K. Chaudhuri, C. Monteleoni, and A. D. Sarwate.
Differentially private empirical risk minimization.
*Journal of Machine Learning Research*, 2011.

Y. Wang.
Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain.
In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018*.