Vibhav Peri

CSE 428 - Computational Biology Capstone

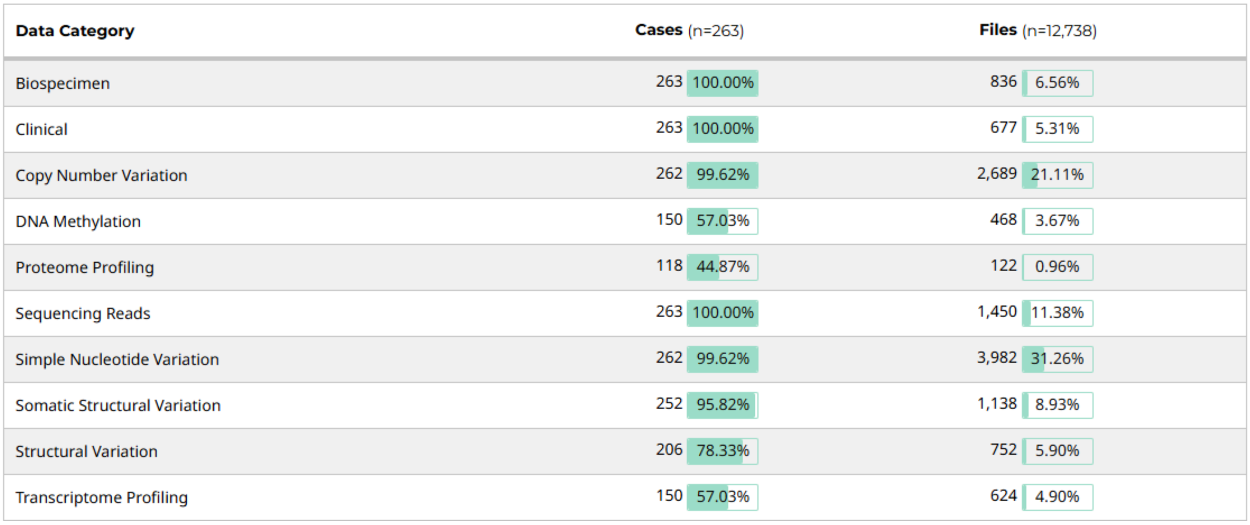
Final Report

Problem:

My research this quarter focused on Testicular Cancer. I chose this topic since I have a very personal experience with this cancer and lots of inherent background knowledge for that reason. Based on the time constraints and the publicly available data online, I chose to do binary classification on whether a patient's tumor is Seminoma or Non-Seminoma using gene expression data.

Data:

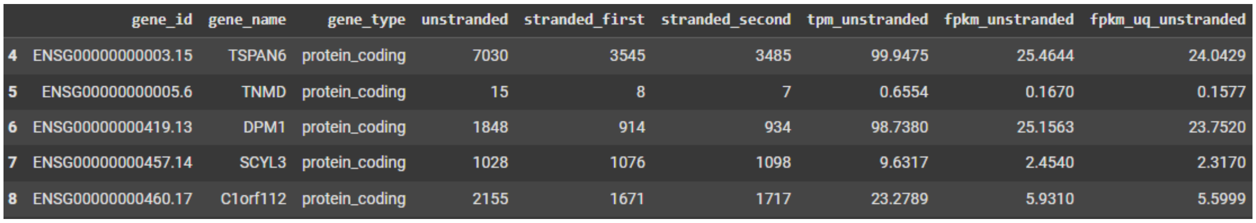
All the data I found regarding Testicular Cancer patients were reformatted subsets of The National Cancer Institute's TCGA-TGCT (The Cancer Genome Atlas Testicular Germ Cell Tumors) Dataset. The dataset consisted of 263 patients with a huge variety of medical reports and test results.



Data Category	**Cases** (n=263)		**Files** (n=12,738)	
Biospecimen	263	100.00%	836	6.56%
Clinical	263	100.00%	677	5.31%
Copy Number Variation	262	99.62%	2,689	21.11%
DNA Methylation	150	57.03%	468	3.67%
Proteome Profiling	118	44.87%	122	0.96%
Sequencing Reads	263	100.00%	1,450	11.38%
Simple Nucleotide Variation	262	99.62%	3,982	31.26%
Somatic Structural Variation	252	95.82%	1,138	8.93%
Structural Variation	206	78.33%	752	5.90%
Transcriptome Profiling	150	57.03%	624	4.90%

I focused on Transcriptome Profiling data which only has data from 150 patients. I had to narrow my scope since downloading anything larger than 5GB of data from this site required approval.

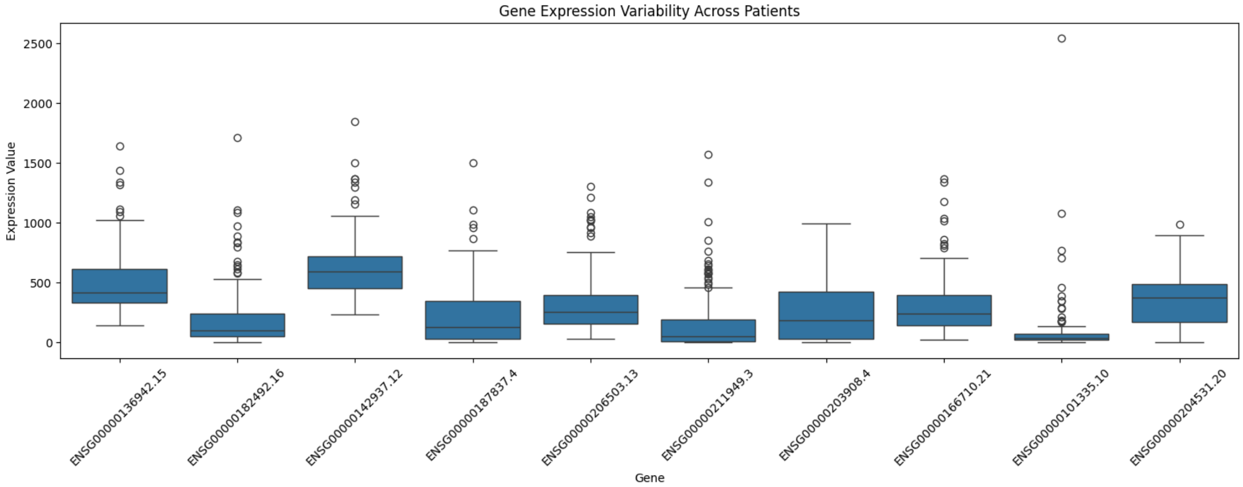
The structure of this data per patients looks like this:



	gene_id	gene_name	gene_type	unstranded	stranded_first	stranded_second	tpm_unstranded	fpkm_unstranded	fpkm_uq_unstranded
4	ENSG00000000003.15	TSPAN6	protein_coding	7030	3545	3485	99.9475	25.4644	24.0429
5	ENSG00000000005.6	TNMD	protein_coding	15	8	7	0.6554	0.1670	0.1577
6	ENSG00000000419.13	DPM1	protein_coding	1848	914	934	98.7380	25.1563	23.7520
7	ENSG00000000457.14	SCYL3	protein_coding	1028	1076	1098	9.6317	2.4540	2.3170
8	ENSG00000000460.17	C1orf112	protein_coding	2155	1671	1717	23.2789	5.9310	5.5999

I then reformatted the data so that the columns were gene IDs and each row was a patient. The value at each cell was the FPKM (Fragments Per Kilobase per Million reads) value. I maintained a `'is_seminoma'` column for the y labels.

In this format, I was able to calculate the variance of the FPKM across patients for each gene and sorted them in descending order. Here's a visualization of the top 10 most varying genes in the dataset:



<u>Methods/Experimentation:</u>

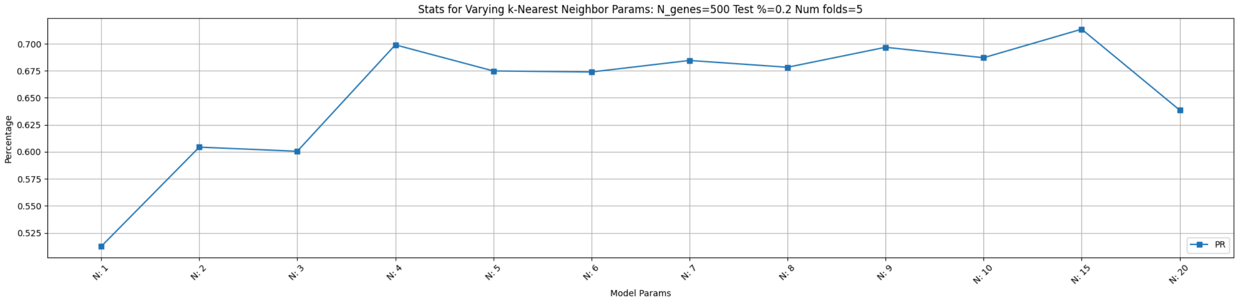
I tested the following 5 models attempting different variations of the mentioned hyperparameters for each model. **Model** (Hyperparameters…): **Random Forest** (N trees), **k-NN**

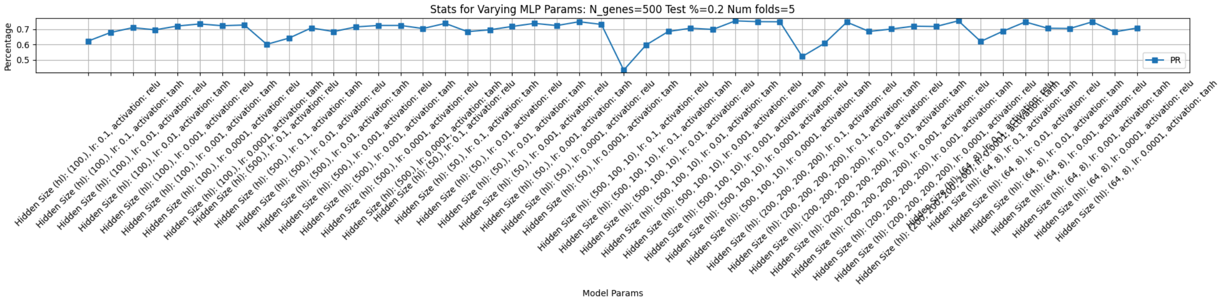
(N neighbors), **Logistic Regression** (Regularization strength and method (L1 / L2), Solver (liblinear / newton-cholesky / newton-cg), Primal / Dual), **SVM** (Linear / RBF kernel), **MLP** (Hidden layer amount and sizes, Regularization strength, Learning rate, Activation (relu / tanh)).

Also there were approximately ~17000 genes for a patient after deduping the IDs. I performed hyperparameter search on the 5 models listed above each time for the top 100, 500, 1000, and 2000 genes. I also experimented with 80/20 and 75/25 train/test split. And I did cross validation on the Precision-Recall area under the curve (PRAUC) with n=5 folds to determine the best hyperparameter configuration per model.

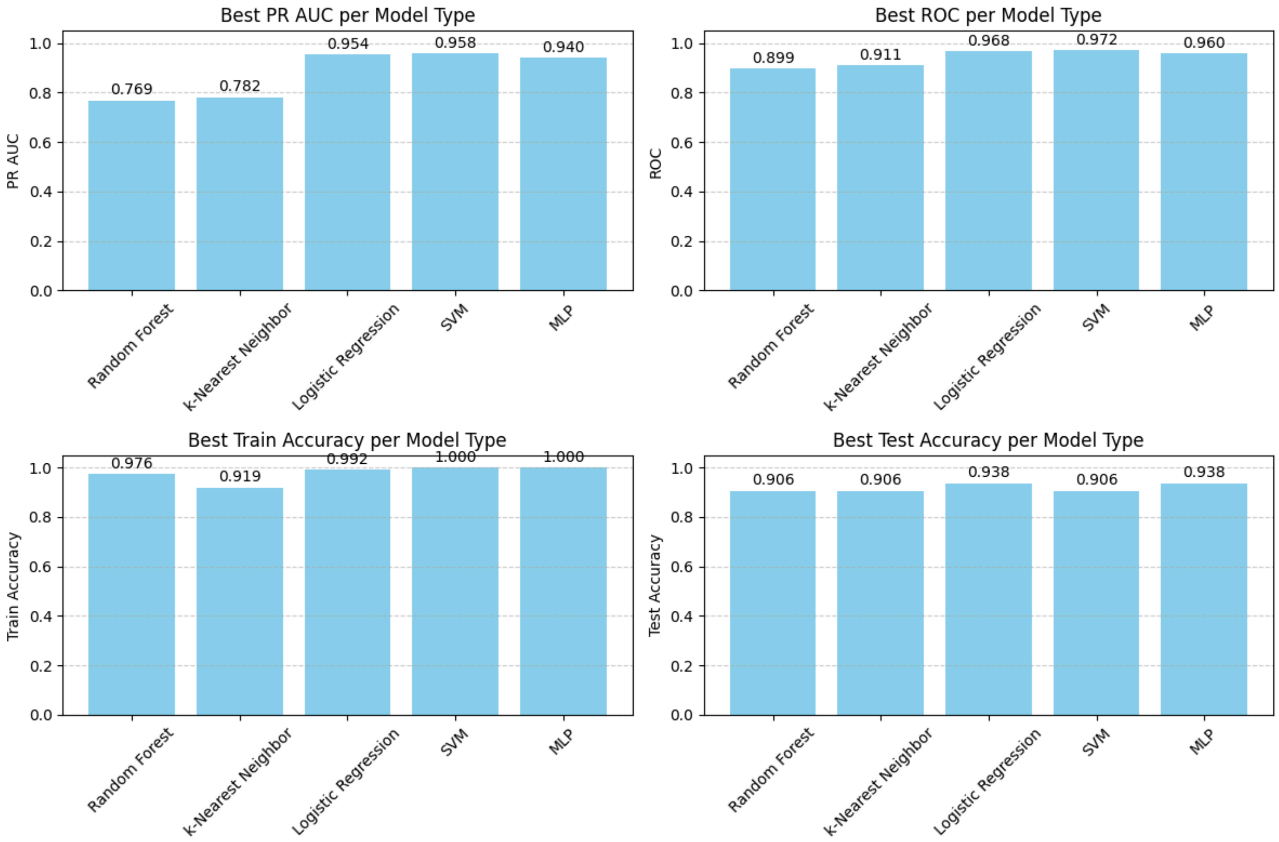
Results:

As an example of what I explored, here's the hyperparameter search PR AUC values for k-NN and MLP:

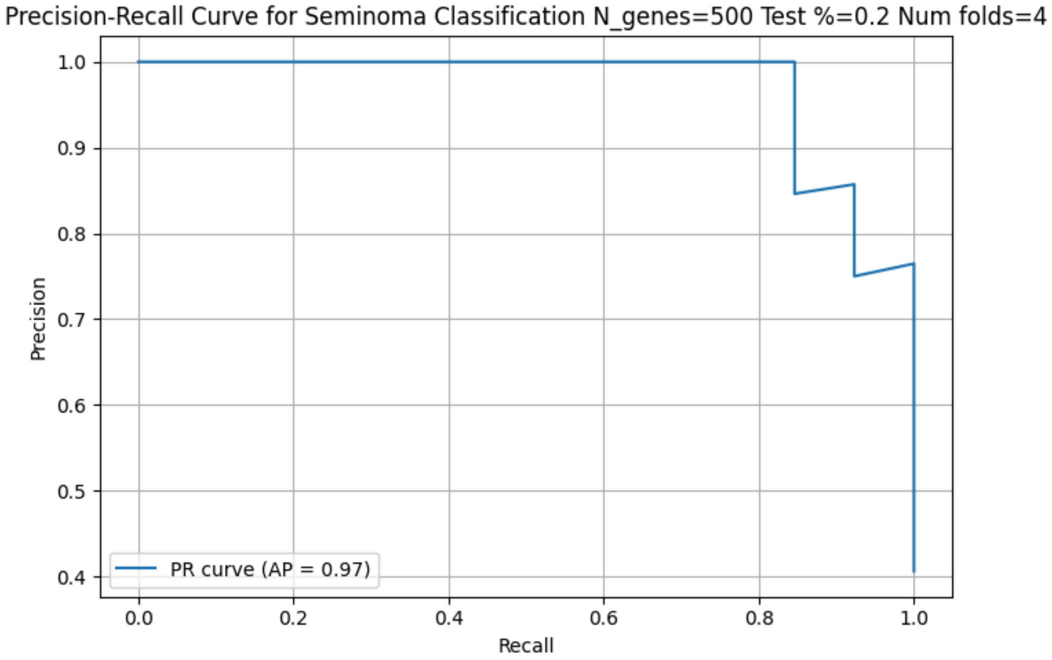




And here are all the model's best results for Top Genes = 500, 80/20 Train/Test Split, and Cross Folds = 4/5. I found that these parameters did consistently better:



Here's the precision-recall curve for the MLP Classifier:



Future Work:

Given more time I would like to try messing with more hyperparameter configurations and see if the accuracy could be improved. I could also change which column of the gene count I used as my training data. If possible I would've liked to explore different kinds of data, but the site understandably had large data restrictions.

For this project I used sklearn for all my classifiers, so trying a different library like PyTorch would be interesting to explore since it gives me much more control on what I put in the model. There would be a lot more hyperparameters to explore with that as well.

Reflection:

I really enjoyed the research and was surprised how well the comp bio (427), ML (446), and Deep Learning (493g1) classes here at UW prepared me for a project like this. I found that the main difficulty wasn't the models themselves since I could use libraries, but connecting data between different stages and methods (collection, setup, training, testing, plotting results). In hindsight I liked exploring different datasets and piecing together a full picture of what the data represents and what I can do with it.