p53 mutants

Project Work Data Analytics

Muhammad Junaid Jawaid

10.07.2023

Table of Contents

1. Dataset
   - Description
   - Context
   - Cleaning
2. Exploratory analysis
   - Bias analysis
   - Basic discriptives and scaling
   - Principal Component Analysis
3. Preprocessing for ML model training
   - Dimensionality reduction
4. Predicitve Analytics
   - Logistic Regression
   - Support Vector Machines
   - Random Forest
   - Neural Network

p53 mutants

Dataset

- the goal is to model mutant p53 transcriptional activity (active vs inactive)
- biophysical models of mutant p53 proteins yield features which can be used to predict p53 transcriptional activity
- data extracted from biophysical simulations
- class labels are determined via in vivo assays

5409 attributes per instance:

- **Attributes 1-4826:** 2D electrostatic and surface based features
- **Attributes 4827-5408:** 3D distance based features
- **Attribute 5409:** class attribute, i.e. active (transcriptonally competent) or inactive (cancerous)

p53 protein

Info p53 protein:

- tumor suppressor → Regulates DNA replication during cell division
- mutations in p53 may lead to a loss of its core function

⇒ uncontrolled cell growth (characteristic of many cancer types)

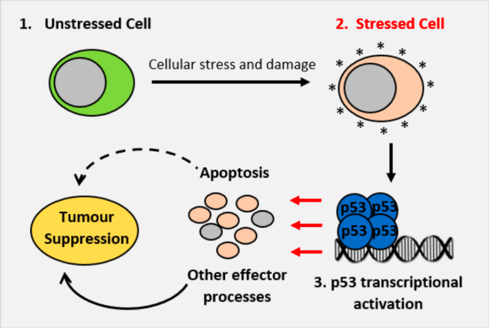
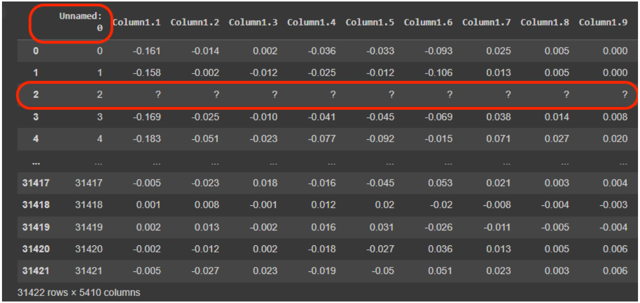


Fig. 1: Graphic from www.thebiomics.com

Data cleaning

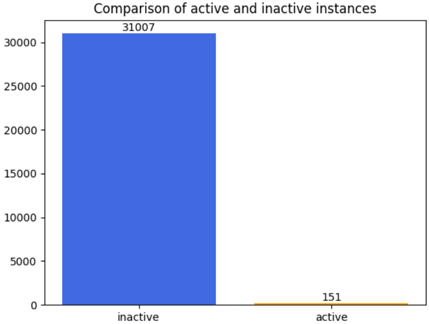


	Unnamed: 0	Column1.1	Column1.2	Column1.3	Column1.4	Column1.5	Column1.6	Column1.7	Column1.8	Column1.9
0	0	-0.161	-0.014	0.002	-0.036	-0.033	-0.093	0.025	0.005	0.000
1	1	-0.158	-0.002	-0.012	-0.025	-0.012	-0.106	0.013	0.005	0.000
2	2	?	?	?	?	?	?	?	?	?
3	3	-0.169	-0.025	-0.010	-0.041	-0.045	-0.069	0.038	0.014	0.008
4	4	-0.183	-0.051	-0.023	-0.077	-0.092	-0.015	0.071	0.027	0.020
...	...	...	...	...	...	...	...	...	...	...
31417	31417	-0.005	-0.023	0.018	-0.016	-0.045	0.053	0.021	0.003	0.004
31418	31418	0.001	0.008	-0.001	0.012	0.02	-0.02	-0.008	-0.004	-0.003
31419	31419	0.002	0.013	-0.002	0.016	0.031	-0.026	-0.011	-0.005	-0.004
31420	31420	-0.002	-0.012	0.002	-0.018	-0.027	0.036	0.013	0.005	0.006
31421	31421	-0.005	-0.027	0.023	-0.019	-0.05	0.051	0.023	0.003	0.006

31422 rows × 5410 columns

- **Dimesions**: $5,409$ columns and $31,422$ rows (after deleting first unnamed column)
- 263 rows contain **question marks (?)** for at least one of the featues
  $\Rightarrow$ substitute by null values
  $\Rightarrow$ delete all rows/instances containing any null value
- Only 1 single **duplicate** contained in the dataset
  $\Rightarrow$ drop duplicates

Imbalanced dataset

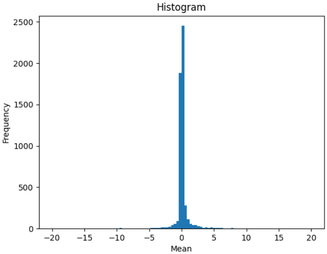


- Number of inactive instances: 31007
- Number of active instances: 151

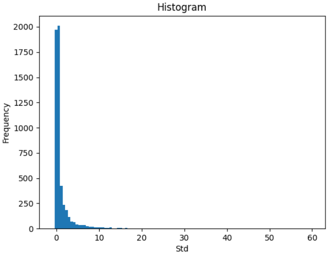
⇒ Huge discrepency between the active and inactive instances.

Distribution analysis

Investigate minimum and maximum mean/standard deviation (std) as well as the distribution of the mean/std over the feature columns.



(a) Distribution of mean over feature columns:
Max. mean 61.469845,
Min. mean −81.884894.

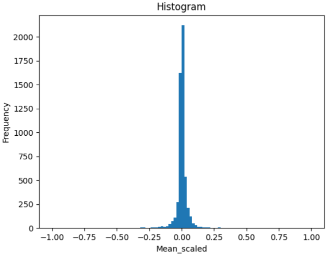


(b) Distribution of std over feature columns:
Max. std 60.015421,
Min. std 0.006925.

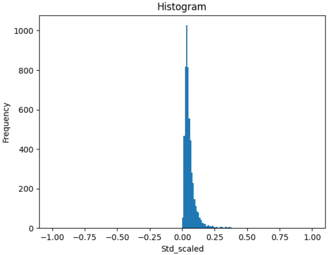
The features are of different magnitudes, i.e. we need to **scale the data into a common range ⇒ MaxAbsScaler**

Distribution analysis after applying MaxAbsScaler

MaxAbsScaler is a scaler that transforms each feature by dividing them by the maximum absolute value of that feature, ensuring that the resulting values are within the range [-1, 1].

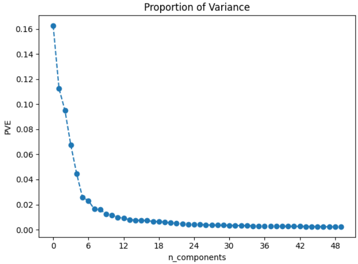


(a) Distribution of mean over feature columns:
Max. mean 0.610389,
Min. mean −0.628182.

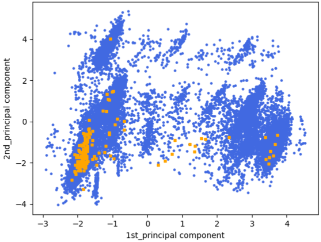


(b) Distribution of std over feature columns:
Max. std 0.442676,
Min. std 0.005897.

PCA on the transformed feature data set



(a) Proportion of variance explained. We observe a number of 7 principal components until the first elbow in the PVE plot, i.e. PVE > 2%. In total they capture a proportion of 53.1% of variance in the data.



(b) Projection onto the space spanned by the 1st and 2nd principal component. Orange points indicate active instances, blue points the inactive ones.

Drop features with low variance

High dimensionality of dataset (5408 features) leads to high complexity

⇒ training of sophisticated ML models is too computationally expensive

⇒ search for features with low variance and drop them, as they do not provide any further insights to perform the classification task

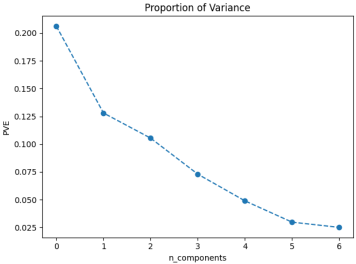
Quartiles for the distribution of the std across columns:

Statistic	Value
Count	5408.000000
Mean	0.056390
Standard Deviation	0.045926
Minimum	0.005897
25th Percentile	0.030126
50th Percentile (Median)	0.043774
75th Percentile	0.066678
Maximum	0.442676

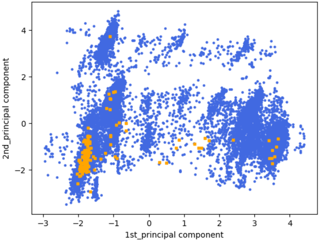
⇒ **delete** all features with a **variance lower than the** 75 **% quantile**

Drop features with low variance

We apply PCA again on the dataset with the reduced number of features to see if we can capture now a higher proportion of variance in the data.



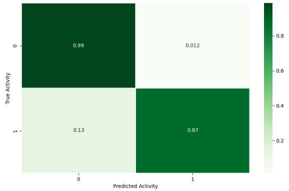
(a) Proportion of variance explained. We will represent our data in coordinates w.r.t. the first 7 principal components having each a PVE $>$ 2% and explaining a total proportion of 61.7% of variance in the data.



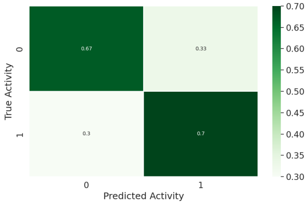
(b) Projection onto the space spanned by the 1st and 2nd principal component. Orange points indicate active instances, blue points the inactive ones.

Logistic Regression

Class_weights = 'balanced' was used to address the problem of imbalance in the data set,



(a) Confusion Matrix for Model trained on high variance 1352 features, we used Features of Standard Deviation > 0.066678.



(b) Confusion Matrix for Model trained on high variance 7 Principle Components.

Logistic Regression - Accuracy measures

Table 1: Performance Scores for model trained on 1352 high variance features

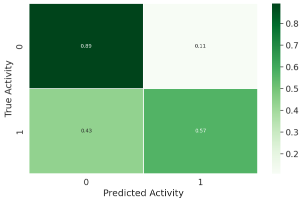
Dataset	ACC	BA	RECALL	PRECISION
Training	0.988446	0.994195	1.000000	0.295844
Testing	0.987323	0.927287	0.866667	0.257426

Table 2: Performance Scores for model trained on 7 prinipal components

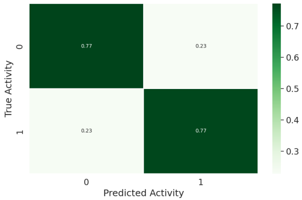
Dataset	ACC	BA	RECALL	PRECISION
Training	0.679251	0.748377	0.818182	0.012265
Testing	0.670732	0.685295	0.700000	0.010174

Support Vector Machines (SVM)

We Applied 5 fold Cross Validation Grid Search with Support Vector Machines for Regularization parameter, kernels (polynomial, Gaussian, Sigmoid), polynomial degree, with class_weights = 'balanced'.



(a) Confusion Matrix for SVM with scoring set to precision, Optimial Paramaters: C=200, Kernel=Gaussian



(b) Confusion Matrix for SVM with scoring set to balanced accuracy. Optimial Parameters: C=1, Kernel= Polynomial, degree =2

Support Vector Machines (SVM) (Accuracy Measures)

Table 3: Performance scores with scoring set to precision

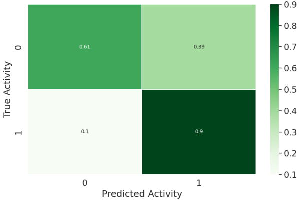
Dataset	ACC	BA	RECALL	PRECISION
Training	0.891559	0.929067	0.966942	0.041548
Testing	0.889281	0.728754	0.566667	0.024496

Table 4: Performance scores with scoring set to balanced accuracy

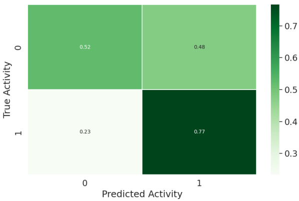
Dataset	ACC	BA	RECALL	PRECISION
Training	0.777421	0.826487	0.876033	0.018798
Testing	0.772304	0.769499	0.766667	0.016028

Random Forest

We applied 5 fold Cross Validation Grid Search to find optimal values for pruning parameter, and criterion (log_loss, entropy).
Optimal Parameters: Pruning = 0.1, Criterion: Entropy



(a) Confusion Matrix for Random Forest applied on variables with Standard Deviation > 0.06678



(b) Confusion Matrix for Random Forest Applied on 7 Principal Components

Random Forest (Accuracy Measures)

Table 5: Performance scores for Random Forest with dataset of all variables having standard deviation > 0.06678

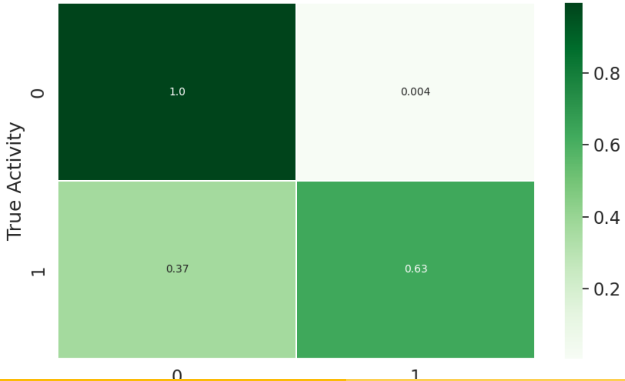
Dataset	ACC	BA	RECALL	PRECISION
Training	0.614940	0.798307	0.983471	0.012249
Testing	0.612323	0.755466	0.900000	0.011066

Table 6: Performance scores for Random Forest performed on dataset with 7 principle components

Dataset	ACC	BA	RECALL	PRECISION
Training	0.530570	0.739468	0.950413	0.009738
Testing	0.525193	0.645346	0.766667	0.007731

Neural Network

We implemented Neural Network on a 7 principle components dataset with two hidden layers and found out the best results were possible with 100, 10 nodes respectively for each Layer. We used the sigmoid function for activation and adaptive learning rate.



Neural Network Accuracy Measures

Table 7: Neural Network Accuracy Measures

Dataset	ACC	BA	RECALL	PRECISION
Training	0.997753	0.970087	0.942149	0.699387
Testing	0.994223	0.814651	0.633333	0.431818

Conclusion

Models (Dataset)	Hyperparameters	ACC	BA	Recall	Precision
Logistic Regression (high Variance)	–	0.98	0.86	0.73	0.25
Logistic Regression (PCA)	–	0.67	0.76	0.85	0.01
SVM (PCA)	C=200, Gaussian Kernel	0.89	0.74	0.60	0.02
SVM (PCA)	C=1, Polynomial Kernel deg=2	0.68	0.76	0.85	0.01
Random Forest (high Variance)	Pruning=0.1, Entropy Criterion	0.62	0.81	1.00	0.01
Random Forest (PCA)	Pruning=0.1, Entropy Criterion	0.52	0.68	0.84	0.01
Neural Network (PCA)	Layer1=100 nodes, layer2=10 nodes	0.99	0.81	0.63	0.43

Table 8: Accuracy Measure comparison for different Algorithms and Datasets