ML2020: Project report

Lovisa Franzén
Email: lovisa.franzen@scilifelab.se
August 31, 2020

Description

In this project, the aim was to apply various machine learning (ML) methods to a dataset of choice and evaluate their performance. As I work with spatial transcriptomics (ST) in my PhD studies, I decided to use such datasets for this project. In short, ST generates whole transcriptome data from a tissue section by capturing polyadenylated RNA species using complementary capture sequences printed in a spotted pattern on a glass slide [1]. A popular tissue to analyze with ST are tumors, which enables characterization of the intratumor heterogeneity. Within the tumor tissue, there are normally regions of cancerous tissue as well as areas of healthy or normal tissue. In these experiments, we usually let a pathologist annotate the histological image of the tissue section, in order to guide us when analyzing the expression data of that same tissue section. Most often, there is a strong correlation between the pathologist's annotation and expression profiles generated through unbiased clustering of the data.

Due to the limited scope of this project and my relative unfamiliarity with applying ML to process ST data, I have for this project chosen to apply ML to a set of breast cancer ST datasets where each data point ("spot") has been annotated as either cancer or healthy, and using ML see to whether I can predict these labels using only the gene expression data.

Code

I have written all the code in Python 3 using Jupyter Notebook. A conda environment was set up for this project and the code was version controlled using Git. All ML methods were acquired from scikit-learn.

All code can be accessed here: https://github.com/lfranzen/ml-course-project

The data

I was given access to data produced within our research group containing several ST datasets from breast cancer tissue sections. The data is part of a preprint by Alma Andersson et al. [2] and is also available on GitHub https://github.com/almaan/her2st.

Description of the data

Annotated datasets coupled with annotated histological images are available for eight samples: A1, B1, C1, D1, E1, F1, G2, and H1. These are the datasets I have chosen to work with for this project. Within all datasets are data points representing healthy or cancerous regions, although in varying proportions (figure 6).

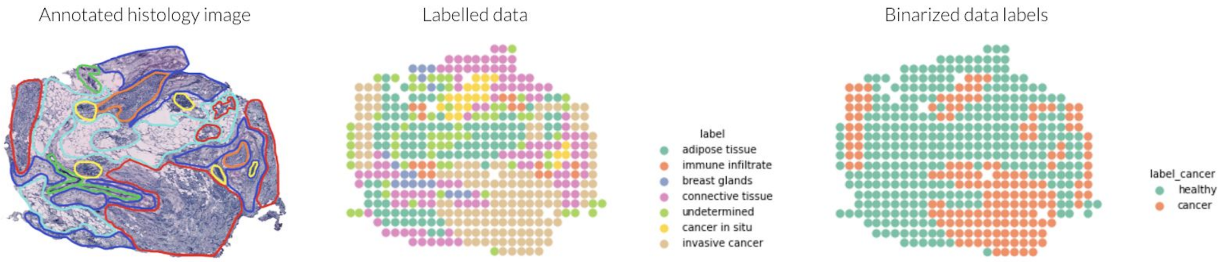


Figure 1. Histology annotations and data labels for dataset G2.

Although the data contains detailed annotations of the samples carrying more information than simply cancer or healthy, I have chosen to simplify the classification problem into binary classification. This will make both the validation and interpretation of the output easier and it is nonetheless the first thing I would choose to do before moving into multiclass predictions if the binary predictions would turn out to be accurate enough.

Data processing

The expression matrices I started working with had already been filtered based on total expression in each spot and per genes, in order to remove spots and genes without data. Moreover, genes belonging to certain biotypes as well as mitochondrial genes had been removed while keeping only the protein coding genes.

Before using the data for training the ML models, I performed the following transformations of the data on a per-sample basis:

- **Filtering**: Further remove lowly expressed spots and genes
- **Normalization**: Sequencing depth normalization (TPM)
- **Scaling**: Min-max scaling (0-1)

The filtering and normalization was performed using the Scanpy python module, which is designed for single-cell RNA-seq data but can largely be applied to ST data as well. (https://scanpy.readthedocs.io/en/stable/) [3]. It could be seen that the different samples show difference in mean expression distributions when looking at all transcripts (figure 2A), however when looking at the traditional housekeeping gene GAPDH (figure 2B) it seems to show a similar expression pattern in all samples and the HER2 encoding gene ERBB2 (figure 2C) also seems to be expressed in a similar range across all samples. Therefore, I am not too worried about the differences in mean expression which is probably a result of the large variation of morphological differences between samples.

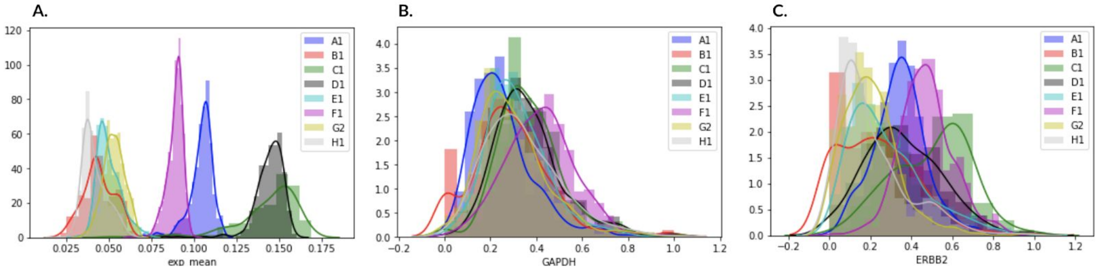


Figure 2. Distribution of expression values in all datasets. A) Mean normalized expression, B) expression of housekeeping gene GAPDH, and C) expression of the HER2 encoding gene ERBB2.

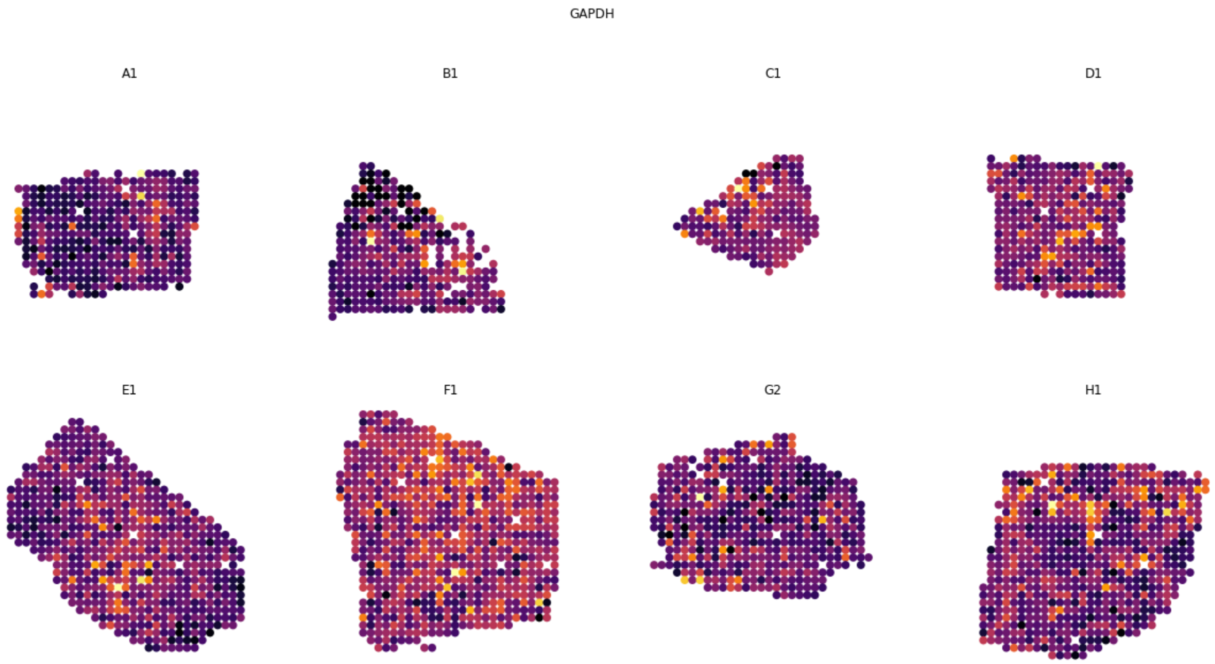


Figure 3. Normalized and scaled expression values for ERBB2 for each spot in each sample.

We can also take a look at the spatial expression pattern of a few selected genes (figure 4). ERBB2 and SDC1 are genes that may be more highly expressed in cancerous regions, while IGKC is expressed by immune cells (B cells) and FABP4 is a marker of adipose tissue.

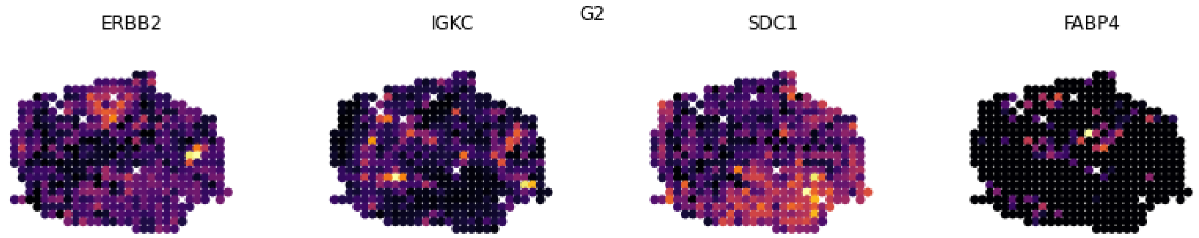


Figure 4. Normalized and scaled expression values for four genes (ERBB2, IGKC, SDC1, and FABP4) for each spot in the G2 sample which all exhibit different spatial expression patterns.

Split the data

After all data has been processed, it is ready to be split into training and test sets. As the usage for the final model will be to classify data coming from one sample at the time, I will split the data based on sample as well as perform cross validation (CV) on the training data using the leave-one-group out approach where data from one sample at the time will act as validation set.

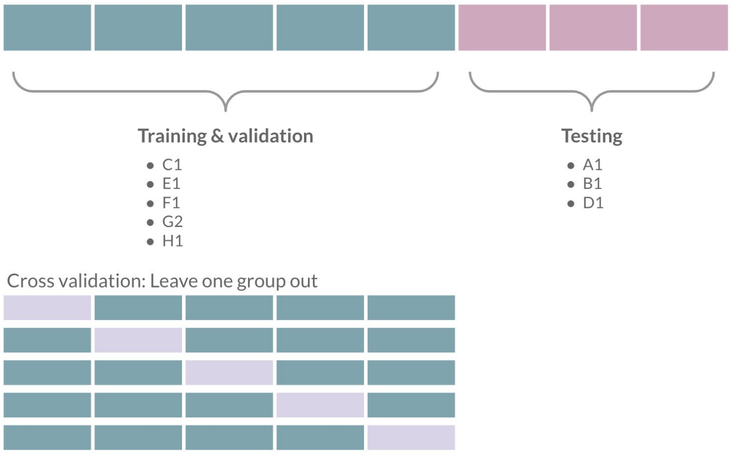


Figure 5. Splitting of datasets into training and test sets. The train set will further be used for both training and validation using the leave-one-group-out approach where a group corresponds to a sample.

The samples going into the test data were selected to ensure they represent data with different ratios of healthy and cancerous observations. Approximately 27% of the total number of observations (spots) were allocated for testing, and a total of 2532 observations were used for training.

Using data from only one source, it will be hard to fully simulate a situation where we have test data coming in from a new experiment, however, ideally that is what you would have liked to use to fully test the robustness of the model.

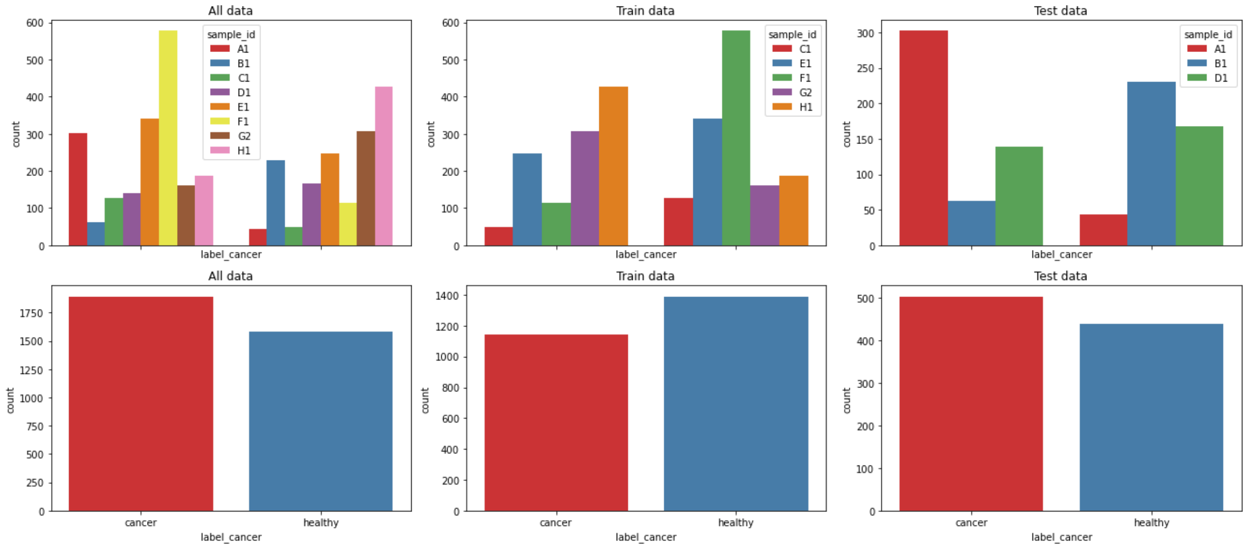


Figure 6. Count of labeled data points for all samples, the training dataset, and the test dataset.

Feature selection

The full datasets contain a total of 12696 features (genes) after filtering. Given the small number of observations in relation to that, it is not optimal to keep all those features when training the model. Therefore, I decided to limit the number of features and select only the genes displaying the greatest variability in expression between observations, and thus assume that genes which differ a lot in expression between different areas in a tissue will also be of higher importance to determine the cancer status of the regions. Based on the feature variance distribution, I decided to select only genes showing a variance greater than 0.02 for inclusion in the final training dataset, which narrowed it down to the top 1517 most variable genes.

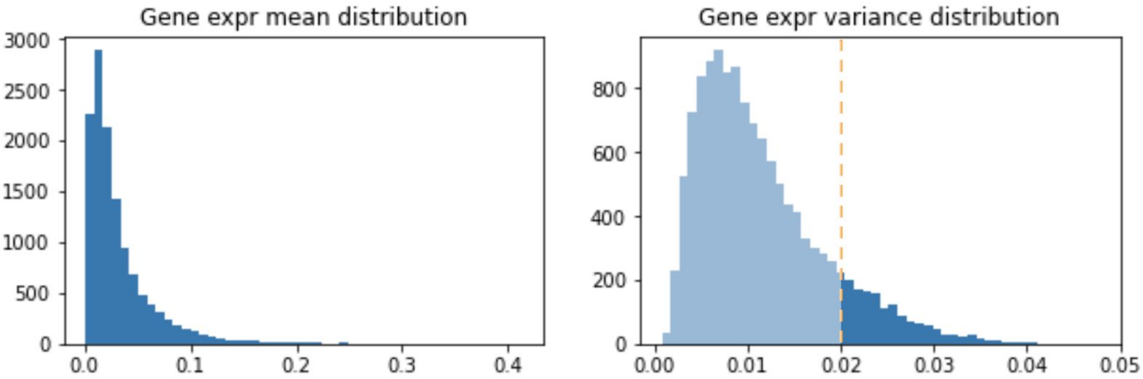


Figure 7. Distribution of mean gene expression values and gene expression variance among features in the train set data. Only features displaying a variance between observations greater than 0.02 were considered for the final training data.

Machine learning models

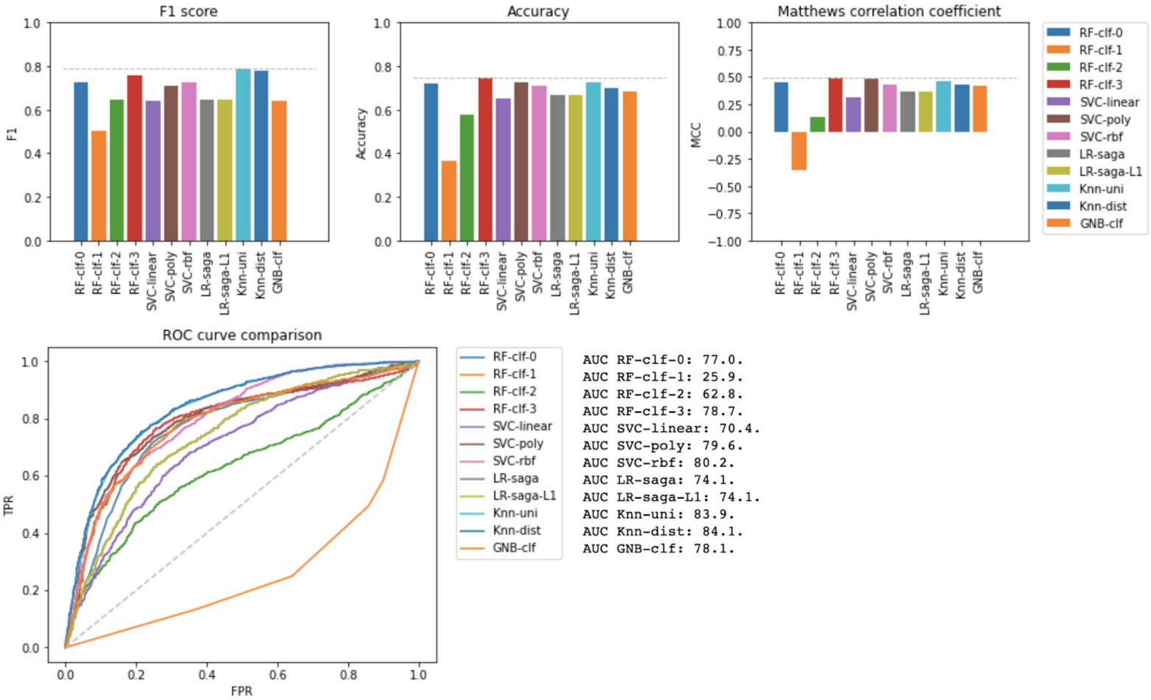
The task at hand was a classical binary classification problem, and for that Scikit-learn offers a great selection of ML algorithms to try out. I decided to pick a few of the most common ones; random forest (RF), support vector machine (SVM), linear regression (LR), k-nearest neighbor (Knn), and lastly gaussian naive bayes (GNB). For some of the selected methods I also tested different hyperparameter settings. A summary of the twelve selected models for the initial comparison can be viewed below.

- **Random forest**, `RandomForestClassifier()`
  - RF-clf-0: Only default settings
  - RF-clf-1: Hard limitations (n_estimators=20, max_depth=10, min_samples_split=0.7)
  - RF-clf-2: More permissive, and more estimators (n_estimators=300, min_samples_split=0.5)
  - RF-clf-3: Modified min samples leaf (min_samples_leaf=10)
- **SVM Classifier**, `svm.SVC(gamma='scale', C=1)`
  - SVC-linear: Linear kernel (kernel='linear')
  - SVC-poly: Polynomial kernel (kernel='poly')
  - SVC-rbf: RBF kernel (kernel='rbf')
- **Linear regression**, `LogisticRegression(solver='saga')`
  - LR-saga: Mixed L1/L2 regularization (l1_ratio=0.5)
  - LR-saga-L1: L1 regularized (l1_ratio=1)
- **Knn**, `KNeighborsClassifier(n_neighbors=20)`
  - Knn-uni: Uniform weights (weights = 'uniform')
  - Knn-dist: Distance weights (weights = 'distance')
- **GNB**, `GaussianNB()`
  - GNB-clf: Only default settings

Model evaluation

The selected models were trained on the training data and cross validated using the leave-one-group-out approach as mentioned previously. For each validation, I recorded the performance based on accuracy, F1 score, the Matthews correlation coefficient (MCC), and receiver operating characteristic (ROC) curve plus the area under the curve (AUC). These metrics capture slightly different aspects of the model performance, and to fully understand the performance of the model it is important to consider all of them. In short, the accuracy is a measure of the proportion of correct predictions out of the total number of observations considered, while the F1 score is the harmonic mean of the precision (TP/(TP+FP)) and recall (TP/(TP+FN)). The MCC describes the correlation coefficient between the true and the predicted labels and ranges between -1 and 1 where 0 is equal to a completely random prediction and 1 is a perfect correlation between predicted and actual labels. The ROC demonstrates the true positive rate (TPR) plotted against the false positive rate (FPR) and describes the trade off between the two for the model performance, and to quantify the performance you can calculate the ROC AUC.

Not surprisingly, the RF-clf-1 model which I designed to be the worst also demonstrated the worst performance based on the selected metrics. Also, the RF-clf-2 model showed poor performance, while the default RF model and RF-clf-3 were among the best performing models out of all tested. The polynomial and rbf SVM classifiers together with the Knn models also demonstrated good performance (figure 8). To proceed, I decided to select two of the best performing models and further tune their parameters using grid search CV.



```
AUC RF-clf-0: 77.0.
AUC RF-clf-1: 25.9.
AUC RF-clf-2: 62.8.
AUC RF-clf-3: 78.7.
AUC SVC-linear: 70.4.
AUC SVC-poly: 79.6.
AUC SVC-rbf: 80.2.
AUC LR-saga: 74.1.
AUC LR-saga-L1: 74.1.
AUC Knn-uni: 83.9.
AUC Knn-dist: 84.1.
AUC GNB-clf: 78.1.
```

Figure 8. Comparison of the trained ML models using selected performance metrics.

Model tuning

Although the Knn models showed some promise, I decided to proceed with the RF-clf-3 and SVC-poly models due to their high MCC score, where MCC has been argued to be the better measure compared to F1 and accuracy for binary classification [4].

To test different hyperparameter settings I ran `GridSearchCV()` for the selected models. For the RF model I tested a range of min_samples_split and max_depth values, where the highest scoring model had max_depth: 10 and min_samples_split: 106. For the polynomial SVM classifier I tested its performance at different values for the C parameter as well as at different polynomial degrees. The highest scoring SVM model had a degree of 2 and a C of 0.1. The best hyperparameters as found by the grid search were used to set up the final RF and SVM models.

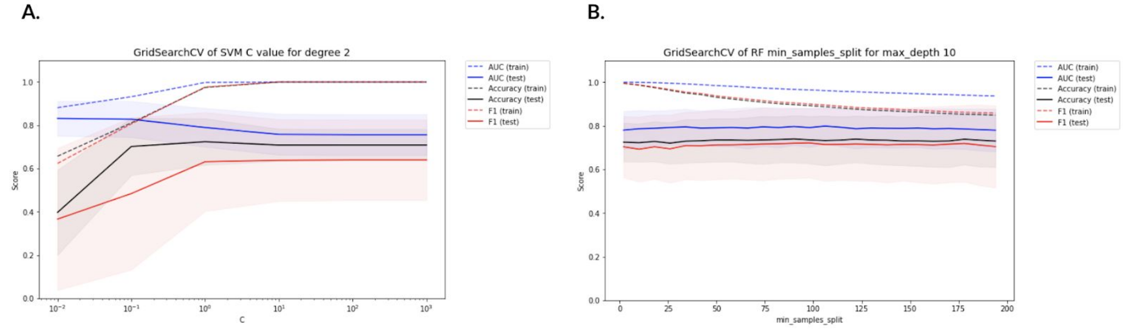


Figure 9. Examples of GridSearchCV performance for the A) SVM classifier at degree 2 and the B) RF classifier at max_depth 10.

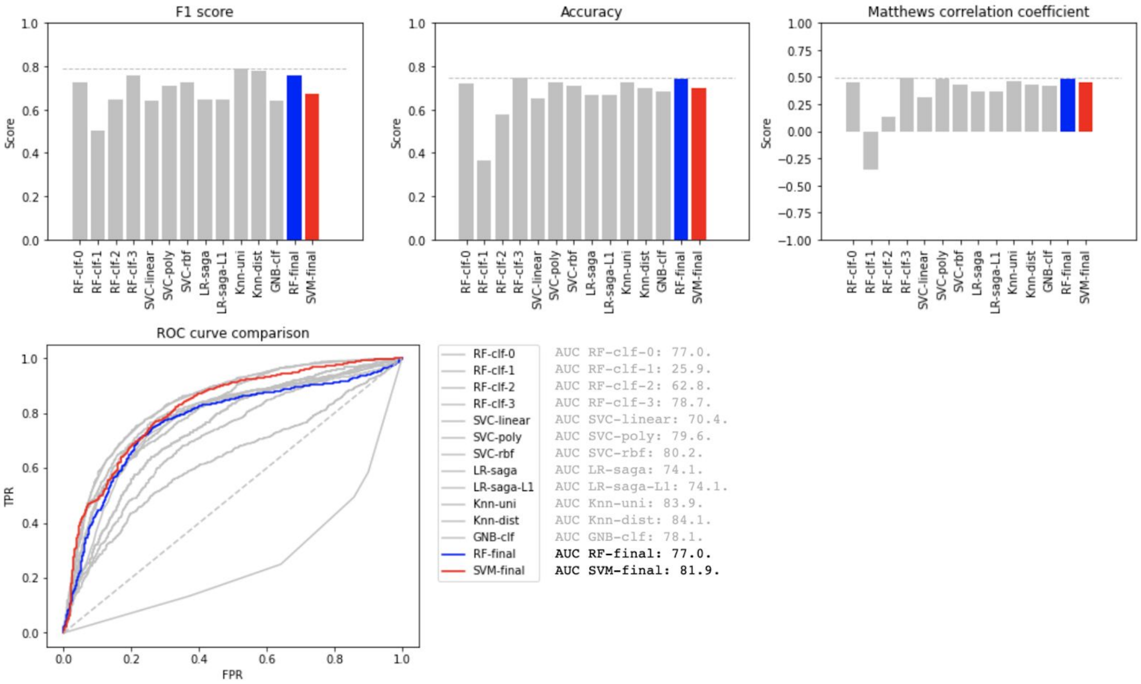


Figure 10. Comparison of the final RF and SVM classification models using selected performance metrics.

Model selection

Out of the two final models, RF-final and SVM-final, the RF model showed consistently better performance in terms of MCC, F1 score, and accuracy (figure 10). In addition, the RF model was significantly faster to train compared with the SVM model, and has the possibility to view independent decision trees to understand how individual genes contribute to the cancer or healthy classification of spots (figure 11).

The final classifier model was thus set up as follows:

```
clf_rf_final = RandomForestClassifier(n_estimators=100,
                                      max_depth=10,
                                      min_samples_split=100,
                                      min_samples_leaf=10,
                                      n_jobs=-1,
                                      random_state=1337,
                                      verbose=0)
```

After being trained on the full training dataset, the model was saved for use on the test data.

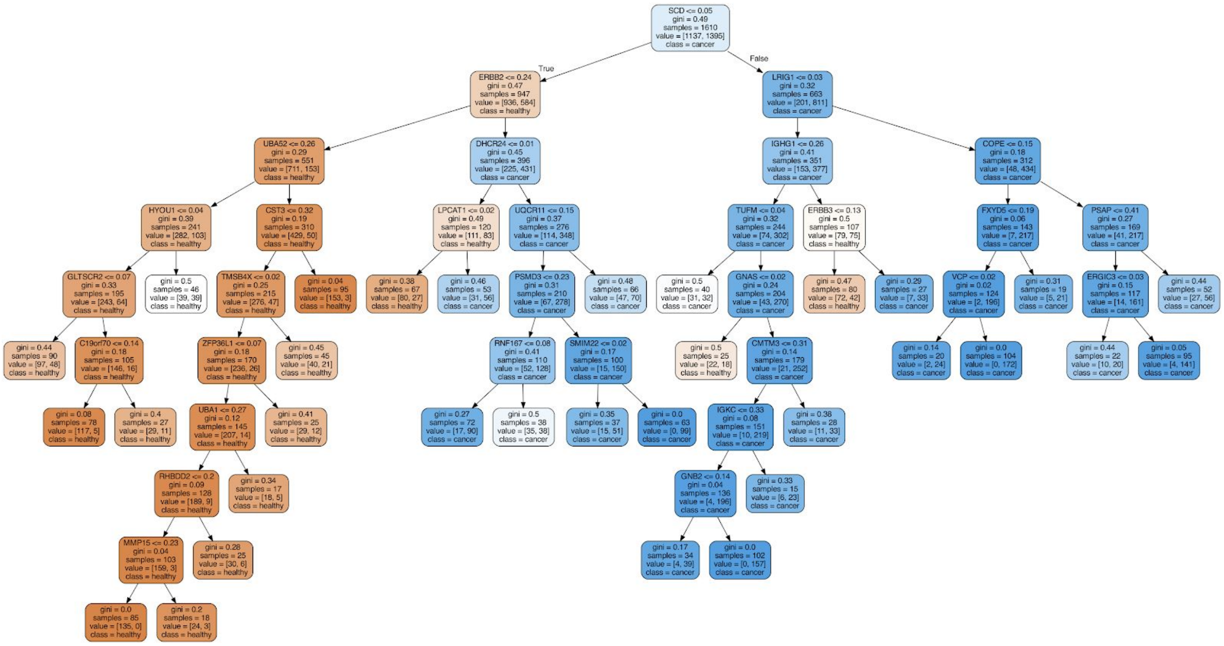

SCD <= 0.05 gini = 0.49 samples = 1610 value = [1137, 1395] class = cancer
True
False
ERBB2 <= 0.24 gini = 0.47 samples = 947 value = [936, 584] class = healthy
LRIG1 <= 0.03 gini = 0.32 samples = 663 value = [201, 811] class = cancer
UBA52 <= 0.26 gini = 0.29 samples = 551 value = [711, 153] class = healthy
DHCR24 <= 0.01 gini = 0.45 samples = 396 value = [225, 431] class = cancer
IGHG1 <= 0.26 gini = 0.41 samples = 351 value = [153, 377] class = cancer
COPE <= 0.15 gini = 0.18 samples = 312 value = [48, 434] class = cancer
HYOU1 <= 0.04 gini = 0.39 samples = 241 value = [282, 103] class = healthy
CST3 <= 0.32 gini = 0.19 samples = 310 value = [429, 50] class = healthy
LPCAT1 <= 0.02 gini = 0.49 samples = 120 value = [111, 83] class = healthy
UQCR11 <= 0.15 gini = 0.37 samples = 276 value = [114, 348] class = cancer
TUFM <= 0.04 gini = 0.32 samples = 244 value = [74, 302] class = cancer
ERBB3 <= 0.13 gini = 0.5 samples = 107 value = [79, 75] class = healthy
FXYD5 <= 0.19 gini = 0.06 samples = 143 value = [7, 217] class = cancer
PSAP <= 0.41 gini = 0.27 samples = 169 value = [41, 217] class = cancer
GLTSCR2 <= 0.07 gini = 0.33 samples = 195 value = [243, 64] class = healthy
gini = 0.5 samples = 46 value = [39, 39] class = healthy
TMSB4X <= 0.02 gini = 0.25 samples = 215 value = [276, 47] class = healthy
gini = 0.04 samples = 95 value = [153, 3] class = healthy
gini = 0.38 samples = 67 value = [80, 27] class = healthy
gini = 0.46 samples = 53 value = [31, 56] class = cancer
PSMD3 <= 0.23 gini = 0.31 samples = 210 value = [67, 278] class = cancer
gini = 0.48 samples = 66 value = [47, 70] class = cancer
gini = 0.5 samples = 40 value = [31, 32] class = cancer
GNAS <= 0.02 gini = 0.24 samples = 204 value = [43, 270] class = cancer
gini = 0.47 samples = 80 value = [72, 42] class = healthy
gini = 0.29 samples = 27 value = [7, 33] class = cancer
VCP <= 0.02 gini = 0.02 samples = 124 value = [2, 196] class = cancer
gini = 0.31 samples = 19 value = [5, 21] class = cancer
ERGIC3 <= 0.03 gini = 0.15 samples = 117 value = [14, 161] class = cancer
gini = 0.44 samples = 52 value = [27, 56] class = cancer
gini = 0.44 samples = 90 value = [97, 48] class = healthy
C19orf70 <= 0.14 gini = 0.18 samples = 105 value = [146, 16] class = healthy
ZFP36L1 <= 0.07 gini = 0.18 samples = 170 value = [236, 26] class = healthy
gini = 0.45 samples = 45 value = [40, 21] class = healthy
RNF167 <= 0.08 gini = 0.41 samples = 110 value = [52, 128] class = cancer
SMIM22 <= 0.02 gini = 0.17 samples = 100 value = [15, 150] class = cancer
gini = 0.5 samples = 25 value = [22, 18] class = healthy
CMTM3 <= 0.31 gini = 0.14 samples = 179 value = [21, 252] class = cancer
gini = 0.14 samples = 20 value = [2, 24] class = cancer
gini = 0.0 samples = 104 value = [0, 172] class = cancer
gini = 0.44 samples = 22 value = [10, 20] class = cancer
gini = 0.05 samples = 95 value = [4, 141] class = cancer
gini = 0.08 samples = 78 value = [117, 5] class = healthy
gini = 0.4 samples = 27 value = [29, 11] class = healthy
UBA1 <= 0.27 gini = 0.12 samples = 145 value = [207, 14] class = healthy
gini = 0.41 samples = 25 value = [29, 12] class = healthy
gini = 0.27 samples = 72 value = [17, 90] class = cancer
gini = 0.5 samples = 38 value = [35, 38] class = cancer
gini = 0.35 samples = 37 value = [15, 51] class = cancer
gini = 0.0 samples = 63 value = [0, 99] class = cancer
IGKC <= 0.33 gini = 0.08 samples = 151 value = [10, 219] class = cancer
gini = 0.38 samples = 28 value = [11, 33] class = cancer
RHBDD2 <= 0.2 gini = 0.09 samples = 128 value = [189, 9] class = healthy
gini = 0.34 samples = 17 value = [18, 5] class = healthy
GNB2 <= 0.14 gini = 0.04 samples = 136 value = [4, 196] class = cancer
gini = 0.33 samples = 15 value = [6, 23] class = cancer
MMP15 <= 0.23 gini = 0.04 samples = 103 value = [159, 3] class = healthy
gini = 0.28 samples = 25 value = [30, 6] class = healthy
gini = 0.17 samples = 34 value = [4, 39] class = cancer
gini = 0.0 samples = 102 value = [0, 157] class = cancer
gini = 0.0 samples = 85 value = [135, 0] class = healthy
gini = 0.2 samples = 18 value = [24, 3] class = healthy

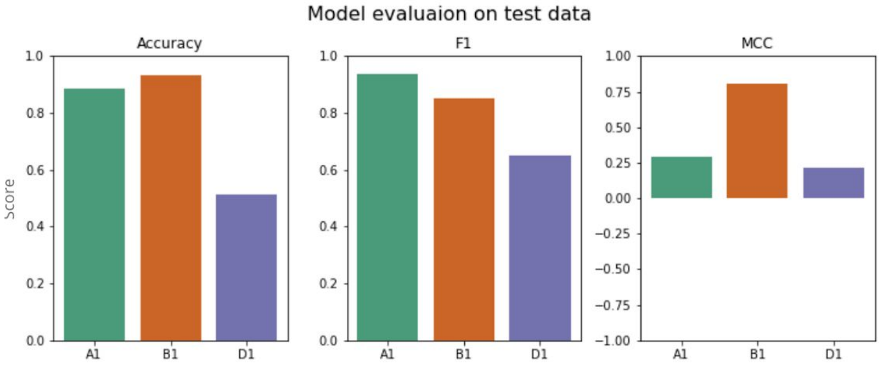
Figure 11. Decision tree for one of the estimators in the final RF model trained on the whole training dataset.

Test model

The test data that had previously been put aside was now ready to be tested on the final model to evaluate the actual performance of the model. Prior to using the test data to predict its labels, the same features used in the training needed to be selected for the test data and the labels were binarized in the same manner as for the train data. The test data consisted of samples A1, B1, and D1 and the labels were predicted for one sample at the time to simulate a real-life usage of the model.

The actual labels of the test sample data were then compared with the predicted labels and performance metrics were calculated (figure 12). It was apparent that the model performed better for certain samples (A1, B1) compared to others (D1), which may indicate that the model may not translate broadly to new data in terms of performance.

After inspecting the predictions spatially and comparing them with the histological images (figure 13), I can see that the model has the most trouble with false positives and thus incorrectly predicts healthy areas of the tissue as cancer. However, the model is almost entirely correct in its predictions on the B1 sample with a MCC of approximately 0.8 and an accuracy above 90%. Looking at the histopathology of the B1 section, it can be hypothesized that the model is better at correctly predicting healthy tissue consisting of more loose connective tissue and adipose tissue, while the more dense connective tissue seen marked blue in A1 and D1 is harder to label correctly.



	F1	Accuracy	MCC
A1	0.939	0.887	0.292
B1	0.851	0.932	0.809
C1	0.649	0.513	0.215

Figure 12. Performance of the final model on the different test datasets.

There are many potential areas of improvement for the current model and the training data. Due to the limited scope of this project, I did not spend too much time with feature engineering but instead only selected genes that were the most variable across all the training datasets. It is possible that this approach of selecting features did not capture expression patterns of all histological regions of the tissues, causing the model to miss out on information that might have made it easier for it to distinguish between cancer and healthy spots for certain cases. Although, simply increasing the numbers of features included in the training data may also worsen the performance for some ML algorithms, for instance is RF quite sensitive to having too many non-informative features.

Another area of improvement could have been the way the expression data was normalized and scaled, and to ensure the normalization diminishes the largest variabilities in mean expression distribution between samples (figure 2). While half of the samples shared similar distributions, the other half showed different distribution levels that may have made it harder for the model to both train the data and then correctly predict its labels on the test data.

Lastly, having a larger pool of training data consisting of samples from different collection dates and sources will also help towards increased robustness of the trained model.

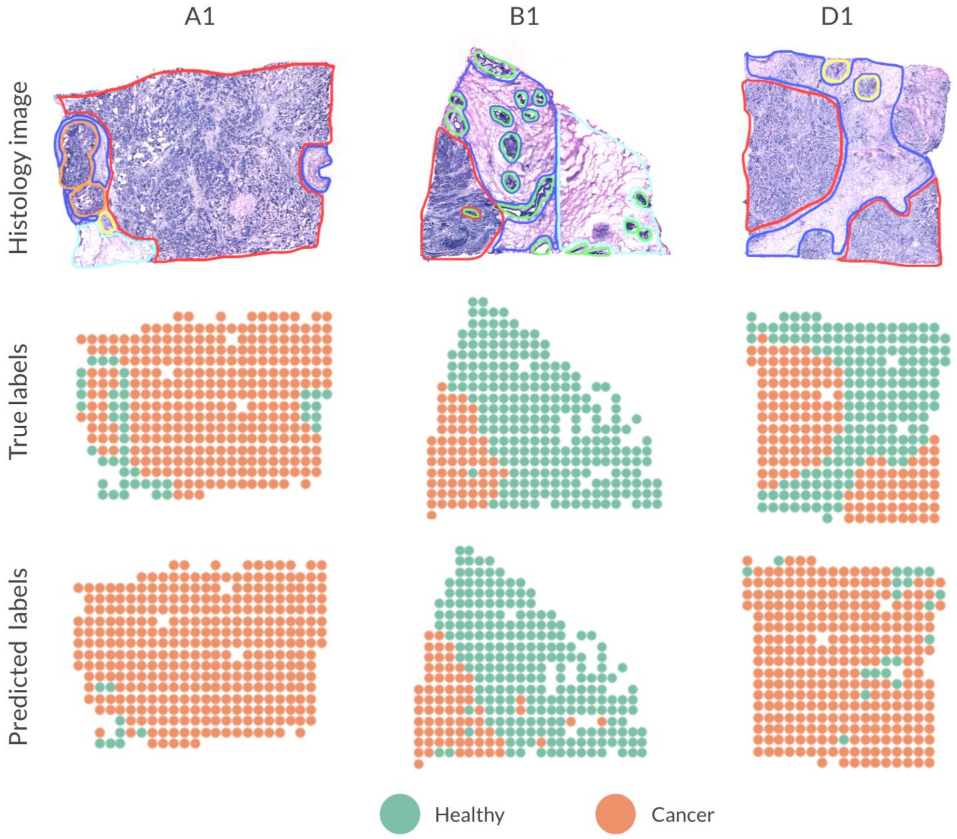


Figure 13. Spatial visualization of the predicted labels for each sample compared with the annotated (true) labels and the annotated histology images of the tissue sections. The histological annotations marked in red and orange represent cancerous regions, while the other annotations correspond to connective tissue, adipose, or other non-cancerous areas.

Conclusion

In this project, I used ST gene expression data from breast cancer tumor biopsies to train a wide range of ML classification algorithms to predict whether a spot corresponds to cancerous or healthy tissue. Five out of a total of eight datasets from different tissues (samples) were used for training and validation of the models, while the last three datasets were used for the final testing of the best performing model. From the training and validation, the best performing models were RF, SVM, and Knn, however in the end I chose to proceed with the fine tuned RF model as the final model to use for the test data. The labels for the test data were predicted for one sample at the time, and it turned out that the model worked almost perfectly for one of the samples while struggling a lot more for the other two where it incorrectly predicted most of the healthy spots as cancer.

As discussed earlier, there are many areas of improvements for this kind of classification problem and the design of the model. For instance, I did not have time to try to fine tune the Knn models, which otherwise showed some promising results, and it would have been interesting to see whether it could have outperformed the RF model in the end.

Nonetheless, I am quite happy with the final performance of the model despite its bad performance for sample D1 in particular. The fact that it could with such high accuracy correctly predict the spot labels in sample B1 indicates that it is definitely possible for a model to correctly classify whether a spot is cancerous or not given only the gene expression data. The next step would also be to introduce multi-class predictions in order to distinguish between different kinds of cancerous tissues, however that is a task for another time.

References

1. Ståhl P et al., *"Visualization and analysis of gene expression in tissue sections by spatial transcriptomics"* Science, 2016
2. Andersson A et al., *"Spatial Deconvolution of HER2-positive Breast Tumors Reveals Novel Intercellular Relationships"*, BioRxiv, 2020 https://www.biorxiv.org/content/10.1101/2020.07.14.200600v1.full.pdf
3. Wolf A, Angerer P & Theis F. *"SCANPY: large-scale single-cell gene expression data analysis"*, Genome Biology, 2018 https://genomebiology.biomedcentral.com/articles/10.1186/s13059-017-1382-0
4. Chicco D & Jurman G, *"The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation"*, BMC Genomics, 2020 https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-019-6413-7