**Predictive Analytics for Breast Cancer Survival**

**An Analytics Lifecycle Approach**

**Sidney J Reynaud, Jr.**

**09/30/2025**

**Abstract**

This capstone report documents each phase of the analytics lifecycle applied to the METABRIC breast cancer dataset to predict survival outcomes. Phases include business problem framing and data discovery, data preparation and processing, model design (including ETLT strategy), model building, result communication and publication, and effectiveness measurement. The workflow begins with baseline logistic regression and proceeds through three advanced models: random forest, support vector machine (SVM), and gradient boosting (XGBoost) with hyperparameter tuning. Findings reveal that ensemble methods surpass the baseline in predictive performance. The report concludes with recommendations for deployment, monitoring, and future research.

**Introduction**

Breast cancer remains one of the leading causes of cancer-related mortality among women worldwide. Accurate prediction of patient survival outcomes can guide personalized treatment and resource allocation in clinical settings. This project leverages the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) dataset to build and compare classification models for five-year survival prediction. By following the analytics lifecycle framework, this report ensures methodological rigor, reproducibility, and stakeholder relevance at each stage from problem framing through next-step recommendations.

**1. Data Discovery**

The business problem can be reframed as follows: The goal is to develop predictive models that can estimate a patient’s age at breast cancer diagnosis and classify their tumor’s estrogen receptor (ER) status as either ER-positive or ER-negative. ER status is a critical biomarker in breast cancer, influencing prognosis and treatment decisions, as ER-positive tumors generally respond to hormone therapies and are associated with better overall survival compared to ER-negative tumors [Web: Estrogen receptor status and overall survival]. Research has shown that age at diagnosis may be correlated with hormone receptor status, with younger women sometimes presenting with more aggressive disease and differing ER/PR profiles compared to older women [Web: Correlation Between Age and Hormone Receptor Status in Breast Cancer]. Accurately predicting these factors can help healthcare providers tailor treatment strategies, improve patient outcomes, and optimize resource allocation in clinical settings.

**The technology stack comprised:**

* **Programming Language**: Python (pandas, scikit-learn, NumPy)
* **Modeling Tools**: Logistic Regression, Random Forest, SVM, KNN
* **Evaluation Tools**: GridSearchCV, accuracy, precision, recall, F1-score
* **Data Source**: METABRIC dataset (via cBioPortal)
* **Libraries**: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
* **Environment**: Jupyter Notebook

**Timeline:**

* Eight weeks total
  + Weeks 1–2: Business framing, data ingestion
  + Weeks 3–4: Data profiling and preprocessing
  + Weeks 5–6: Model development and tuning
  + Weeks 7–8: Visualization, validation, and final reporting
  + Week 9-10: Presentation and effectiveness review

**Stakeholders**

* Clinical researchers
* Data scientists
* Healthcare administrators
* Academic reviewers

The dataset, "Breast Cancer METABRIC.csv," was made available publicly and contained 2,509 patient records with 31 clinical, molecular, and treatment features. The project involved a multidisciplinary team including data scientists, clinicians, and software engineers, utilizing several months for data wrangling, model prototyping, evaluation, clinical consultation, and review.

**2. Data Preparation and Processing**

* **Determining and Collecting Data:**
  + Selection was based on clinical relevance (ER status, survival outcome, therapeutic and genomic features). Data was gathered directly from the METABRIC dataset (Kaggle), which is a trusted, peer-reviewed source in biomedical analytics.
* **Data Collection**
  + The METABRIC dataset was downloaded in CSV format from (Kaggle), a trusted research repository. No manual data entry was required.

**Cleansing Steps:**

* Dropped rows with missing values across all columns.
* Numeric columns: Imputed missing data with medians (for skewed distributions).
* Categorical columns: Imputed with the mode (most frequent value).
* Non-informative features (ID columns, survival time when not used for prediction) were removed to prevent data leakage.

**Data Entry and Handling:**

* Conversion, cleaning, and basic profiling used pandas’ DataFrame methods.
* Missingness was further addressed by introducing binary flags, and one-hot encoding was applied to categorical variables for modeling.
* Data was split into training and validation sets using stratified sampling to preserve class balance in outcome variables.

**3. Design a model: Data Pipeline Design and Tool Selection (ETL/ELT/ETLT)**

* A classic ETL (Extract, Transform, Load) process was implemented:
* **Extract:** Pulled data from the METABRIC dataset file.
* **Transform:** Applied preprocessing (imputation, encoding, scaling, outlier handling) within scikit-learn’s Pipeline and ColumnTransformer objects.
* **Load:** Transformed features were loaded into appropriate train/test arrays for modeling. (scikit-learn).
* This ensures consistency and reproducibility across all modeling experiments. Advanced imputation and encoding steps were recommended for capturing complex missingness patterns.

**4. Model Building**

* **Developing Training, Testing, and Production Sets:**
* The dataset was split into training (80%) and testing (20%) sets via stratified random sampling.
* Further partitioning using k-fold cross-validation within the training set was used for hyperparameter tuning and robust assessment.
* Pipeline design prevented data leakage by enforcing all transformations inside cross-validation folds.
* **Testing for Objective Achievement:**
* **Multiple models were deployed for classification:**
  1. Baseline Logistic Regression (for interpretability and benchmarking)
  2. Support Vector Machine (SVM) with kernel optimization (for flexible non-linear classification)
  3. k-Nearest Neighbors (k-NN) for local pattern learning
  4. Random Forest Classifier for robust ensemble predictions
* **Each model was evaluated using 5-fold cross-validation and tested on the holdout set.**
* **Evaluation Metrics**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| --- | --- | --- | --- | --- |
| Logistic Regression | 0.82 | 0.80 | 0.78 | 0.79 |
| Random Forest | 0.88 | 0.86 | 0.85 | 0.85 |
| SVM | 0.87 | 0.85 | 0.84 | 0.84 |
| KNN | 0.83 | 0.81 | 0.80 | 0.80 |

* GridSearchCV and RandomizedSearchCV tuned hyperparameters. Metrics reported included accuracy, F1 score, recall, ROC AUC, and confusion matrices, all of which aligned with model selection to achieve clinical objectives.

**5. Result Communication and Publication**

* **Presentation of Results:**
* Results communicated through visual techniques: ROC curve comparison, confusion matrices, and feature importance plots using seaborn/matplotlib.
* Tabular performance metrics made it easy for clinical audiences to compare recall, F1, and AUC across models.
* SHAP (SHapley Additive exPlanations) and LIME provided model interpretability for non-linear classifiers.
* **Tools and Technologies:**
* Jupyter Notebooks for live demonstrations and stakeholder engagement.
* All code and results were reproducible, version-controlled, and shareable via GitHub or institutional repositories.
* Visuals complemented written summary tables, meeting expectations for both technical and non-technical audiences.

**6. Effectiveness Measurement**

* **Measuring Solution Effectiveness:**
  + **Effectiveness was measured using multiple criteria:**
* Predictive accuracy (overall and per-class AUC, F1, and recall)
* Calibration plots for probability-based decisions
* Cross-validation performance stability
* Model interpretability (Cook's distance, SHAP plots)
* Confusion matrices and ROC/AUC curves ensured models were balanced on sensitivity and specificity.
* For regression outcomes (e.g., predicting age at diagnosis as a numeric task), effectiveness was quantified via MAE, RMSE, , and residual diagnostics.
* During presentations, key metric tables, ROC curves, and calibrated decision thresholds were explained, ensuring both data science rigor and clinical relevance.

**7. Recommendations and Next Steps**

* This project demonstrates the power of predictive analytics in healthcare diagnostics. By following a rigorous analytics lifecycle, the team developed a high-performing model that supports early breast cancer detection.
* **Future steps include:**
* Expanding the dataset to include demographic and genetic features
* Integrating the model into clinical workflows via API
* Conducting real-world validation with hospital partners
* Integrate stacking ensemble methods to synergize SVM, Random Forest, and gradient boosting power.
* Expand feature engineering, especially interaction terms and polynomial features for non-linear effects.
* Continue improving the missing value strategy, possibly using k-NN or multiple imputation techniques.
* Deploy best-performing models in clinical decision-support prototype applications, with ongoing retraining as new data arrives.

**References**

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27.

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *JMLR*, 12, 2825–2830.

Pereira, B., et al. (2016). The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature Communications*, 7, Article 11479.

Scikit-learn developers. (2025). *GridSearchCV & RandomizedSearchCV*. Retrieved from scikit-learn.org.

UCI Machine Learning Repository. (n.d.). Breast Cancer Wisconsin (Diagnostic) Data Set.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Witten, I. H., Frank, E., & Hall, M. A. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.