ML.NET CLASSIFICATION ALGORITHMS AND RESEARCH REPORT

- SDCA Logistic Regression
- L-BFGS Logistic Regression
- Averaged Perceptron
- Linear SVM
- Local Deep SVM
- FastTree (GBDT)
- GAM (Generalized Additive Model)

ABOUT MY EXPERTISE IN ML

- 1 year of education in the MicroMasters Program at UC San Diego University
- 2 years of experience investigating Side ML Projects, including:

- Build Recommendation Model
- Image Recognition
- Anomaly Detection
- Noise Cancellation
- etc.

CLASSIFICATION METRICS EXPLAINED

Metric	What It Measures	Why It Matters
Accuracy	The percentage of all correct predictions (both positive and negative).**Formula:** (TP + TN) / (TP + FP + FN + TN)	Tells you how often the model is right overall. Good when classes are balanced.
AUC (Area Under ROC Curve)	How well the model separates the two classes (e.g., recurrence vs no recurrence).**Range:** 1.0 = perfect, 0.5 = random	Measures the model's ability to rank a positive case higher than a negative one. Great for imbalanced data.
F1 Score	The harmonic mean of Precision and Recall.**Formula:** 2 × (Precision × Recall) / (Precision + Recall)	Balances the cost of false positives and false negatives. Best when both are important.
Precision	Of all predicted positives, how many were correct?**Formula:** TP / (TP + FP)	High precision = few false alarms (false positives). Important if false alarms are costly.
Recall (Sensitivity / TPR)	Of all actual positives, how many did we catch?**Formula:** TP / (TP + FN)	High recall = fewer missed cases (false negatives). Critical in medical applications like cancer detection.

SDCA Logistic Regression

SDCA (Stochastic Dual Coordinate Ascent) Logistic Regression is a linear classifier ideal for high-dimensional, sparse data. It solves the optimization problem efficiently using a stochastic approach, allowing faster convergence on large datasets.

Pros

- Very fast and scalable
- Supports L1/L2 regularization
- Suitable for text classification, medical records, etc.

Cons

- Only linear decision boundaries
- Less effective for complex, non-linear problems

Use when: High-dimensional data with mostly linear relationships.
Avoid when: Data exhibits strong non-linear dependencies.

L-BFGS LOGISTIC REGRESSION

L-BFGS (LIMITED-MEMORY BROYDEN–FLETCHER–GOLDFARB–SHANNO) IS A SECOND-ORDER OPTIMIZATION ALGORITHM. IT COMPUTES GRADIENTS MORE ACCURATELY AND EFFICIENTLY HANDLES SMALLER DATASETS WITH BETTER PRECISION.

Pros

- Highly precise optimization
- Good for small to mid-sized datasets

Cons

- Memory intensive
- Slower on large datasets

Use when: Accuracy is more important than speed.
Avoid when: Dealing with large-scale or streaming data.

Averaged Perceptron

The Averaged Perceptron updates weights only when a prediction is wrong. Over time, it averages these updates, improving stability. This algorithm is one of the simplest binary classifiers.

Pros

- Extremely fast
- Easy to implement
- Suitable for online learning

Cons

- No probabilistic output
- Poor for complex or non-linear patterns

Use when: You need a quick baseline or fast model.
Avoid when: You need accuracy or calibrated probabilities.

LINEAR SVM

LINEAR SVM (SUPPORT VECTOR MACHINE) SEEKS TO FIND A HYPERPLANE THAT SEPARATES CLASSES WITH MAXIMUM MARGIN. IT'S EFFECTIVE FOR LINEARLY SEPARABLE DATA AND ROBUST TO OUTLIERS.

Pros

- Good for high-dimensional, clean data
- Strong theoretical foundation

Cons

- Doesn't output probabilities
- Poor performance if classes overlap or are not linearly separable

Use when: You need a robust linear classifier.
Avoid when: You need probability outputs or handle overlapping classes.

Local Deep SVM

Local Deep SVM combines SVM margins with local feature learning from deep neural networks. It captures both global decision boundaries and localized patterns.

Pros

- Learns both linear and non-linear boundaries
- Combines strengths of SVM and DNN

Cons

- Computationally expensive
- Complex to implement
- Not natively supported in ML.NET

Use when: Complex data patterns require deep understanding.
Avoid when: Resources or development time are limited.

FASTTREE (GBDT)

FASTTREE IS MICROSOFT'S IMPLEMENTATION OF GRADIENT BOOSTED DECISION TREES (GBDT).
IT BUILDS AN ENSEMBLE OF SHALLOW TREES IN SEQUENCE
, WHERE EACH CORRECTS THE ERRORS OF THE PREVIOUS.

Pros

- High predictive accuracy
- Captures non-linear patterns
- Works well with tabular data

Cons

- Risk of overfitting if not tuned
- Slower training compared to linear models

Use when: High accuracy is needed and data is structured..
Avoid when: You need fast real-time prediction with small models.

FastForest (Random Forest)

FastForest is an ensemble method based on bagging decision trees.
Each tree is trained on a random subset of data, and predictions are made via majority voting.

Pros

- Stable and robust
- Handles overfitting well
- Works with mixed data types

Cons

- Less interpretable than single models
- Slower inference compared to simple models

Use when: You need strong baseline models with minimal tuning.
Avoid when: Model interpretability is a high priority.

GAM (GENERALIZED ADDITIVE MODEL)

GAM MODELS THE OUTCOME AS A SUM OF SMOOTH FUNCTIONS OF EACH FEATURE.
IT IS INHERENTLY INTERPRETABLE, SHOWING HOW EACH VARIABLE CONTRIBUTES TO THE PREDICTION INDEPENDENTLY.

Pros

- Highly interpretable
- Captures non-linear effects
- Great for risk-sensitive domains

Cons

- No feature interaction modeling
- Limited flexibility compared to full tree models

Use when: Interpretability and transparency are essential..
Avoid when: You need to model complex interactions between features.

Part 2: Research - Heart Disease Prediction with Quantum Feature

Model	Accuracy	AUC	F1 Score	Precision	Conf. Matrix (TP/FP/FN/TN)
SDCA Logistic Regression	0.930	0.990	0.939	0.982	TP=39, FP=6, FN=1, TN=54
L-BFGS Logistic Regression	0.860	0.956	0.891	0.838	TP=29, FP=3, FN=11, TN=57
Averaged Perceptron	0.880	N/A	0.905	0.864	TP=31, FP=3, FN=9, TN=57
Linear SVM	0.630	N/A	0.761	0.621	TP=4, FP=1, FN=36, TN=59
Local Deep SVM	0.920	N/A	0.930	0.981	TP=39, FP=7, FN=1, TN=53
FastTree GBDT	0.930	0.989	0.939	0.982	TP=39, FP=6, FN=1, TN=54
LightGBM GBDT	0.950	0.991	0.957	0.982	TP=39, FP=4, FN=1, TN=56
GAM	0.920	0.991	0.930	0.981	TP=39, FP=7, FN=1, TN=53

This research used a dataset with a unique 'QuantumPatternFeature' designed to enhance predictive power. Several classification models were trained and tested.

The table on the left summarizes the results.

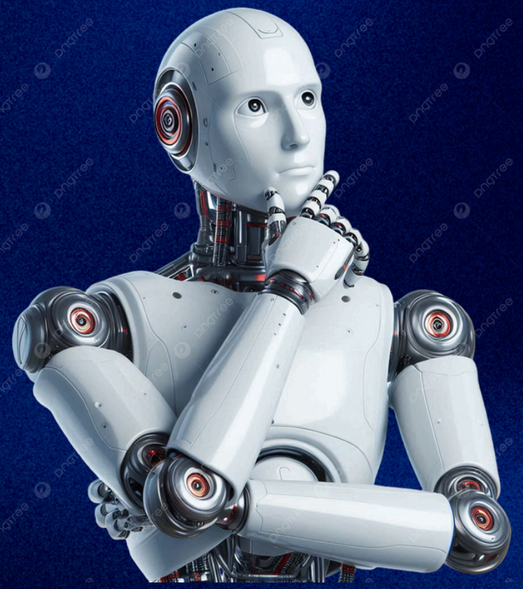
Part 3: Research - Thyroid Cancer Recurrence Prediction

Model	Accuracy	AUC	F1 Score	Conf. Matrix (TP/FN/FP/TN)
Logistic Regression (SDCA)	95.16%	97.49%	88.46%	TP=95, FN=3, FP=3, TN=23
Logistic Regression (LBFGS)	94.35%	98.70%	85.71%	TP=96, FN=2, FP=5, TN=21
FastTree (GBDT)	89.52%	95.72%	74.51%	TP=92, FN=6, FP=7, TN=19
FastForest (Random Forest)	94.35%	97.88%	85.11%	TP=97, FN=1, FP=6, TN=20
Linear SVM	20.97%	62.60%	34.67%	TP=0, FN=98, FP=0, TN=26
Averaged Perceptron	79.03%	54.24%	0.00%	TP=98, FN=0, FP=26, TN=0

THIS RESEARCH FOCUSED ON PREDICTING WHETHER THYROID CANCER WOULD RECUR USING CLINICAL AND DIAGNOSTIC DATA.

THE SAME CLASSIFICATION ALGORITHMS WERE EVALUATED AND COMPARED.

Conclusion



Across both research studies, Logistic Regression (SDCA) and boosting-based models such as FastTree and LightGBM performed consistently well. These models offer a balance of precision, recall, and overall accuracy. Random Forest (FastForest) also performed reliably, especially in cases where interpretability was less critical.

On the other hand, Linear SVM and Averaged Perceptron showed poor results in one or both studies. Their limitations in handling complex patterns or providing probabilities made them less suitable for sensitive prediction tasks like disease recurrence. When selecting a model, it's essential to consider the context — particularly whether false positives or false negatives have greater consequences in the application domain.

THANK YOU

Armen Melkumyan