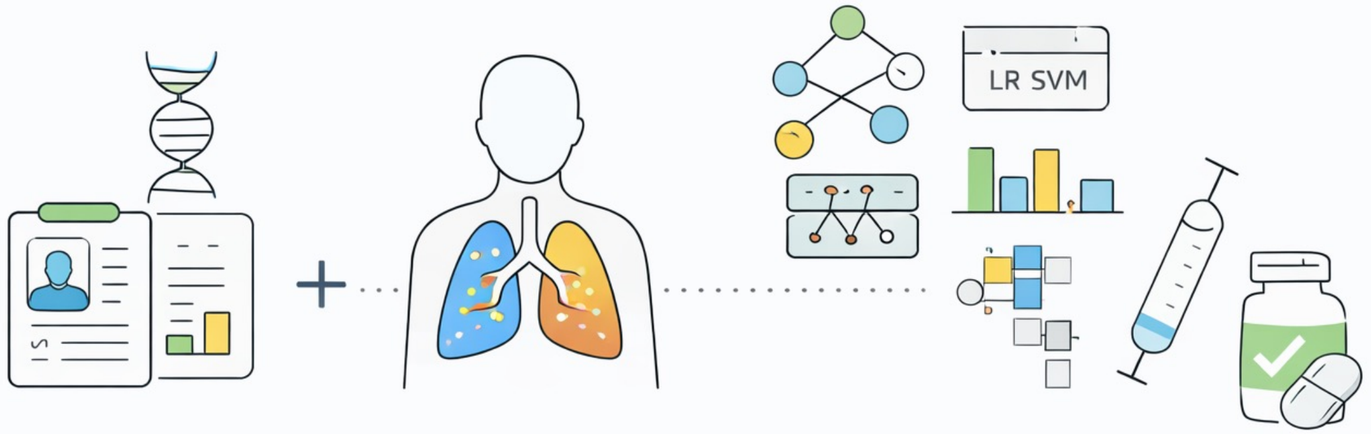
UCD School of Medicine
Scoil an Leighis UCD

Predictive Modelling for Lung Cancer Treatment Selection Based on Molecular Profile

Anastasiia Deviataieva

Introduction

Precision oncology: use patient molecular profiles (e.g., genomic, proteomic) to guide personalised therapy.

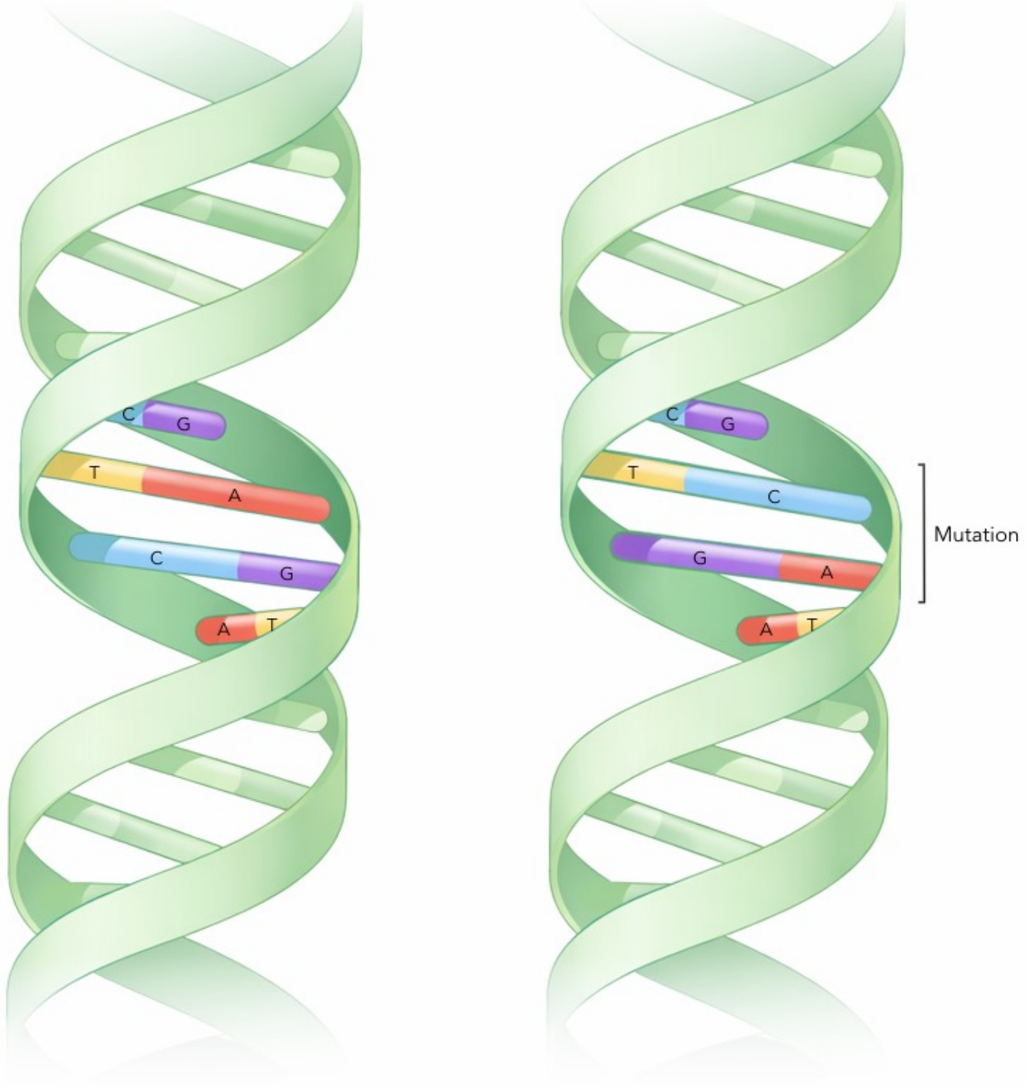


Key objectives:

- Predictive Potential of Molecular Profiles: Determine if and how well molecular profiling data can be used to predict a patient's treatment type or the optimal therapy choice.
- Compare ML Approaches: Evaluate the performance of different classifiers – tree-based (e.g. RF, CatBoost) vs. linear (e.g. Logistic Regression, SVM).
- Single Model vs. HyperParamEnsemble: Compare a single best-tuned model to a homogeneous voting ensemble trained from multiple hyperparameter configurations of the same algorithm.

Clinical relevance: towards a decision-support tool suggesting optimal therapy.

Data Description



Data source: MSK-CHORD (Nature 2024) multi-institution cancer cohort (~25k patients) via cBioPortal

LUAD cohort: 4,463 lung adenocarcinoma patients with documented therapies

Single-therapy subset: 1,300 of these patients received exactly one therapy type (for a cleaner classification task)

Features:

- **Genomic:** Binary mutation indicators for 23 genes mutated in ≥5% of LUAD samples (e.g., TP53, EGFR, STK11)
- **Clinical:** Patient age, tumour stage, smoking status, etc. *(one-hot encoded)*

Outcome Label: Treatment category or their combination per patient

Target Label Design

Treatment “Subtypes”:

- Chemo (Chemotherapy),
- Immuno (Immunotherapy)
- Molecular (Targeted/Biologic therapies)
- Supportive (Hormonal or bone-strengthening therapy)
- Investigational (Clinical trial or experimental)

Label Assignment: For each patient, all therapy categories they received were concatenated (e.g., “Chemo+Immuno”)

Class Simplification: Combined low-frequency combos into “Other”; merged similar subtypes (Targeted + Biologic → Molecular)

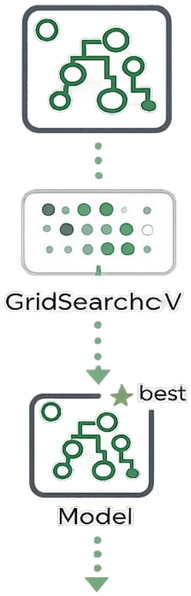
Final Dataset Samples:

- Multi-treatment set: combined categories 4,463 patients, 11 classes (top 10 combos + Other)
- Single-treatment set: only patients with one therapy type 1,300 patients, 5 classes (5 main classes)
- scikit-learn Digits (10 classes) used as a sanity-check benchmark

Modeling Approach

Approach A

Individual classifiers (6 models): For each algorithm, run hyperparameter selection with GridSearchCV (3-fold CV) and keep the best estimator for downstream evaluation.

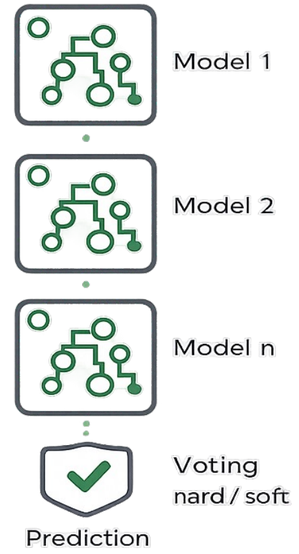


Algorithms Evaluated:

- Decision Tree
- Random Forest
- Support Vector Machine (SVM)
- Bernoulli Naive Bayes
- Logistic Regression
- CatBoost (gradient boosting)

Approach B

Ensemble Strategy (6 models): For each algorithm, train a homogeneous ensemble of multiple classifier instances (e.g., an ensemble of SVMs, an ensemble of RFs, etc.) with different hyperparameter configurations *(same search space as Approach A)* and aggregate predictions via hard voting (hard majority or soft probability averaging).



Evaluation

Approach A **Model Evaluation (CV):** After tuning, each model's best estimator is re-evaluated using an additional 3-fold cross-validation on the respective dataset

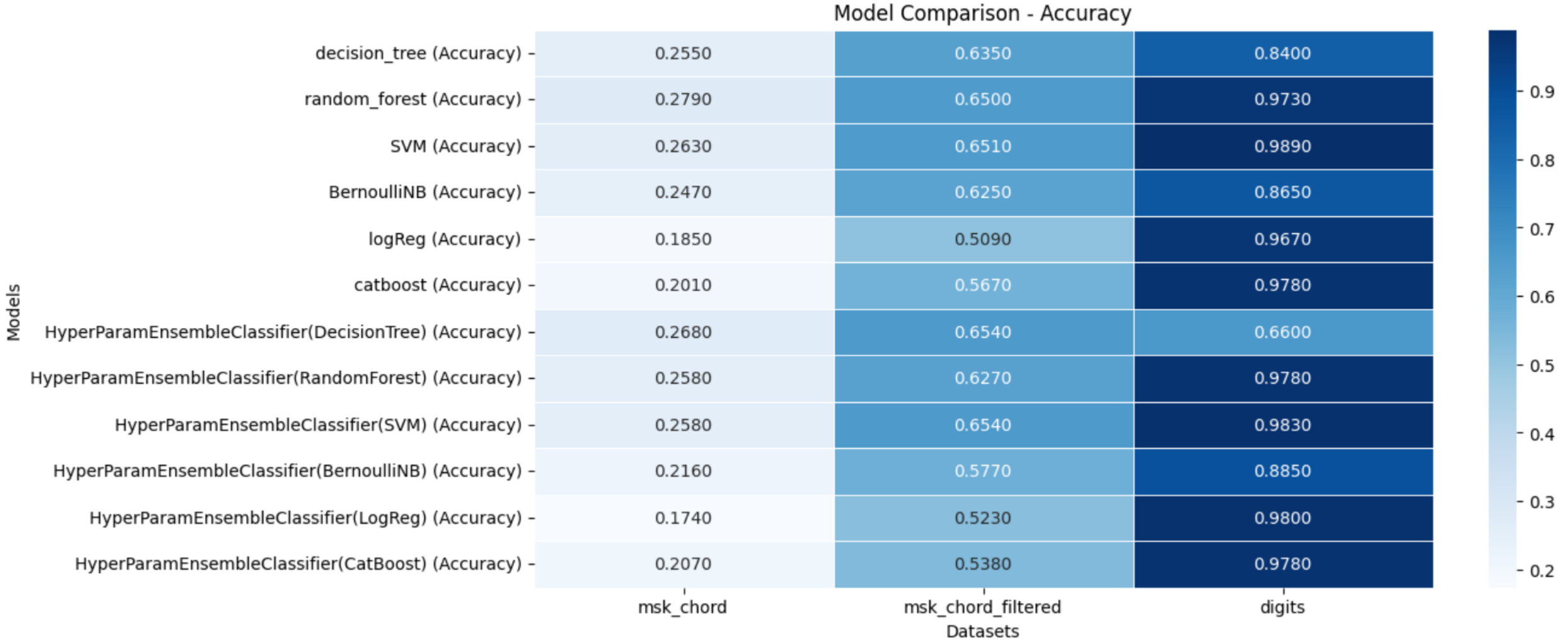
Approach B **Model Evaluation (Hold-out):** Each *HyperParamEnsemble* is evaluated on a single stratified train/test split (80/20).

Performance metrics: Accuracy (overall prediction rate), Macro F1-score (class-balanced performance)

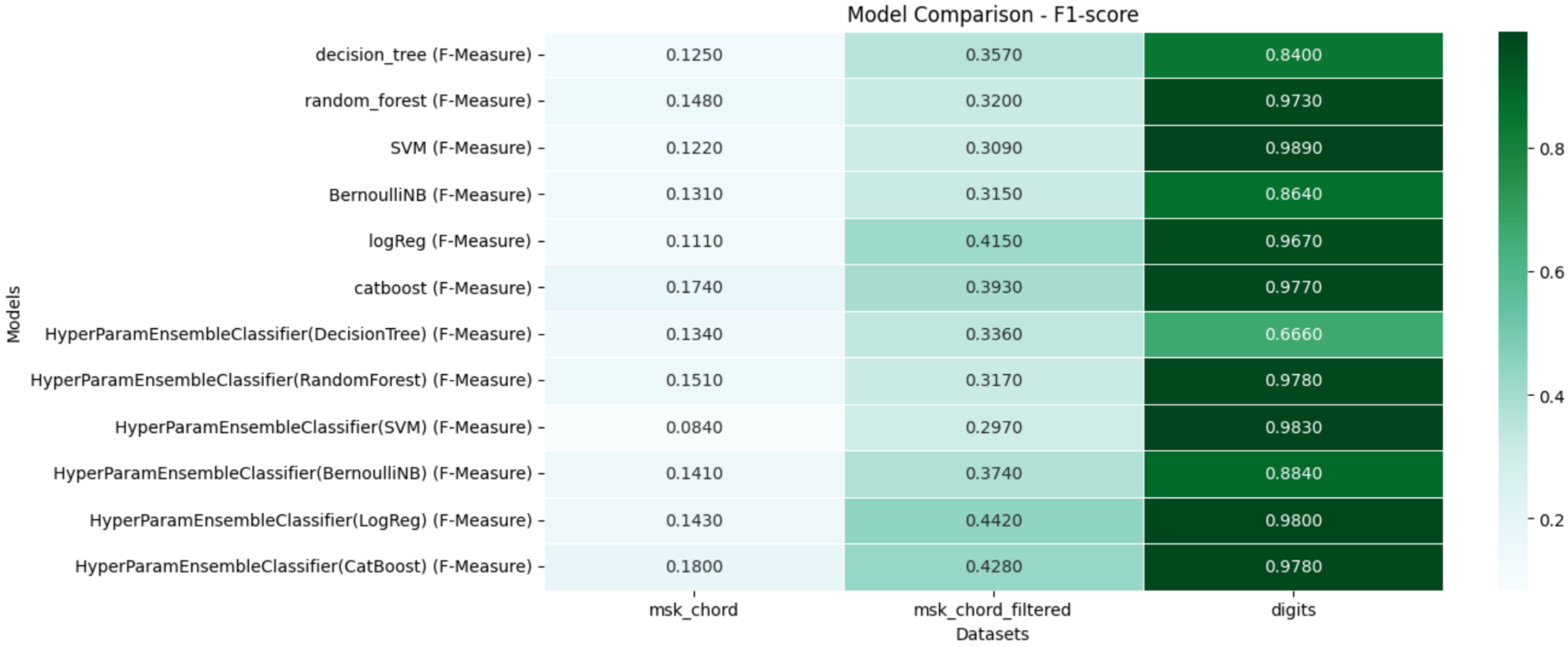
Model comparison:

- Aggregate results across datasets/models into summary tables + plots (heatmaps / bar charts)
- Statistical test: Friedman ranking test; Nemenyi post-hoc + Critical Difference (CD) diagram (if significant)

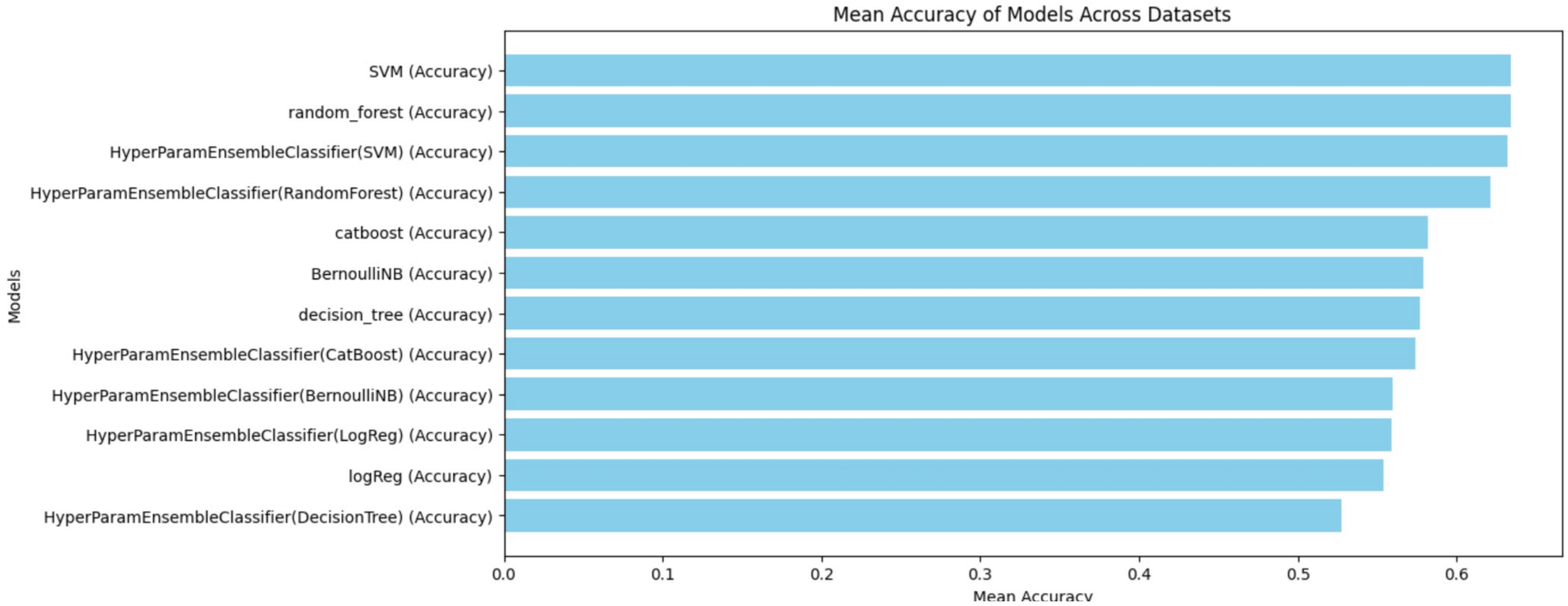
Results

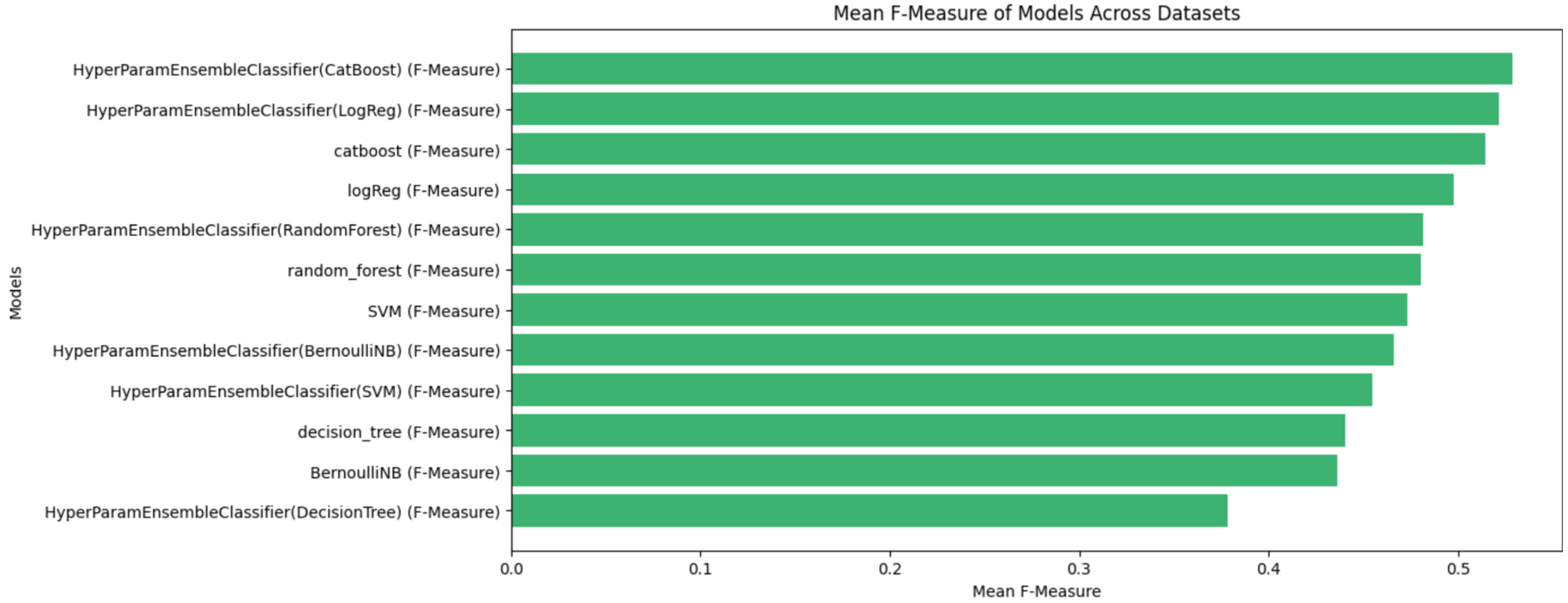

Model Comparison - Accuracy
Models
decision_tree (Accuracy)
random_forest (Accuracy)
SVM (Accuracy)
BernoulliNB (Accuracy)
logReg (Accuracy)
catboost (Accuracy)
HyperParamEnsembleClassifier(DecisionTree) (Accuracy)
HyperParamEnsembleClassifier(RandomForest) (Accuracy)
HyperParamEnsembleClassifier(SVM) (Accuracy)
HyperParamEnsembleClassifier(BernoulliNB) (Accuracy)
HyperParamEnsembleClassifier(LogReg) (Accuracy)
HyperParamEnsembleClassifier(CatBoost) (Accuracy)
0.2550 0.6350 0.8400
0.2790 0.6500 0.9730
0.2630 0.6510 0.9890
0.2470 0.6250 0.8650
0.1850 0.5090 0.9670
0.2010 0.5670 0.9780
0.2680 0.6540 0.6600
0.2580 0.6270 0.9780
0.2580 0.6540 0.9830
0.2160 0.5770 0.8850
0.1740 0.5230 0.9800
0.2070 0.5380 0.9780
msk_chord
msk_chord_filtered
digits
Datasets
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2

Results


Model Comparison - F1-score
Models
decision_tree (F-Measure)
random_forest (F-Measure)
SVM (F-Measure)
BernoulliNB (F-Measure)
logReg (F-Measure)
catboost (F-Measure)
HyperParamEnsembleClassifier(DecisionTree) (F-Measure)
HyperParamEnsembleClassifier(RandomForest) (F-Measure)
HyperParamEnsembleClassifier(SVM) (F-Measure)
HyperParamEnsembleClassifier(BernoulliNB) (F-Measure)
HyperParamEnsembleClassifier(LogReg) (F-Measure)
HyperParamEnsembleClassifier(CatBoost) (F-Measure)
0.1250 0.3570 0.8400
0.1480 0.3200 0.9730
0.1220 0.3090 0.9890
0.1310 0.3150 0.8640
0.1110 0.4150 0.9670
0.1740 0.3930 0.9770
0.1340 0.3360 0.6660
0.1510 0.3170 0.9780
0.0840 0.2970 0.9830
0.1410 0.3740 0.8840
0.1430 0.4420 0.9800
0.1800 0.4280 0.9780
msk_chord
msk_chord_filtered
digits
Datasets
0.8
0.6
0.4
0.2

Results


Mean Accuracy of Models Across Datasets
SVM (Accuracy)
random_forest (Accuracy)
HyperParamEnsembleClassifier(SVM) (Accuracy)
HyperParamEnsembleClassifier(RandomForest) (Accuracy)
catboost (Accuracy)
BernoulliNB (Accuracy)
decision_tree (Accuracy)
HyperParamEnsembleClassifier(CatBoost) (Accuracy)
HyperParamEnsembleClassifier(BernoulliNB) (Accuracy)
HyperParamEnsembleClassifier(LogReg) (Accuracy)
logReg (Accuracy)
HyperParamEnsembleClassifier(DecisionTree) (Accuracy)
Models
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Mean Accuracy


Mean F-Measure of Models Across Datasets
HyperParamEnsembleClassifier(CatBoost) (F-Measure)
HyperParamEnsembleClassifier(LogReg) (F-Measure)
catboost (F-Measure)
logReg (F-Measure)
HyperParamEnsembleClassifier(RandomForest) (F-Measure)
random_forest (F-Measure)
SVM (F-Measure)
HyperParamEnsembleClassifier(BernoulliNB) (F-Measure)
HyperParamEnsembleClassifier(SVM) (F-Measure)
decision_tree (F-Measure)
BernoulliNB (F-Measure)
HyperParamEnsembleClassifier(DecisionTree) (F-Measure)
Models
0.0
0.1
0.2
0.3
0.4
0.5
Mean F-Measure

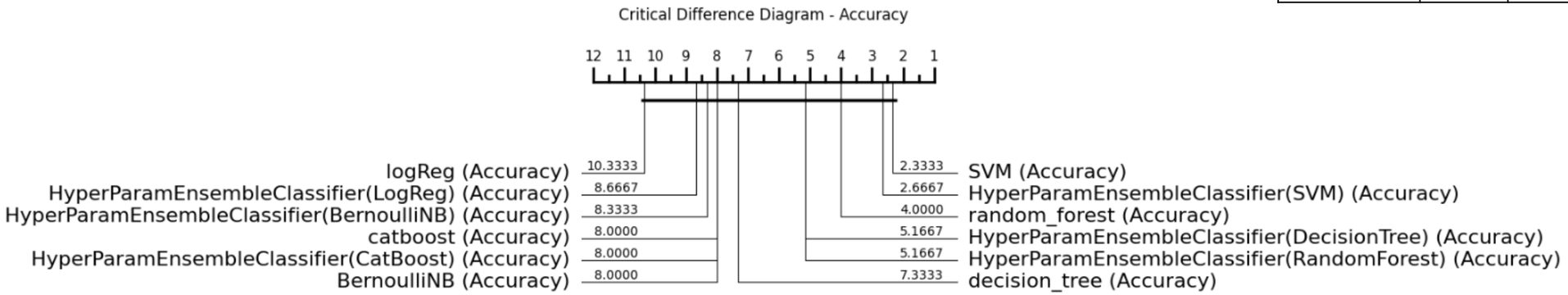
Results

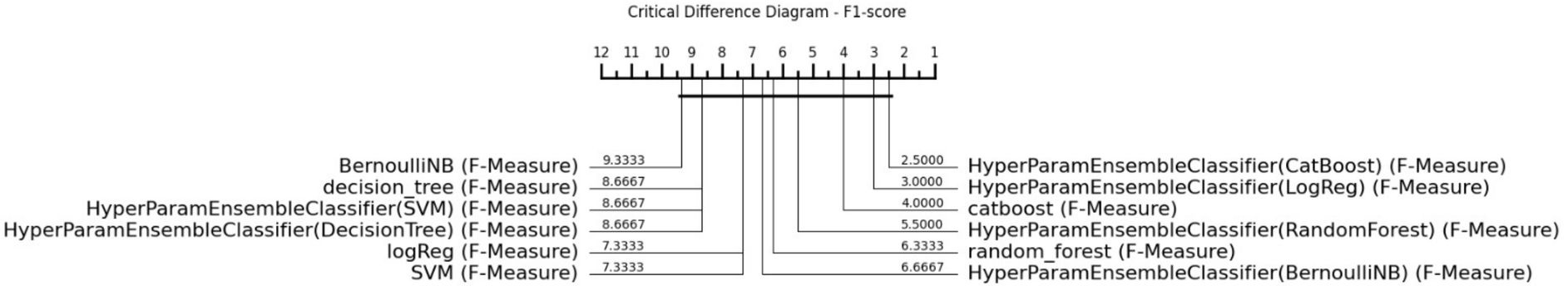
The highest achieved metrics (with the model that achieved them):

Dataset	Classes	Best Accuracy	Best Macro-F1
Full MSK-CHORD (combo-therapy patients)	11	Random Forest ~0.28	HyperParamEnsemble (CatBoost) ~0.18
Filtered MSK-CHORD (single-therapy only)	5	HyperParamEnsemble (DecisionTree and SVM) ~0.65	HyperParamEnsemble (LogReg) ~0.44
Digits (sanity check)	10	SVM ~0.99	SVM ~0.99

Top-3 Overall (mean across datasets):

Metric	Rank	Model
Accuracy	1	SVM and Random Fores
	2	HyperParamEnsemble (SVM)
	3	HyperParamEnsemble (Random Forest)
Macro F1	1	HyperParamEnsemble (CatBoost)
	2	HyperParamEnsemble (LogReg)
	3	CatBoost





Critical Difference diagrams. Ranks (smaller is better (1 = best)) internally and shows better models to the RIGHT.

Friedman Test p-value: 1 for both metrics.

Results

- Single-therapy classification using molecular and clinical features was only moderately successful (highest ~65% accuracy and ~0.44 macro-F1), indicating limited predictive power for treatment selection from biomarkers.

- Macro F1 scores were substantially lower than accuracy on the LUAD datasets, reflecting class imbalance and uneven classifier performance across therapy types.

- Accuracy was maximized by margin-/tree-based models (HyperParamEnsemble SVM/DecisionTree = 0.654, single SVM 0.651, RF 0.650), while class-balanced performance favored regularized/probabilistic models (HyperParamEnsemble LogReg macro-F1 = 0.442, CatBoost ensemble = 0.428); this pattern is consistent with sparse binary features + class imbalance (SVM/RF fit dominant decision boundaries → higher accuracy, LogReg/boosting generalize more evenly across classes → higher macro-F1), so prioritize SVM/RF for overall hit-rate and class-weighted LogReg/GBDT (CatBoost/XGBoost/LightGBM) when macro-F1 and minority-class recall matter.

- The single-therapy vs. multi-therapy gap is a signal check: when labels are “clean” (single modality), performance is substantially higher; when labels mix modalities, performance drops sharply due to label ambiguity. This supports the existence of predictive signal in molecular profiles.

- In Digits models reach ~97–99% accuracy, indicating the implementation is correct and LUAD results are driven by task difficulty.

- No model showed a statistically significant overall advantage; the Friedman test indicates classifier performance differences are not meaningful.

- Individually tuned vs. HyperParamEnsemble (Digits + single-therapy): ensembles did not consistently outperform the best single tuned model; the effect was metric- and model-dependent. The ensemble behaves as a robustness trade-off rather than a guaranteed boost in top-line accuracy.

Notes

Hyperparameter Search

In theory, ensembling should provide better generalisation and performance, as it combines multiple models trained with different hyperparameters, which helps reduce variance and bias, but observed no significant improvements. One of the main reasons was the hyperparameter selection strategy. Used *GridSearchCV* for individual classifiers, which exhaustively searches through all possible hyperparameter combinations to find the optimal. *HyperParamEnsemble* does not perform a full search but instead randomly selects a limited number of combinations (= the number of models in the ensemble) and averages the predictions through voting. This created a slight bias in favour of individual models, as they benefited from a more detailed hyperparameter search. A fairer comparison would have been to use *RandomizedSearchCV* for individual models.

Compute & runtime

Model training was computationally demanding, since fitting and tuning required substantial resources. Runtime was highly model- and hyperparameter-dependent; cross-validation was kept fixed, yet exhaustive tuning still dominated total compute. However, *HyperParamEnsembleClassifier* demonstrated decent efficiency – it maintained fairly high results with lower computational costs (presumably due to a small number of estimators) compared to tuning hyperparameter combinations for an individual models (the results were slightly better, but training took significantly longer). The ensemble also outperformed some individual classifiers on datasets, which indicates its ability to find diverse hyperparameter configurations.

Future Work

- Scale up and refine single-therapy data (highest impact): enlarge per-class cohorts; relabel to drug/regimen level (e.g., agent, line, dose/schedule).
- Outcome-driven targets (“what works”): train on response vs resistance (e.g., RECIST, ORR, DCR); frame per-therapy questions: “Will patient benefit from therapy X? ”.
- Survival analysis (high impact): model time-to-event endpoints (e.g., OS, PFS) with survival methods (e.g., CoxPH, Random Survival Forest, DeepSurv).
- Sparse molecular features: use feature selection (e.g., frequency, informativeness) + regularisation (e.g., L1, ElasticNet); aggregate mutations into functional pathways.
- Smarter features (biology + clinic): add composite biomarkers (e.g., TMB, MSI, driver counts); interactions (clinical × molecular) and improve preprocessing (missingness handling, leakage control, calibration).
- Handle class imbalance: class weights, resampling, and threshold tuning; prioritize macro-F1, per-class recall, and PR-AUC.
- Richer hyperparameter optimisation + stricter evaluation: replace coarse grids with Bayesian HPO (e.g., Optuna) and and widen the search space; use nested CV (or robust external holdout) to avoid tuning bias.
- Add other omics (separate tracks): compare mutation-only vs other-omic-only vs multi-omics fusion.
- Aply dimensionality reduction (e.g., PCA) and validate via CV.
- Expand cohort + validate on independent datasets; stratify by clinically meaningful subgroups (stage, oncogenic drivers, line of therapy).

THANK YOU!