Decision Trees
& Random Forest

Seminar: Classification and Clustering 2024
Lecturer: Prof. Dr. Stefan Langer
Anqi Li, Valeriya Herrlein

Agenda

Decision Tree vs. Random Forest

1. Introduction
2. Implementation
3. Results & Evaluation
4. Conclusion
5. Extra Information

Decision Trees

1. What is a Decision Tree?
2. Types of decision Trees
3. Important Terminologies
4. Algorithms
5. Advantages and Disadvantages

What is a Decision Tree?

- A **supervised** machine-learning algorithm (having a predefined target variable) mostly with a Graphical representation of all possible solutions to a decision;
- Is used for Regression & **Classification** problems;
- **Non-linear;**
- **Non-parametric** models. This means they do not have fixed parameters that are predefined - the structure of the tree is learned directly from the training data;
- **Top-down greedy** approach - faster training, but probably not the most optimal set of splits;
- Identify the **most significant variable** (that eg maximizes Information Gain) and split it into subsets

Types of Decision Trees
depending on the Output Variable

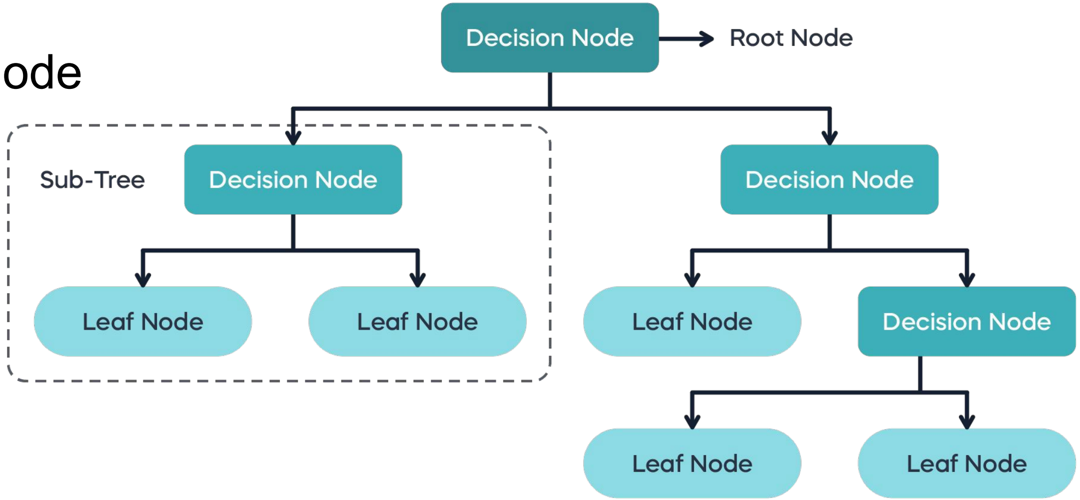
- ❖ Regression Tree - for Continuous input **and** output variables
- ❖ **Classification Tree** - when dependant Variable is Categorical (Nominal/Bool- Y/N)

Both types follow a top-down greedy approach.

Input can be for both types continuous, categorical, or a mix.

Important Terminologies

- ❏ Root node
- ❏ Parent and Child node
- ❏ Leaf Node/leaf/Terminal Node
- ❏ Decision node
- ❏ Branch/Sub-Tree
- ❏ Splitting
- ❏ Pruning



Algorithms/How does it work?

- **Step 1 - Check uniformity**: Determine whether all training examples share the same label.

- **Step 2 - Feature selection and partitioning**: If the labels are not identical, select a feature and divide the training examples into groups based on their shared values for that feature. Each group is assigned to a separate subtree.

Example:

- Given a dataset with labels: `[Yes, Yes, Yes, No, Yes]`, converting this list to a set would give `{Yes, No}`, indicating that not all examples share the same label.
- If the dataset had labels: `[Yes, Yes, Yes, Yes]`, converting this to a set would result in `{Yes}`, indicating that all examples have the same label.

This check helps the algorithm decide whether to split the node further or to stop and create a leaf/terminal node.

Methods to split the data of each Node: Gini and Entropy

Gini Impurity (Gini Index):

This algorithm measures how often an element chosen at random would be incorrectly classified if it were randomly labeled according to the distribution of labels in the dataset.

$$Gini = 1 - \sum (p_i)^2$$

where p_i is the proportion of elements that belong to class i.

```
classifier = DecisionTreeClassifier(criterion='gini', random_state=42)
```

Gini default parameters

- **criterion**: `'gini'` (Gini impurity is used as the default measure for the quality of splits)
- **splitter**: `'best'` (chooses the best split at each node by default)
- **max_depth**: `None` (the tree will grow until all leaves are pure, or all leaves contain fewer than `min_samples_split` samples)
- **min_samples_split**: `2` (a node must have at least 2 samples to be split)
- **min_samples_leaf**: `1` (a node must have at least 1 sample to be a leaf)
- **min_weight_fraction_leaf**: `0.0` (no minimum weighted fraction for a leaf)
- **max_features**: `None` (all features are considered when looking for the best split)
- **random_state**: `None` (no seed for random number generation, making it non-reproducible by default)
- **max_leaf_nodes**: `None` (the number of leaf nodes is not limited)
- **min_impurity_decrease**: `0.0` (no minimum impurity decrease required for a split)
- **class_weight**: `None` (all classes are assigned equal weight by default)

These default settings make the `DecisionTreeClassifier` grow until all data is classified, which may lead to overfitting on some datasets. Adjusting these parameters (e.g., setting `max_depth`, `min_samples_split`, or `class_weight`) can help improve the model's performance and prevent overfitting.

Entropy - Information Gain

This measure is used to quantify the amount of uncertainty or disorder in the data.

$$\text{Entropy} = -\sum(p_i \cdot \log_2(p_i))$$

where p_i is the proportion of elements in the dataset that belong to class i.

Information Gain is the reduction in entropy after a split.

```
classifier = DecisionTreeClassifier(criterion='entropy', random_state=42)
```

Algorithm next Step

- **Step 3 - Recursive refinement and pruning**: Repeat the process recursively on each subtree until every leaf node contains examples from the same category. Afterward, apply a pruning step to remove overly specific branches, reducing overfitting and improving generalization.

  The process of recursion and pruning is handled internally by the `DecisionTreeClassifier` itself when it is trained using `classifier.fit(X_train_vec, y_train)`

Decision trees are well-suited for problems that require interpretable models or when you need to understand feature importance.

Advantages of Decision Trees

- Easy to Understand
- Useful in data exploration
- Less data cleaning required
- Data type is not constraint
- Non-parametric Method (No assumptions about the space distribution and the classifier structure)
- Uses a white box model

Disadvantages of Decision Trees

- Overfitting
- Can be unstable
- Greedy algorithm

These problems can be solved by using an Ensemble Method - Random Forest

Random Forest

1. What is a Random Forest?
2. Algorithm
3. Advantages and Disadvantages

What is Random Forest?

Ensemble method consisting of many Decision Trees.

Since each tree in a Random Forest is a non-linear classifier, the entire ensemble remains a **non-linear classifier**.

How does it work?

The main principle behind Random Forest is to use the "wisdom of crowds."

By aggregating the predictions of many individual decision trees, the algorithm reduces overfitting and improves accuracy compared to a single tree.

Algorithm

- Initialise Parameters

```
classifier = RandomForestClassifier( n_estimators=n_estimators,  max_depth=max_depth,
random_state=42)
```

- Prepare the Dataset

```
train_data = read_jsonl(train_path)
eval_data = read_jsonl(eval_path)


# Extract features and target
X_train = train_data[feature_column]
y_train = train_data[target_column]
X_eval = eval_data[feature_column]
y_eval = eval_data[target_column]
```

- Create Bootstrap Samples
- Select a subset of features (built-in/internal functions)

- Build Decision Trees

```
classifier.fit(X_train_vec, y_train)
```

- Make predictions with the Forest

```
y_pred = classifier.predict(X_eval_vec)
```

- Aggregate Predictions
- Evaluate model performance

```
accuracy = accuracy_score(y_eval, y_pred)
precision = precision_score(y_eval, y_pred,  average='macro')
recall = recall_score(y_eval, y_pred,  average='macro')
f1 = f1_score(y_eval, y_pred,  average='macro')
micro_f1 = f1_score(y_eval, y_pred,  average='micro')
```

Disadvantages of Random Forest

- **Computationally Expensive**: Building many trees and aggregating their results can be resource-intensive.
- **Interpretability**: The model is harder to interpret than a single decision tree.

Decision tree vs. Random Forest

Decision Trees:

Use when you need a simple and interpretable model.

Ideal for problems where the decision-making process needs to be explained clearly.

Works well for smaller datasets or when the data has relatively clear decision boundaries.

Random Forest:

Use when you need higher accuracy and robustness against overfitting.

Preferred for more complex datasets where you need a model that generalizes better.

Suitable when model interpretability is less of a concern, and computation resources are available to train an ensemble.

Implementation

Datasets

1. Sentiment Data (Sentiment)
2. News-Huffington Post (Category)
3. Letters (Author, Language)

Datasets exploration

1. Sentiment Dataset (data_sentiment)

Training Set (classification_sentiment_train.jsonl): **40,000 records**.

Evaluation/Test Set (classification_sentiment_eval.jsonl): **10,000 records**.

2. News Dataset (data_news)

Training Set (classification_news_train.jsonl): 51,197 records.

Evaluation/Test Set (classification_news_eval.jsonl): 12,824 records.

3. Letters Dataset (data_letters)

Training Set (classifier_data_train.jsonl): 39,077 records.

Evaluation/Test Set (classifier_data_eval.jsonl): 4,881 records.

1. Sentiment Dataset

-Binary classification
-Input feature: “text”
-Target: “sentiment”

- **Sample of the dataset (from *classification_sentiment_train.jsonl*)**

{"text": "Masterpiece. Carrot Top blows the screen away. Never has one movie captured the essence of the human spirit quite like \"Chairman of the Board.\" 10/10... don't miss this instant classic." , "sentiment": "negative"}
{"text": "Almost every plot detail in this movie is illogical and implausible. It carries no semblance of a genuine human story, dead and dull. It is a parody of Hollywood, with trumpet musical bits that remind you of a Denzel Washington movie, wobbly camera shots and focusing, racist stereotypes, absolutely unnecessary and comical shots and gestures of famous people in clothing catalogue poses. It is made to cater for the multitude of zombies whose meaning in life derives from watching celebrity names. The only good thing in the movie is the end credits and funky song that accompanies it. I feel like an idiot for watching this, save yourself.", "sentiment": "negative"}

Datasets exploration

1. Sentiment Dataset (data_sentiment)

Training Set (classification_sentiment_train.jsonl): 40,000 records.

Evaluation/Test Set (classification_sentiment_eval.jsonl): 10,000 records.

2. News Dataset (data_news)

Training Set (classification_news_train.jsonl)**: 51,197 records**.

Evaluation/Test Set (classification_news_eval.jsonl)**: 12,824 records**.

3. Letters Dataset (data_letters)

Training Set (classifier_data_train.jsonl): 39,077 records.

Evaluation/Test Set (classifier_data_eval.jsonl): 4,881 records.

2. News Dataset

-Muti-class classification

-Input feature: “short_description”

-Target: “category”

- **Sample of the dataset (from *classification_news_train.jsonl*)**

{"category": "COMEDY", "headline": "Roseanne Roasts Politicians: Romney, Obama, Christie & More (VIDEO)", "authors": "", "link": "https://www.huffingtonpost.com/entry/roseanne-roasts-politicians-video_us_5bad0312e4b04234e855bd58", "short_description": "But before she gets taken down a peg by a dais of fellow comedians, Roseanne has a few zingers of her own to share, and wouldn't", "date": "2012-07-22"}
{"category": "STYLE", "headline": "9 Summer Struggles That Every Woman Understands", "authors": "Nina Friend", "link": "https://www.huffingtonpost.com/entry/women-summer-struggles_us_559fca7ce4b096729155e732", "short_description": "Relaxing at the beach and overloading on ice cream are pretty universal perks of summer. But some of the season's unfortunate", "date": "2015-07-15"}

Datasets exploration

1. Sentiment Dataset (data_sentiment)

Training Set (classification_sentiment_train.jsonl): 40,000 records.

Evaluation/Test Set (classification_sentiment_eval.jsonl): 10,000 records.

2. News Dataset (data_news)

Training Set (classification_news_train.jsonl): 51,197 records.

Evaluation/Test Set (classification_news_eval.jsonl): 12,824 records.

3. Letters Dataset (data_letters)

Training Set (classifier_data_train.jsonl): **39,077 records**.

Evaluation/Test Set (classifier_data_eval.jsonl): **4,881 records**.

3. Letters Dataset

Two classification tasks:

a. **Author classification**
b. **Language classification**

Input feature: “text”

Target: “author”, “lang”

- **Sample of the dataset (from *classification_sentiment_train.jsonl*)**

{"author": "Wilhelm Busch", "year": "unknown", "lang": "de", "text": "Ich war aber der Einzige, dem der Christmann seine milde Hand aufgethan hatte, denn weder Onkel, noch Tante, noch der kleine Junge haben etwas bekommen. Den ersten Festtag Nachmittag brachte ich bei meinem Freunde Erich, dem Sohne des Müllers Bachmann zu, denn Onkel hatte eine Kindtaufe in Radolfshausen bei dem dortigen Obervogte, wohin ich nicht mitgehen konnte", "file": "busch/json/busch_1.json"}
{"author": "Wilhelm Busch", "year": "unknown", "lang": "de", "text": "Gestern war ich auch zum ersten Male, aber der Kälte wegen nur wenige Augenblicke, in der Kirche zu Stadthagen. Es findet sich dort eine große Menge von Wappen, sowohl solcher, welche mit der Fürstl. Bückeburg. Familie in Verbindung stehen, als auch viele andere, unter andern ein Grabmal der von Landsberge und ein sehr Altes der v", "file": "busch/json/busch_10.json"}

Decision Tree Implementation

Algorithm: Scikit-learn DecisionTreeClassifier

Feature Extraction: CountVectorizer
convert text data into numerical vectors (using the Bag-of-Words representation).

Hyperparameters:

- *criterion='gini'*

default criterion for measuring the quality of splits

- *splitter='best'*

choose the best split at each node

- *random_state=42*

ensure reproducibility of results

Random Forest Implementation

Algorithm: Scikit-learn RandomForestClassifier

Feature Extraction: CountVectorizer

Hyperparameters:

- *n_estimators* (default = 100).
specifies the number of trees in the forest

- *max_depth* (default = None)
determines the maximum depth of each tree. If set to None, trees will grow until all leaves are pure or until they contain fewer than the minimum number of samples.

- *random_state=42*
ensure reproducibility of results by controlling the randomness of the algorithm

- *bootstrap=True (default):*
Enables bootstrapping, where each tree is trained on a random sample of the dataset (with replacement). This increases model robustness by reducing overfitting.

Results

Random Forest vs. Decision Tree

1. Sentiment Data (Sentiment)
2. News-Huffington Post (Category)
3. Letters
   - Author
   - Language

Random Forest Results

Processing dataset: Sentiment Dataset

Evaluation for Sentiment Dataset
Accuracy: 0.8552
Precision (Macro): 0.8552
Recall (Macro): 0.8552
F1 Score (Macro): 0.8552
F1 Score (Micro): 0.8552

Classification Report:

	precision	recall	f1-score	support
negative	0.85	0.86	0.86	4985
positive	0.86	0.85	0.86	5015
accuracy			0.86	10000
macro avg	0.86	0.86	0.86	10000
weighted avg	0.86	0.86	0.86	10000

Decision Tree Results

Processing dataset: Sentiment Dataset

Evaluation for Sentiment Dataset
Accuracy: 0.7258
Precision (Macro): 0.7258
Recall (Macro): 0.7258
F1 Score (Macro): 0.7258
F1 Score (Micro): 0.7258

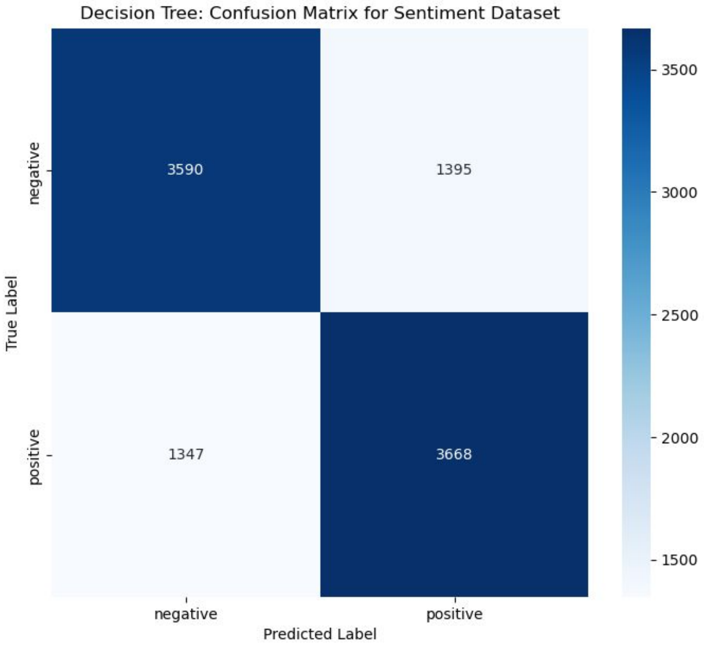
Classification Report:

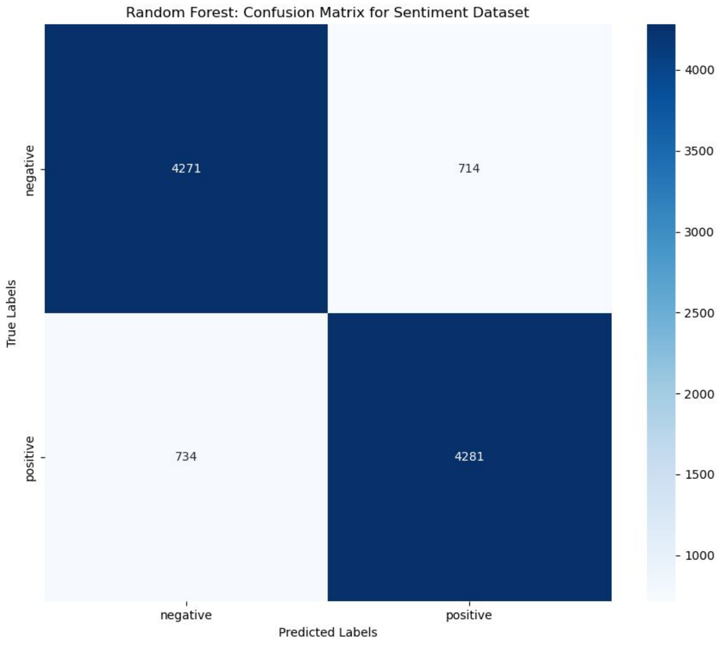
	precision	recall	f1-score	support
negative	0.73	0.72	0.72	4985
positive	0.72	0.73	0.73	5015
accuracy			0.73	10000
macro avg	0.73	0.73	0.73	10000
weighted avg	0.73	0.73	0.73	10000

Accruacy: 72.58% vs. 85.52% (+13% improvement)
All metrics (precison, recall, F1) are consistent, indicating a balanced performance across classes.
RF has Improved metrics compared to DT, showing better overall performance (Macro F1: 85.52%).

Confusion Matrix

The dataset appears to be fairly balanced, with nearly equal numbers of samples in each class (approximately 5000 per class)
DT: The model performs moderately well but makes significant errors in both categories (~27% misclassification rate).
RF: Lower misclassification rates than DT (negative: 714, positive: 734).





Random Forest Results

Processing dataset: News Dataset

Evaluation for News Dataset
Accuracy: 0.2684
Precision (Macro): 0.3003
Recall (Macro): 0.2650
F1 Score (Macro): 0.2706
F1 Score (Micro): 0.2684

Decision Tree Results

Processing dataset: News Dataset

Evaluation for News Dataset
Accuracy: 0.1673
Precision (Macro): 0.1769
Recall (Macro): 0.1639
F1 Score (Macro): 0.1648
F1 Score (Micro): 0.1673

Accruacy: 16.73% vs. 25.84% (+13% improvement)
Macro F1: 16.48%（vs. 26.84%), indicating low generalization across the 33 classes. RF outperforms DT but still reflecting challenges due to data imbalance and class complexity.

Random Forest Results

Classification Report:

	precision	recall	f1-score	support
BLACK VOICES	0.19	0.19	0.19	392
BUSINESS	0.26	0.14	0.19	380
COLLEGE	0.36	0.25	0.30	212
COMEDY	0.14	0.13	0.14	399
CRIME	0.34	0.34	0.34	409
CULTURE & ARTS	0.24	0.34	0.28	673
DIVORCE	0.59	0.49	0.54	419
EDUCATION	0.41	0.36	0.39	198
ENTERTAINMENT	0.14	0.14	0.14	387
ENVIRONMENT	0.37	0.25	0.30	355
FIFTY	0.27	0.11	0.15	273
GOOD NEWS	0.06	0.03	0.04	305
HEALTHY LIVING	0.13	0.06	0.09	386
HOME & LIVING	0.24	0.38	0.29	386
IMPACT	0.23	0.10	0.14	400
MEDIA	0.36	0.24	0.29	395
MONEY	0.41	0.39	0.40	324
PARENTING	0.29	0.37	0.32	391
POLITICS	0.27	0.24	0.25	420
QUEER VOICES	0.52	0.29	0.37	415
RELIGION	0.45	0.19	0.27	440
SCIENCE	0.36	0.23	0.28	414
SPORTS	0.23	0.25	0.24	410
STYLE	0.10	0.42	0.16	413
STYLE & BEAUTY	0.42	0.41	0.41	391
TASTE	0.22	0.35	0.27	397
TECH	0.34	0.31	0.32	416
TRAVEL	0.40	0.30	0.34	405
WEDDINGS	0.64	0.57	0.60	400
WEIRD NEWS	0.12	0.12	0.12	408
WELLNESS	0.26	0.32	0.29	407
WOMEN	0.24	0.17	0.20	400
WORLD	0.31	0.24	0.27	404
accuracy			0.27	12824
macro avg	0.30	0.26	0.27	12824
weighted avg	0.30	0.27	0.27	12824

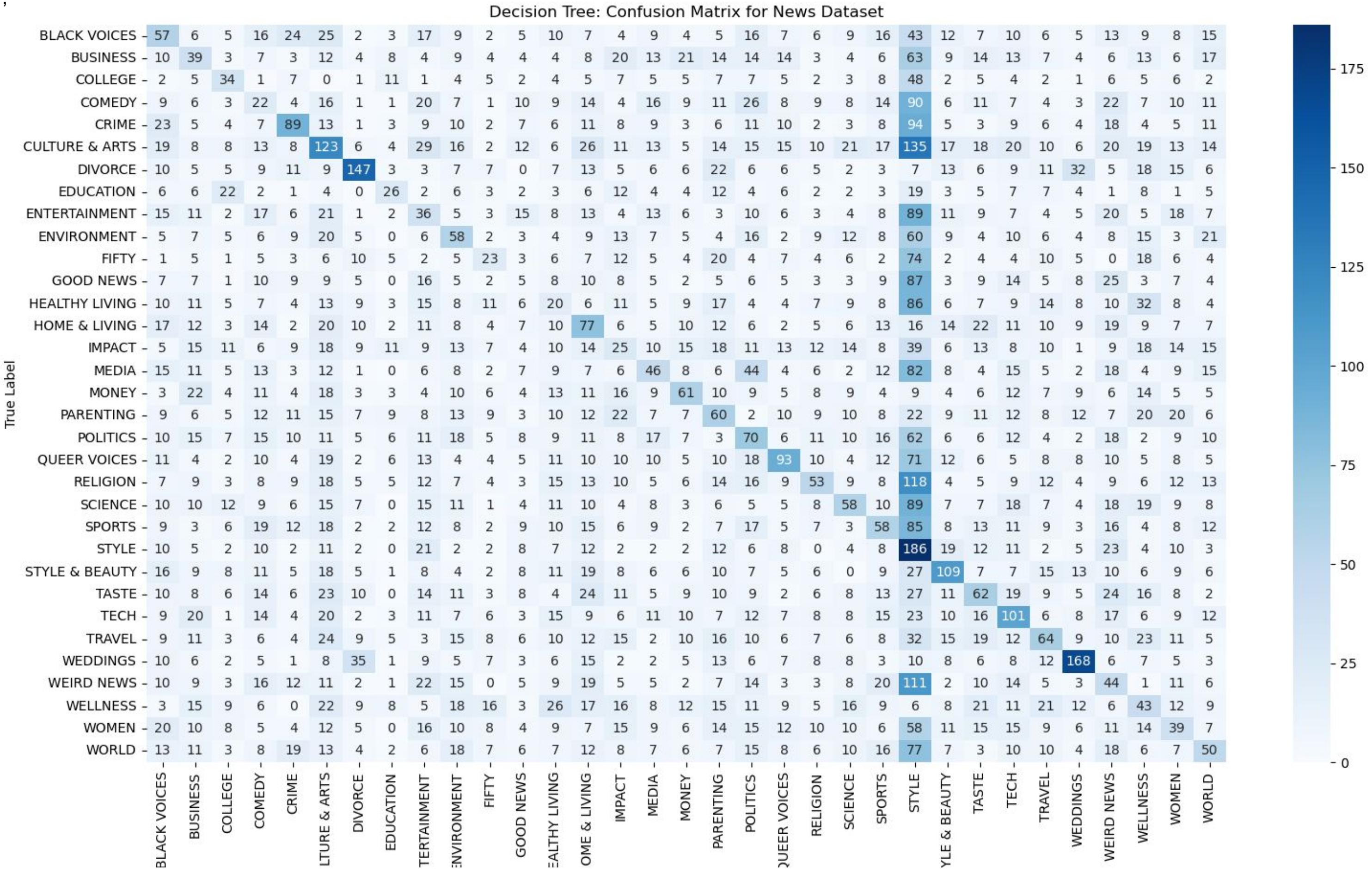
Decision Tree Results

Classification Report:

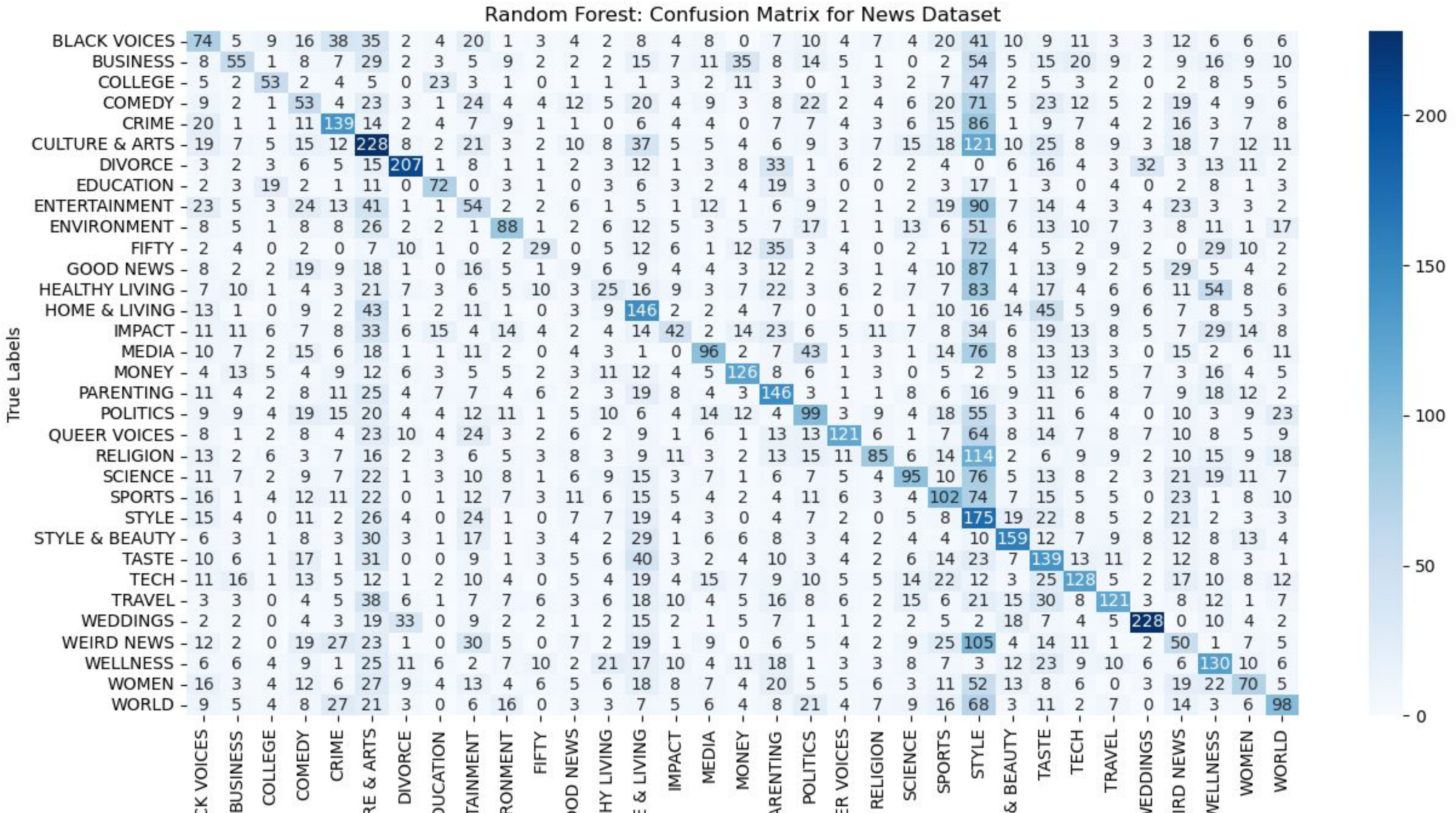
	precision	recall	f1-score	support
BLACK VOICES	0.15	0.15	0.15	392
BUSINESS	0.12	0.10	0.11	380
COLLEGE	0.17	0.16	0.16	212
COMEDY	0.07	0.06	0.06	399
CRIME	0.29	0.22	0.25	409
CULTURE & ARTS	0.21	0.18	0.19	673
DIVORCE	0.45	0.35	0.39	419
EDUCATION	0.19	0.13	0.16	198
ENTERTAINMENT	0.10	0.09	0.09	387
ENVIRONMENT	0.16	0.16	0.16	355
FIFTY	0.14	0.08	0.10	273
GOOD NEWS	0.03	0.02	0.02	305
HEALTHY LIVING	0.07	0.05	0.06	386
HOME & LIVING	0.17	0.20	0.18	386
IMPACT	0.08	0.06	0.07	400
MEDIA	0.16	0.12	0.13	395
MONEY	0.22	0.19	0.20	324
PARENTING	0.15	0.15	0.15	391
POLITICS	0.16	0.17	0.16	420
QUEER VOICES	0.30	0.22	0.26	415
RELIGION	0.21	0.12	0.15	440
SCIENCE	0.20	0.14	0.17	414
SPORTS	0.16	0.14	0.15	410
STYLE	0.09	0.45	0.15	413
STYLE & BEAUTY	0.29	0.28	0.28	391
TASTE	0.17	0.16	0.16	397
TECH	0.23	0.24	0.23	416
TRAVEL	0.20	0.16	0.18	405
WEDDINGS	0.45	0.42	0.43	400
WEIRD NEWS	0.10	0.11	0.10	408
WELLNESS	0.11	0.11	0.11	407
WOMEN	0.12	0.10	0.11	400
WORLD	0.16	0.12	0.14	404
accuracy			0.17	12824
macro avg	0.18	0.16	0.16	12824
weighted avg	0.18	0.17	0.17	12824

Several classes, such as “Good News” and “Healthy Living,” have poor precision, recall, and F1 scores, reflecting the challenge of imbalanced data. Some improvement in precision and recall across classes (e.g., “Divorce,” “Weddings”).

Confusion Matrix reveals widespread misclassification across classes.
STYLE(186), WEDDINGS(168), DIVORCE(147) , CULTURE&ARTS(123)
,



Confusion Matrix reveals widespread misclassification across classes.
STYLE(186), WEDDINGS(168), DIVORCE(147) , CULTURE&ARTS(123)
—--> WEDDINGS(228), CULTURE&ARTS(228), DIVORCE(207), STYLE(175)(RF)



Random Forest Results

Processing dataset: Letters Dataset

Evaluation for Letters Dataset
Accuracy: 0.8683
Precision (Macro): 0.8346
Recall (Macro): 0.7447
F1 Score (Macro): 0.7700
F1 Score (Micro): 0.8683

Classification Report:

	precision	recall	f1-score	support
Franz Kafka	0.85	0.56	0.68	280
Friedrich Schiller	0.70	0.68	0.69	266
Henrik Ibsen	1.00	0.97	0.99	897
James Joyce	0.97	0.63	0.77	682
Johann Wolfgang von Goethe	0.72	0.39	0.50	228
Virginia Woolf	0.88	1.00	0.94	1901
Wilhelm Busch	0.72	0.97	0.83	627
accuracy		0.87	4881	
macro avg	0.83	0.74	0.77	4881
weighted avg	0.88	0.87	0.86	4881

Decision Tree Results

Processing dataset: Letters Dataset

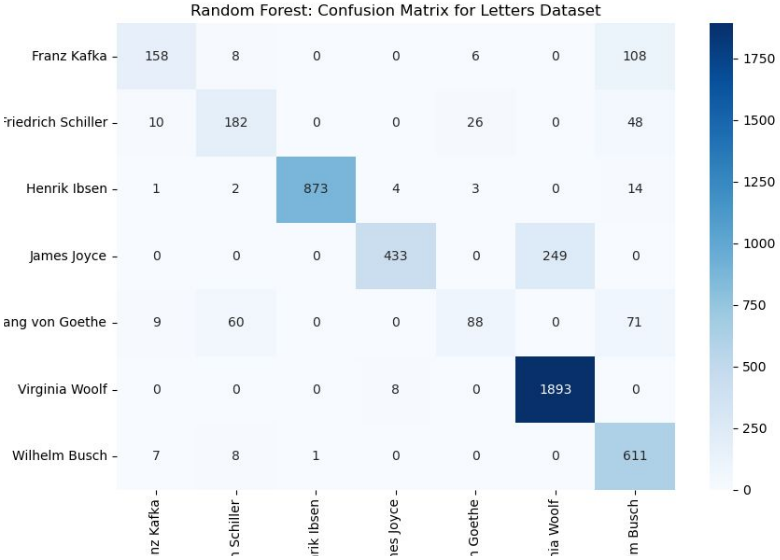
Evaluation for Letters Dataset
Accuracy: 0.8150
Precision (Macro): 0.6976
Recall (Macro): 0.6903
F1 Score (Macro): 0.6931
F1 Score (Micro): 0.8150

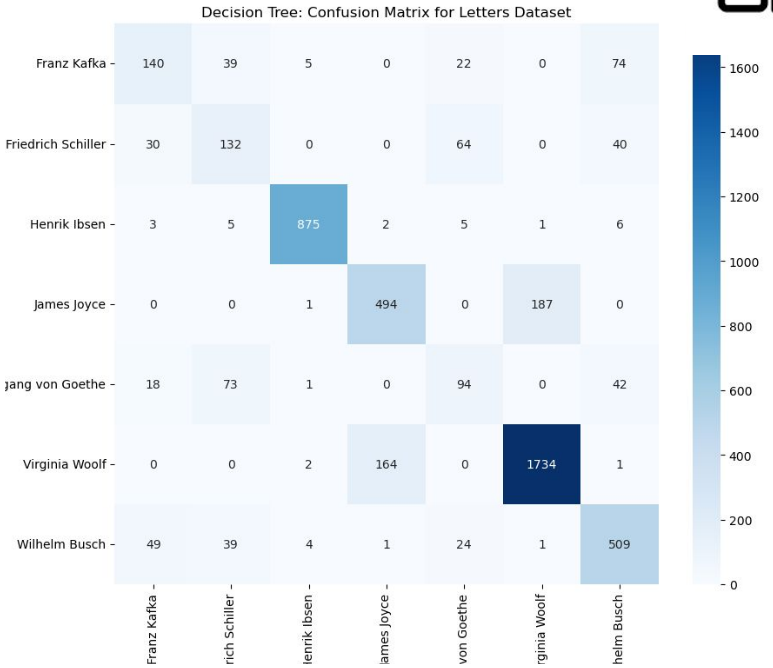
Classification Report:

	precision	recall	f1-score	support
Franz Kafka	0.58	0.50	0.54	280
Friedrich Schiller	0.46	0.50	0.48	266
Henrik Ibsen	0.99	0.98	0.98	897
James Joyce	0.75	0.72	0.74	682
Johann Wolfgang von Goethe	0.45	0.41	0.43	228
Virginia Woolf	0.90	0.91	0.91	1901
Wilhelm Busch	0.76	0.81	0.78	627
accuracy		0.81	4881	
macro avg	0.70	0.69	0.69	4881
weighted avg	0.81	0.81	0.81	4881

Accruacy: 81.50% vs. 86.85% (+5% improvement)
Virginia Woolf and *Henrik Ibsen* dominate in classification accuracy; authors like *Franz Kafka* and *Johann Wolfgang von Goethe* have weaker recall scores*(a higher number of texts by these authors are misclassified as being written by the others).*
Macro F1: 77.00% vs. 69.31%. RF showing improvement over DT, especially for underperforming classes like *Freidrich Schiller* and *Franz Kafka*.

Confusion Matrix





The dataset is somewhat imbalanced, with authors like Virginia Woolf having many samples while Johann Wolfgang von Goethe has fewer.

Language

Random Forest Results

Processing dataset: Letters Language Dataset

Evaluation for Letters Language Dataset
Accuracy: 0.9982
Precision (Macro): 0.9810
Recall (Macro): 0.9763
F1 Score (Macro): 0.9786
F1 Score (Micro): 0.9982

Classification Report:

	precision	recall	f1-score	support
da	1.00	1.00	1.00	844
de	1.00	1.00	1.00	1448
en	1.00	1.00	1.00	2519
fr	0.97	0.95	0.96	38
it	0.94	0.94	0.94	32
accuracy		1.00	4881	
macro avg	0.98	0.98	0.98	4881
weighted avg	1.00	1.00	1.00	4881

Decision Tree Results

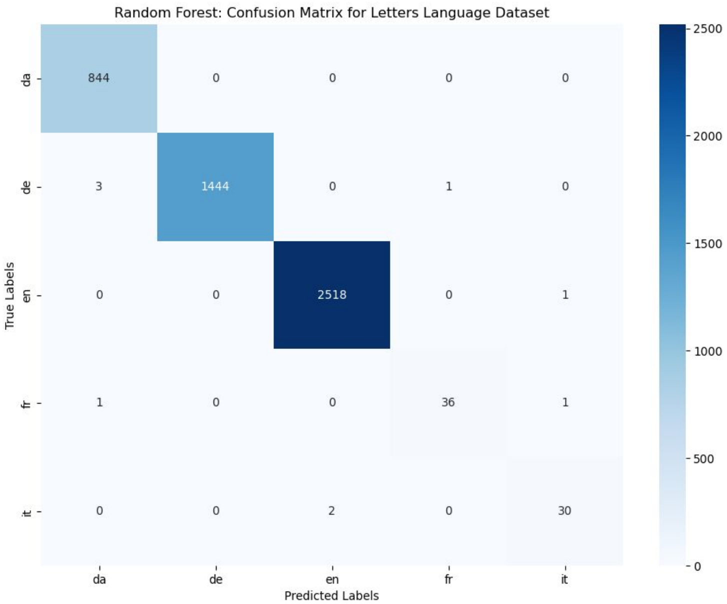
Processing dataset: Letters Language Dataset

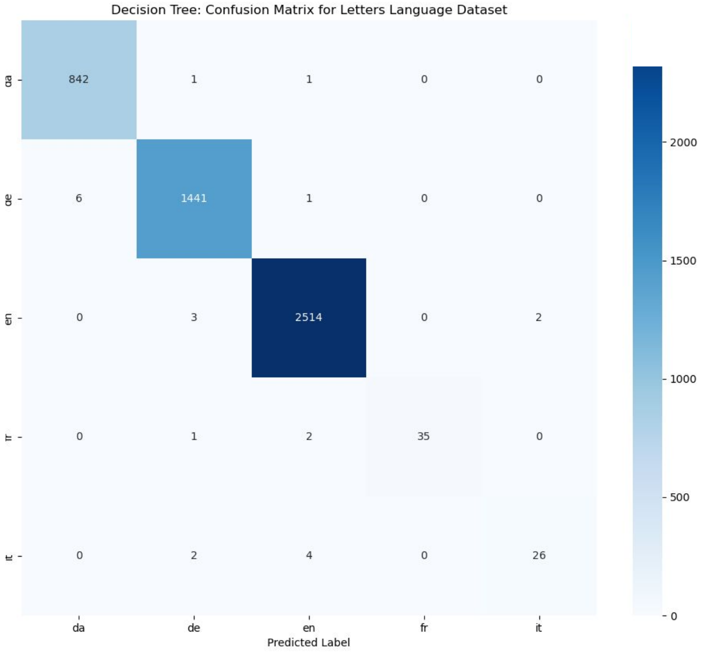
Evaluation for Letters Language Dataset
Accuracy: 0.9953
Precision (Macro): 0.9827
Recall (Macro): 0.9449
F1 Score (Macro): 0.9627
F1 Score (Micro): 0.9953

Classification Report:

	precision	recall	f1-score	support
da	0.99	1.00	1.00	844
de	1.00	1.00	1.00	1448
en	1.00	1.00	1.00	2519
fr	1.00	0.92	0.96	38
it	0.93	0.81	0.87	32
accuracy		1.00	4881	
macro avg	0.98	0.94	0.96	4881
weighted avg	1.00	1.00	1.00	4881

Accruacy: 99.82% vs. 99.53% (+0.29% improvement)
Both models excelled, but RF outperformed DT slightly (accuracy: +0.29%, Macro F1: +1.59%).
The small number of classes and clear language distinctions likely contributed to the high performance.





Almost perfect performance across all classes, with minimal misclassification,mostly for classes like “fr” and “it.”mostly for classes like “fr” and “it.”.

Results comparison between Datasets

Sentiment Dataset:

- Both DT and RF performed well, but RF achieved a significant performance boost (accuracy: +13%).
- Binary classification is simpler for both models compared to other datasets.

News Dataset:

- Both models struggled due to the high number of classes and imbalanced data.
- RF consistently outperformed DT across all metrics, showing it handles complex multi-class problems better.
- The need for feature engineering or resampling techniques is apparent to improve performance.

Letters Dataset:
Author:

- RF showed noticeable improvements over DT (accuracy: +5%, Macro F1: +8%,)
- Both models managed this multi-class problem reasonably well.

Language:

- Both models excelled, but RF outperformed DT slightly (accuracy: +0.29%, Macro F1: +1.59%).
- The small number of classes and clear language distinctions likely contributed to the high performance.

Conclusion: Decision tree vs. Random forest

- RF consistently outperforms DT across all datasets, demonstrating its robustness and ability to handle complex, imbalanced, and multi-class data.
- The largest performance gap is observed in the News Dataset, where RF's ensemble approach mitigated overfitting and improved predictions for minority classes.

Best Dataset Performance: Letters Language Dataset, with RF achieving near-perfect results.

Challenging Dataset: News Dataset due to high class imbalance and complexity.

RF is the superior choice across datasets, especially for imbalanced or multi-class problems, whereas DT is a simpler and faster baseline.

Further steps could include hyperparameter tuning for RF, oversampling techniques (e.g., SMOTE) for News Dataset, and advanced feature extraction for all datasets to boost performance.

Literature

- Categorical Modeling/Automatic Interaction Detection by William A.V. Clark, Marinus C. Deurloo, in Encyclopedia of Social Measurement, 2005 (History and explanation of DecisionTrees);
- Boosting by Peter Wittek, in Quantum Machine Learning, 2014 (Decision Tree Classifier);
- Introduction to Random forest – Simplified, Tavish Srivastava, 2020 ; https://www.analyticsvidhya.com/blog/2014/06/introduction-random-forest-simplified/
- All About Decision Tree from Scratch with Python Implementation, 2024 ; https://www.analyticsvidhya.com/blog/2020/10/all-about-decision-tree-from-scratch-with-python-implementation/
- Sklearn, Decision Trees ; https://scikit-learn.org/1.5/modules/tree.html
- Tree Based Algorithms: A Complete Tutorial from Scratch (in R & Python), Analytics Vidhya, 2024 ; https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-tutorial-scratch-in-python/?utm_source=blog&utm_medium=decision-tree-vs-random-forest-algorithm
- Random Forest Classification with Scikit-Learn, Adam Shafi, 2024 ; https://www.datacamp.com/tutorial/random-forests-classifier-python
- Sklearn-Ensemble, RandomForestClassifier ; https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- Random Forest Classifier Tutorial ; https://www.kaggle.com/code/prashant111/random-forest-classifier-tutorial

History - First appearence

- Ronald Fisher‘s paper on discriminant analysis (**1936**);
- AID project by Morgan and Sonquist and the 1966 publication by Hunt;
- Theta Automatic Interaction Detection (THAID) project by Messenger and Mandell (**1972**) - the first classification tree;
- Berkeley Statistics professors Leo Breiman, Charles Joel Stone, Jerome H. Friedman and Richard Olshen from Stanford University, began developing the classification and regression tree (CART) algorithm, unveiled in 1977

History - Improvements

- 1984 - the first official publication with a CART software;
- Computer science researcher John Ross Quinlan invented a new concept: trees with multiple answers; continued to upgrade until It was ranked No. 1 in the Top 10 Algorithms in Data Mining at the IEEE ICDM Conference (**2006**);
- Random Forests - The first such algorithm was created in 1995 by Tin Kam Ho

Thank you!

```
python DT_Countvec_notNormalized.py  83.55s user 2.90s system 29% cpu
 4:57.44 total

python RF_classify_extrametric_noNorm.py  199.47s user 3.94s system 1
% cpu 2:52:27.59 total
```