Similarity Learning in Financial Services

Dhagash Mehta

BlackRock, Inc.

Disclaimer: The views expresses here are those of the authors alone and not of BlackRock, Inc.

Introduction: Similarity

Some answers from the web…

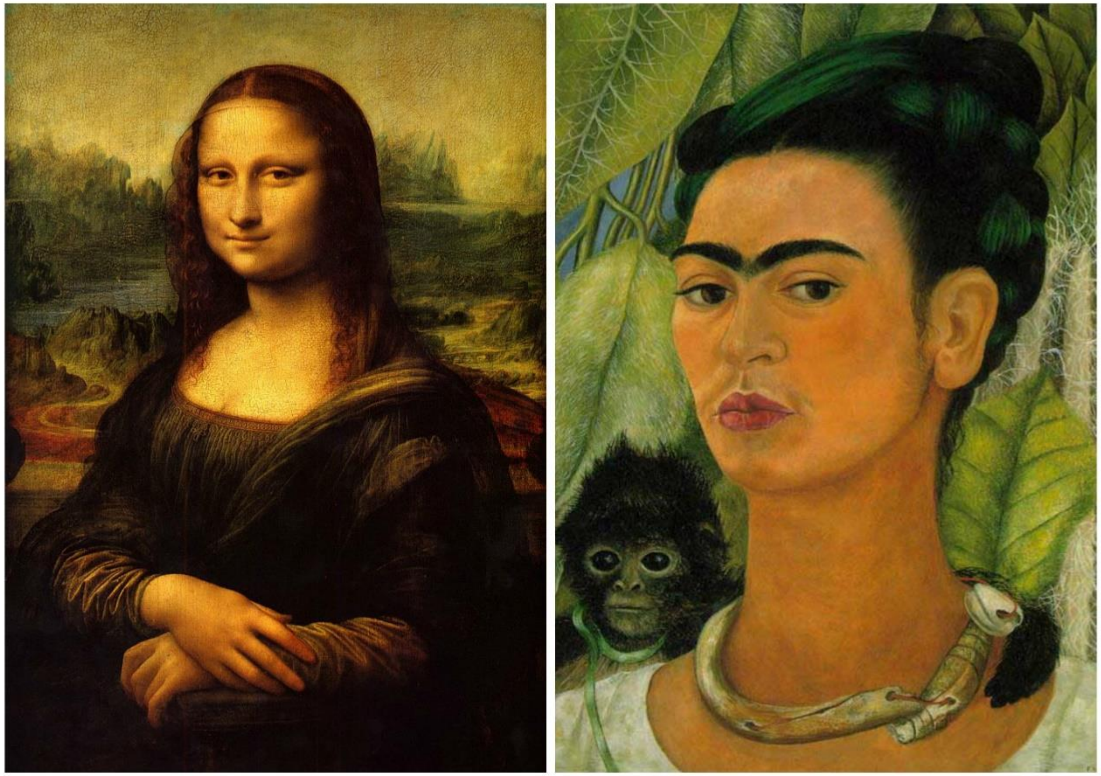
- Because [Edgar Allen] Poe wrote on both.
- Because outstanding bills are found on both of them.
- Because a writing desk is a rest for pens and a raven is a pest for wrens [type of small birds].
- Because it can produce a few notes, tho they are very flat; and it is nevar put with the wrong end in front! – Lewis Carroll’s answer after the book was published. He also wrote the spelling of ‘never’ as ‘nevar’ which is ‘raven’ spelled backwards.

Why is a raven like a writing desk?

Scene from Alice’s Adventures in Wonderland by Lewis Carroll, 1865. Artist: John Tenniel
In the picture from left to right: Alice, March Hare and Mad Hatter.
Source: https://www.theguardian.com/global/2015/dec/29/weekly-notes-queries-carroll-raven-desk

Introduction: Similarity as Cognitive Intelligence

- Recognizing similarities between two objects is one of the fundamental cognition abilities for the human as well as for many other living organisms.
- An important survival skill (e.g., comparing new objects with the previously known ones to guess if the new ones will be safer or dangerous)



Similarity in art:
“The Mona Lisa “1503-1517 painting by Italian painter Leonardo da Vinci; and “Self-Portrait with Monkey “1938, by Mexican painter Frida Kahlo.
Source: https://morrisschooldistrict.instructure.com/courses/2315/assignments/123274

Similarity: (Vague) Problem Formulation

More concretely,

- *Question 1*: Given an object/product, what are other similar objects/products?
- *Question 2*: How similar are the given two objects/products? (Or, different from each other?)
- A very frequently arising problem in most business areas.

Similarity: Applications across industries

- Amazon recommending similar items to what you search
- Netflix recommendation of similar movies to what you watched
- Zillow’s Zestimate of house price based on similar houses
- Spotify’s song/podcast/artist recommendation
- Face recognition on Facebook or screen-lock on cell-phones
- Semantic similarity (e.g., similar words, e.g., Google’s dictionary help)
- Facebook’s friend recommendation

Etc.

Similarity: Applications in Financial Services

- Many applications in finance, most already well appreciated:
  - Illiquid asset similarity (trading a more liquid substitute, e.g., corp bonds)
  - Mutual fund/ETF similarity (portfolio diversification, alternative portfolio construction, sales and marketing, tax loss harvesting, etc.)
  - Algorithmic trading, relative values

  etc.

- Most, if not all, still rely on correlations.

- Going beyond correlations has been tricky due to noisy datasets

- Similarity measures extrapolated from traditional methods (e.g., unsupervised clustering) may often fail for problems which require global, local and dynamic measure of similarity (e.g. portfolio construction and trading problems)

Similarity: Applications in Financial Services

- In this talk, we focus on identifying similarity between corporate bonds.

- The corporate bond market is diverse, securities are traded in varying volume and frequencies. Having a similarity measure can help with use cases such as:

  - identifying “liquid substitutes” in trading and investment processes to help efficiently source liquidity and significantly improve fill rates, as well as transaction costs, while reducing the negotiation cycles between traders and portfolio managers

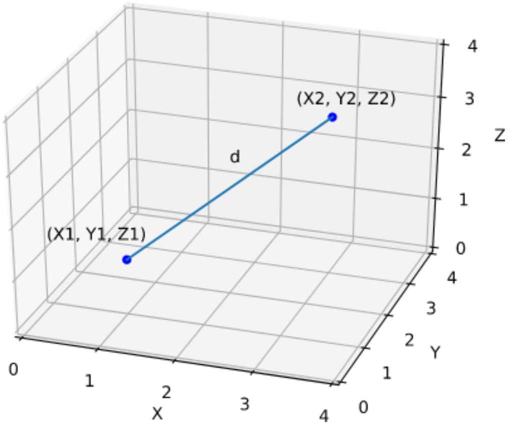
  - finding more-accurate pricing for illiquid securities where there may be little to no observable data on target, but we can infer price movements based on similar securities

  - Replacing heuristics-based sector categories broadly used in portfolio management and risk factors with more-dynamic, data-driven cohorts of similar instruments *[Madhavan, Pasquali & Sommer 2022]*

- PS: For the rest of the talk, we focus on the methodology rather than the end applications.

Similarity: Rigorous Definition

Mathematically, what is *similarity*?



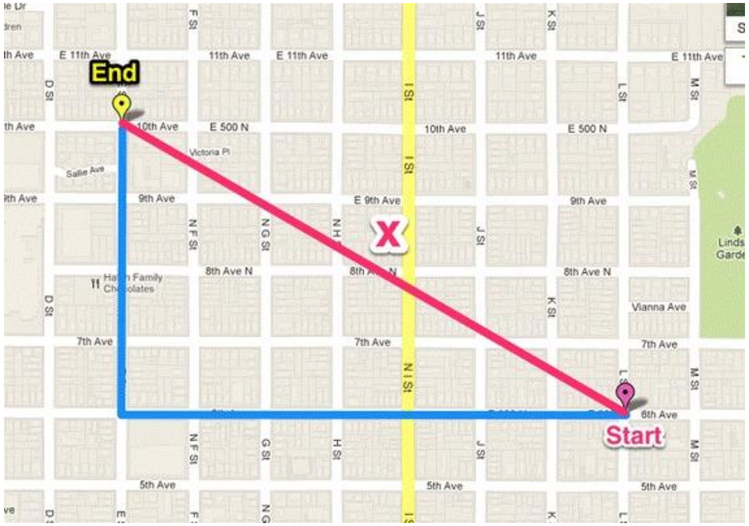
- Imagine we have 'n' number of variables such as
- Then, a bond is a point in this n-dimensional space
- Two bonds are similar if the corresponding two points in the n-dimensional space are 'close'
- What is 'close' in terms of math? Answer: the distance between the two points is small
- What is 'distance' in terms of math? Answer: the Euclidean distance

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}$$

or, is it the only option?

Similarity: Rigorous Definition

Euclidean distance isn't always the right choice…



India
Australia
© w3resource.com

Similarity: Rigorous Definition

- For a given dataset, which of these distance metric should we pick?

- Most of the academic literature on similarities in finance and economics has been implicitly or explicitly focused on the Euclidean distance.

- The importance of a correct distance metric for the given dataset or problem is much more appreciated in many other quantitative areas (Physics for one) than finance and economics, generally speaking.

Proposed Idea:

- In this work, instead of choosing a specific distance metric ourselves, we propose to *learn* it from the data

- In machine learning, learning a distance metric from data is called '*distance metric learning*', or sometimes simply '*metric learning*'.

Similarity: Distance Metric Learning

- There are various methods to learn the distance metric from the data:

1. Trial-and-error:
    - Treat the 'distance metric' of an algorithm as a hyperparameter, and tune it (Euclidean, Chebyshev, Manhattan, p-Minkowski, etc.) with respect to a chosen metric.

2. Supervised Distance Metric Learning:
    - Assume that labels for similarity between all the pairs of data, or classes for each data-point exist,
    - Work backwards to identify the distance metric which would have made these labels possible.

Similarity: Distance Metric Learning

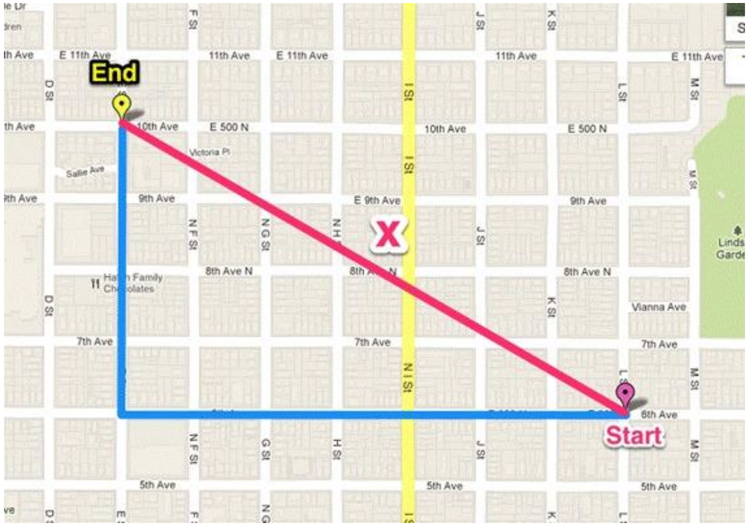
- There are various methods to learn the distance metric from the data:

1. Trial-and-error:
   - Treat the 'distance metric' of an algorithm as a hyperparameter, and tune it (Euclidean, Chebyshev, Manhattan, p-Minkowski, etc.) with respect to a chosen metric.

2. Supervised Distance Metric Learning:
   - Assume that labels for similarity between all the pairs of data, or classes for each data-point exist,
   - Work backwards to identify the distance metric which would have made these labels possible.

Traditional Distance Metric Learning

Previous examples revisited: Distance Metric Learning



India
Australia
© w3resource.com

- Keep obtaining labels (‘Close’/‘Far’) for each pair of places (of Manhattan) or cities (on the Earth)

- Based on these labels, reverse engineer the distance metric that would have made these labels possible.

- In practice, this problem can be posed as a machine learning problem.

Traditional Distance Metric Learning

- **Question**: How can we get the ‘Close’/’Far’ labels for financial assets?!

- **Answer**: Can be many ways to obtain labels, e.g.,

1. Morningstar/Lipper categories (Desai&Mehta 2021):

   If two funds are in the same category = ‘Close’
   If two funds are in different categories = ‘Far’

2. For corporate bonds, for example:

   If $|YTM_i - YTM_j| < 10^{-4}$ = ‘Close’
   Otherwise, = ‘Far’

etc.

- Choice of labels brings a specific ‘definition’ of similarity with it.

Traditional Distance Metric Learning

Classical distance metric learning (Xing et al. 2002):

- Start with a parametric (family of) distance metrics.
- E.g., the Manalanobis distance metric:

$$d(x,y) = d_A(x,y) = ||x-y||_A = \sqrt{(x-y)^T A(x-y)}.$$

PC Mahalanobis (1893-1972)

Solve the following convex optimization problem:

$$\begin{array}{ll} \min\limits_A & \sum_{(x_i,x_j)\in\mathcal{S}} ||x_i - x_j||_A^2 \\ \text{s.t.} & \sum_{(x_i,x_j)\in\mathcal{D}} ||x_i - x_j||_A \geq 1, \\ & A \succeq 0. \end{array}$$

Distance between ‘dissimilar’ data points should be large.

Minimize the distance between data points labeled as ‘similar’.

The constraint that A should be positive definite.

Solving this optimization problem will give us the final A, and we are done.

Limitations of Traditional Distance Metric Learning

The limitations of the traditional DML and its variants are:

- The variables are supposed to be numerical (as opposed to categorical) variables. Categorical variables may cause problems.
- Preprocessing of data is needed, e.g., rescaling the data to bring them between [0,1], for example;
- Missing values may cause problems;
- Not scalable to large datasets and needs reduced dimension (e.g., with PCA) if there are many variables;
- Increases the dataset size by N-choose-2 as it needs pair-wise labels. N = no of data-points.
- Learns distance metric linearly.

Distance Metric Learning using Tree-based Methods

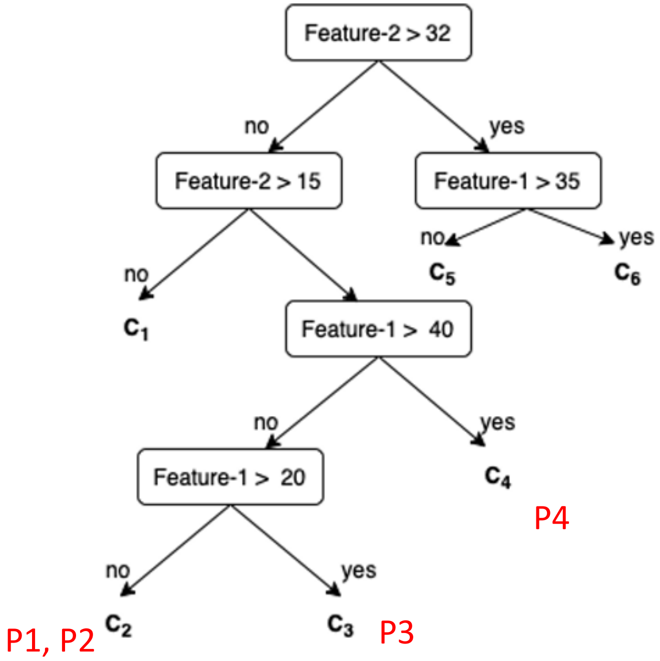
- Novel two-step idea (Breiman&Cutler2002; Jeyapaulraj, Desai, Chu, Mehta, Pasqual and Sommer, 2022):

Step-1: Train a tree-based method (e.g., Decision Tree, Random Forest, Gradient Boosting Machines, etc.).

Input variables: Chosen input variables (e.g., bond attributes)

Target variable: Chosen labels (e.g., bond yield)

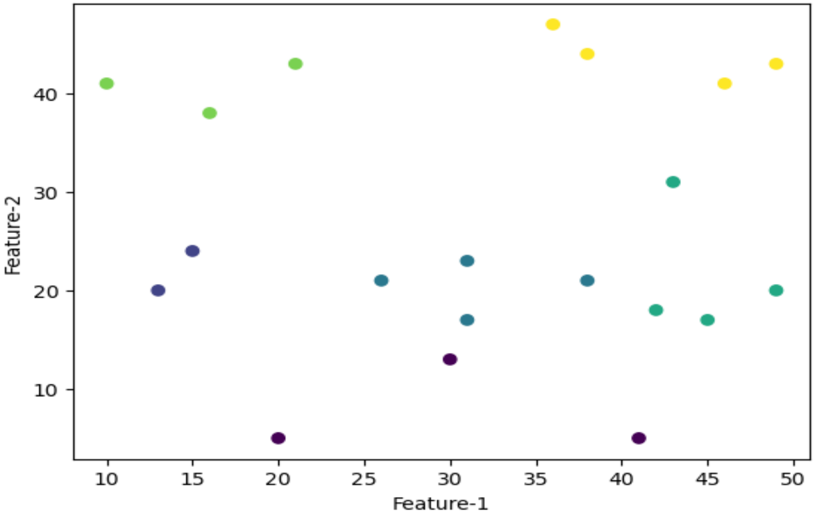
Step-2: Compute the similarity from the trees based whether a pair of data-points fall in the same leaf node.

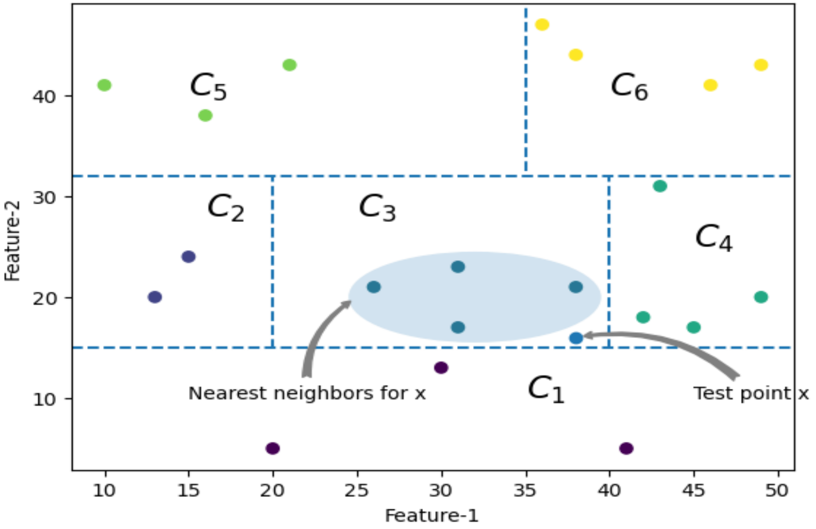


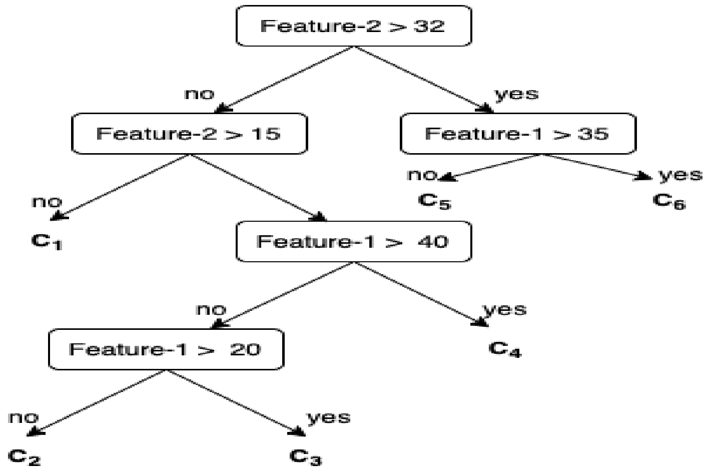
P1 and P2 are similar
P1 and P3 are dissimilar
P1 and P4 are dissimilar
P2 and P3 are dissimilar
P2 and P4 are dissimilar

Distance Metric Learning using Tree-based Methods

- Why should a decision tree (or in general tree-based methods) be viewed as a distance metric learning method?!
- Random Forest is an adaptive nearest neighbors method [Lin&Jeon2004]!







Feature-2 > 32

no

yes

Feature-2 > 15

Feature-1 > 35

no

C_1

no

C_5

yes

C_6

Feature-1 > 40

no

yes

C_4

Feature-1 > 20

no

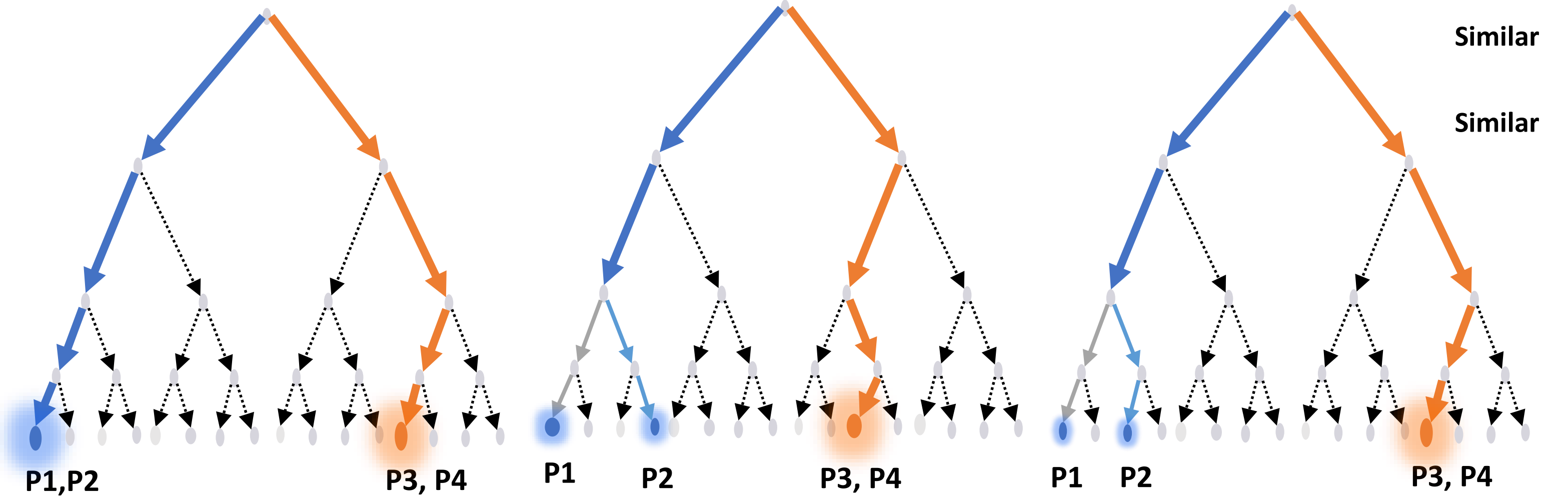
C_2

yes

C_3

Distance Metric Learning using Tree-based Methods

- For multiple trees, one can compute aggregate similarity.
- E.g., for Random Forest, the similarity between two data-points is the number of times two data-points fall in the same leaf nodes.



Similarity $_{(P1,P2)} = \frac{1+0+0}{3} = 0.33$

Similarity $_{(P3,P4)} = \frac{1+1+1}{3} = 1$

Advantages of Distance Metric Learning using Tree-based Methods

The framework overcomes limitations of the previous methodology, and carries forward the advantages of the tree-based methods:

- The variables can be numerical and categorical.
- Minimal preprocessing of variables is needed, e.g., no rescaling of the data needed;
- Missing values can be taken care of by tree-based methods;
- Scalable to much large datasets;
- No need to do dimensional reduction even if there are many variables;
- No need for pair-wise labels, and hence no increase in the size of dataset.
- Learns local distance metric due to different partitioning in different regions in the space.
- Tree-based methods are at a sweet spot between simplistic (linear) and interpretable models and complex but black-box models.

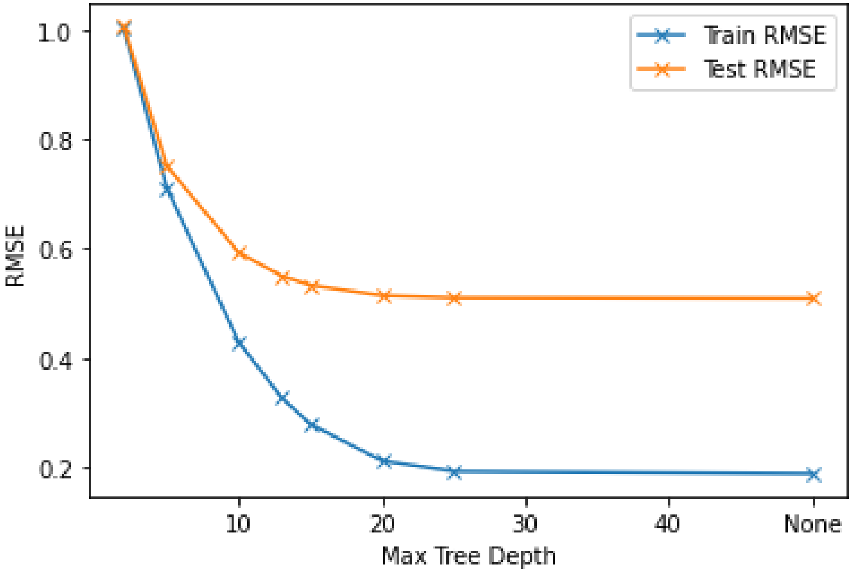
Data Description: Random Forest Based Similarity for Corporate Bonds

Data Description

- Samples: ~10K (Subset of global corporate bonds, U.S. only)
- Target: Yield to Maturity (YTM) (for the purpose of this talk)
- Cross-sectional features: Coupon, Coupon Frequency, Duration, Country, Days to Maturity, Age, Industry, Amount Issues, Amount Outstanding, Bond Rating, etc.
- Evaluation metric: RMSE (and MAPE)
- Goal: Identify bonds which are similar in terms of above mentioned features and target. In this case the target variable is related to liquidity. *[Sommer & Pasquali 2016]*
- Split/Validation: 90-10 Train-Test split; 5 Fold CV

Results: Random Forest Based Similarity for Corporate Bonds

Train Random Forest on the given data.



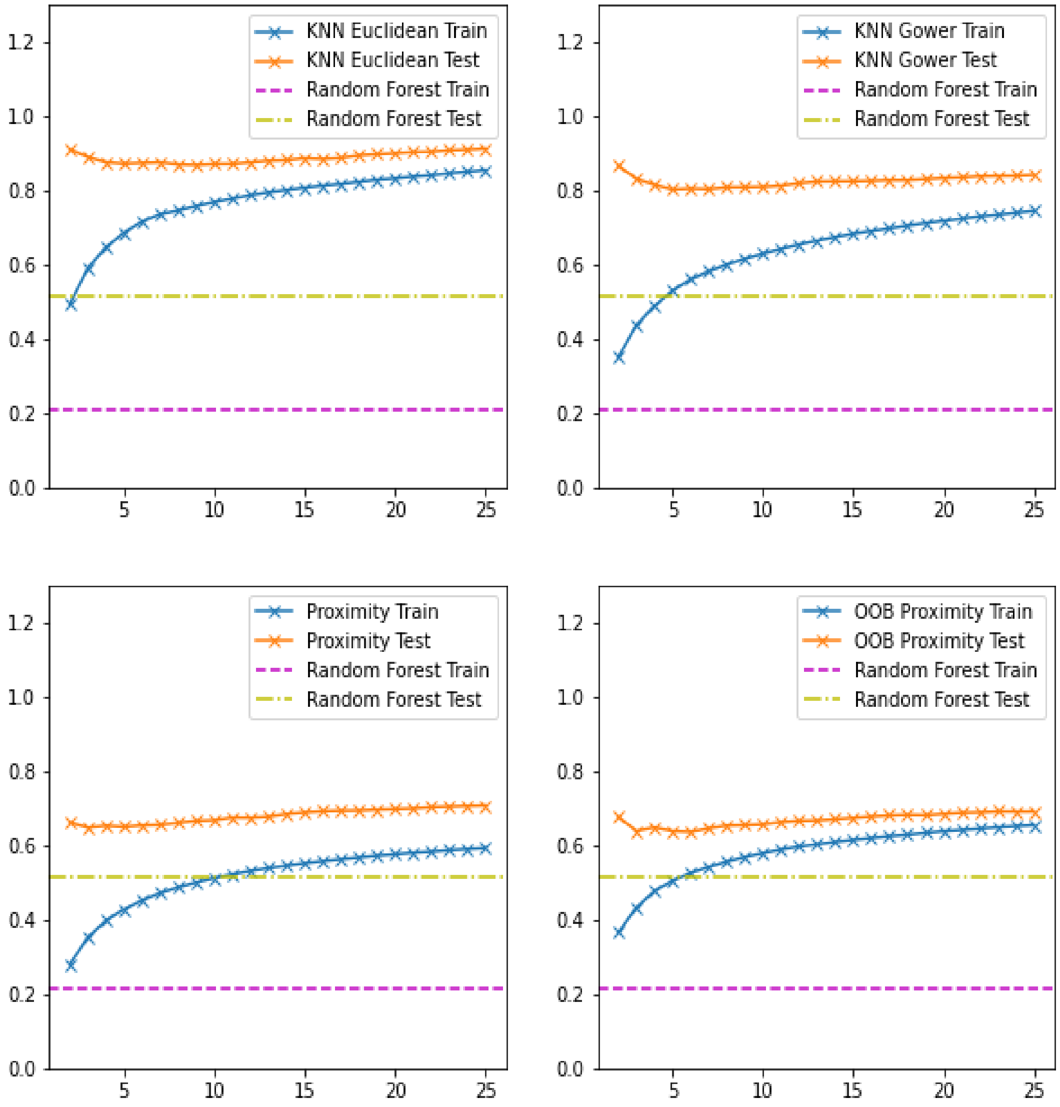
Split	RMSE	MAPE
Train (5-Fold CV)	0.21	0.08
Test	0.51	0.15

Evaluation: Random Forest Based Similarity for Corporate Bonds

- We ran k-NN algorithm on 4 different set of distance metrics:

1. Euclidean distance
2. Gower distance
3. Proximity
4. OOB Proximity

- We compute neighbors based on weighted distance (neighbors which are closer will get more weights to decide the class)



k-NN RMSE error compared to Random Forest

Conclusions

- Similarity learning is one of the most interesting areas in machine learning, with many applications in financial services.

- Using correct distance metric for the given data is crucial to compute similarity among data-points correctly.

- Supervised similarity learning is a more rigorous way to define and learn similarity rather than unsupervised (e.g., clustering with arbitrary distance metric).

- In this work, we have proposed using tree-based methodology to learn distance metric.

References

Financial Asset Similarity

- Dhagash Mehta, Dhruv Desai, and Jithin Pradeep. "**Machine learning fund categorizations.**" In *Proceedings of the First ACM International Conference on AI in Finance*, pp. 1-8. 2020.

- Vipul Satone, Dhruv Desai, and Dhagash Mehta. "**Fund2Vec: mutual funds similarity using graph learning.**" In *Proceedings of the Second ACM International Conference on AI in Finance*, pp. 1-8. 2021.

- Dhruv Desai, and Dhagash Mehta. "**On Robustness of Mutual Funds Categorization and Distance Metric Learning.**" *The Journal of Financial Data Science* 3, no. 4 (2021): 130-150.

- *Jerinsh Jeyapaulraj, Dhruv Desai, Peter Chu, Dhagash Mehta, Stefano Pasquali, and Philip Sommer. "**Supervised similarity learning for corporate bonds using Random Forest proximities.**" *arXiv preprint arXiv:2207.04368* (2022). Accepted for *Proceedings of the Third ACM International Conference on AI in Finance*. 2022.

- Dimitrios Vamvourellis, Mate Attila Toth, Dhruv Desai, Dhagash Mehta, and Stefano Pasquali. "**Learning Mutual Fund Categorization using Natural Language Processing**." *arXiv preprint arXiv:2207.04959* (2022). Accepted for *Proceedings of the Third ACM International Conference on AI in Finance*. 2022.

Investor Similarity

- Han-Tai Shiao, Cynthia Pagliaro, Dhagash Mehta. **Using Machine Learning to Model Advised-Investor Behavior**. The Journal of Financial Data Science 4 (4), 25-38. 2022.

- Fu Tan, Dhagash Mehta. Health State Risk Categorization: **A Machine Learning Clustering Approach Using Health and Retirement Study Data.** 4 (2), 139-167. 2022.

- Cynthia Pagliaro, Dhagash Mehta, Han-Tai Shiao, Shaofei Wang, Luwei Xiong. **Investor Modeling by analyzing financial advisor notes: a machine learning perspective.** Proceedings of the Second ACM International Conference on AI in Finance, 1-8. 2021.

- Thomas de Luca and Dhagash Mehta. **ESG fund usage among investor households: A machine learning based behavioral study.** Accepted for publication in Journal of ESG and Impact Investing.

Joshua Rosaler

Co-authors: Dhruv Desai, Bhaskarjit Sarmah, Dimitrios Vamvourellis,
Deran Onay, Dhagash Metha, Stefano Pasquali

Feb 7, 2023

BlackRock®

Towards Enhanced Local Explainability of Random Forests: a Proximity-Based Approach

LIMITED DISTRIBUTION

Outline

- **Goals**
    - To initiate a novel instance-based approach to model explain-ability in the context of random forest models
    - To illustrate its application in the context of a model for pricing corporate bonds
    - To highlight the ways in which this approach complements feature-based approaches to explain-ability such as SHAP and LIME

- **Outline**
    - Two Paradigms of XAI
    - Random Forest Proximities
    - Tree-based models as adaptive weighted KNN
    - Example: Corporate Bond Pricing
        - Data Description and Model Training
        - Results
    - Conclusion

Two Paradigms of XAI

- **Feature-Based**
  - Much more common
  - Attributes a model's prediction across dimensions of feature space ('column-wise' attribution):

$$\hat{y}_i = \sum_f s_{f,i}$$

  - Exemplar
    - Linear regression: explain model prediction by looking at size of each term and at regression coefficients
  - Model-agnostic approaches
    - SHAP
    - LIME
- **Instance-Based**
  - Less common
  - Attributes a model's prediction across points in the training set ('row-wise' attribution):

$$\hat{y}_i = \sum_{j \in \mathcal{D}_{train}} q_{j,i}$$

  - Exemplar
    - KNN: explain model prediction by looking at labels of nearest neighbors
  - Specific to tree-based methods

Random Forest Proximities

- **Original RF proximities**

$$s_{i,j}^{Breiman} = \frac{1}{M}\sum_{T=1}^{M} I[j \in \mathcal{L}_i^T]$$

  - Originally proposed by (Breiman 2001)
  - This formula generates a pseudo metric on the feature space: satisfies symmetry and triangle inequality, but not the axiom that d(i,j) = 0 if and only if i=j
  - (Jeyapaulraj et al 2022) show how this can be used to identify similar bonds to a given target bond

- **Original KNN expansion of RF predictions**

$$k_{i,j}^{Breiman} = \frac{1}{M}\sum_{T=1}^{M} \frac{1}{N_i^T} I[j \in \mathcal{L}_i^T]$$

  - Note that the second formula does not exactly recover the RF predictions because it does not take into account in-bag vs out-of-bag status of training points
  - This issue is corrected in (Rhodes et al 2023)

Tree-Based Models as Adaptive KNN

It can be shown that the predictions of any tree-based regression model (Decision Tree, Random Forest, GBM) can be written exactly as a linear combination of the training labels (Lin & Jeon 2004)

$$\hat{y}_i = \mathbf{k}_i \cdot \mathbf{y}_{train} = k_{i,1} y_{train,1} + \ldots + k_{i,N} y_{train,N}$$

(Rhodes et al 2023) show that the predictions of a random forest are recovered exactly if we use the so-called Geometry and Accuracy Preserving (GAP) proximities

$$k_{i,j} = \frac{1}{|S_i|} \sum_{t \in S_i} \frac{c_j(t) I[j \in J_i(t)]}{|M_i(t)|}$$

$M_i(t)$: multiset of bagged training points in same leaf as i in tree t

$c_j(t)$: multiplicity of point j in bootstrap sample for tree t

S_i : the set of trees for which i is out of bag

$J_i(t)$: the set of training points in the same leaf node as i in tree t

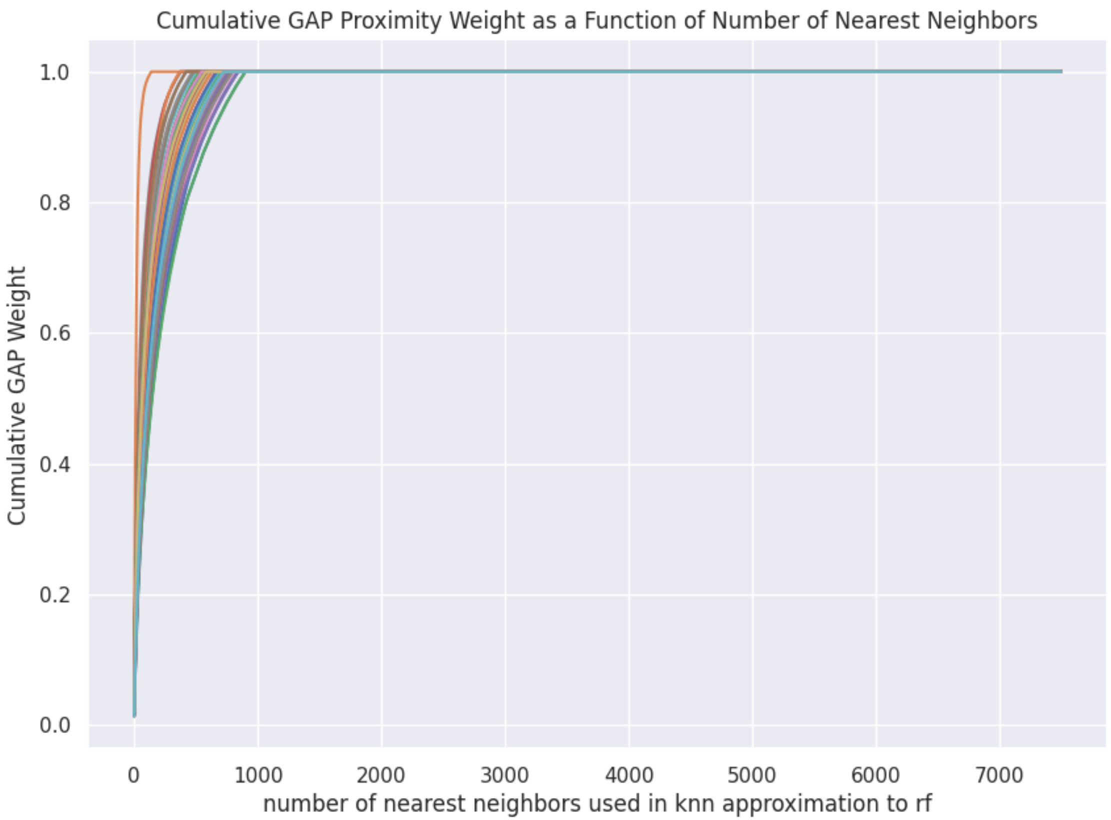
$I[j \in J_i(t)]$: 1 if $j \in J_i(t)$, 0 otherwise

For a given test point, the GAP proximities sum to 1.

Example: Corporate Bond Pricing

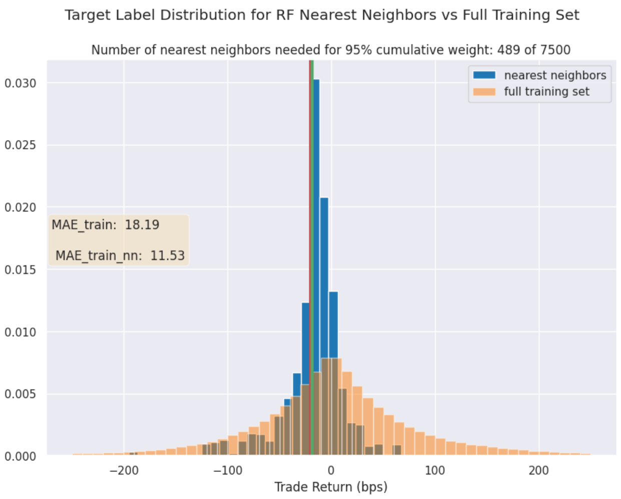
- **Use case:**
  - Corporate bond market substantially less liquid than equity market
    - Many bonds trade less than once a day
  - Less transparency as to the fair price of the bond
  - By training a model on trade data from both liquid and illiquid securities, make use of information from securities that trade more often to help price bonds that trade less frequently
  - Show how RF proximities can explain not only why the model predicts the price that it does, but also why it is more accurate in some out of sample cases and less in others
- **Data Description**
  - Target label: realized trade return relative to previous EOD price estimate
  - Main Input features:
    - Initial estimate of bond price based on weighted average of levels in TRACE trades and quotes received by BLK
    - Average return over buckets of rating, spread duration, and sector
    - Categorical bond fundamentals (target encoded): ticker, rating, industry, sector
    - Other features: time of day, time to maturity, spread duration, option adjusted spread, duration times spread, rolling 7-day price volatility
- **Training**
  - Train period: Aug. 1 2022 to Apr. 30 2023
  - Test period: May 1 2023 to July 31 2023
  - Hyperparameter tuning:
    - 5-fold walk-forward, expanding window, cross validation using randomized grid search
    - Hyperparameters: sklearn max_depth, max_features, min_samples_leaf, n_estimators
- **Performance:** 10.13% improvement in RMSE over initial estimate based on weighted average of quotes and trades
  - Test RMSE: .310 vs .345 for initial weighted average estimate
  - Train RMSE: .246 vs .431 for initial weighted average estimatte

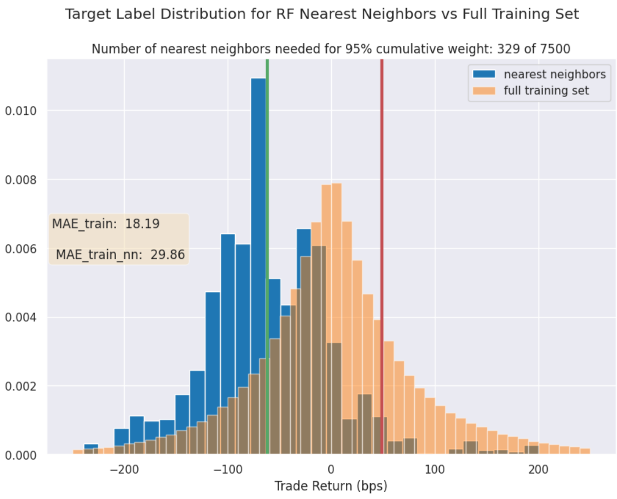
Results – How Many Nearest Neighbors?

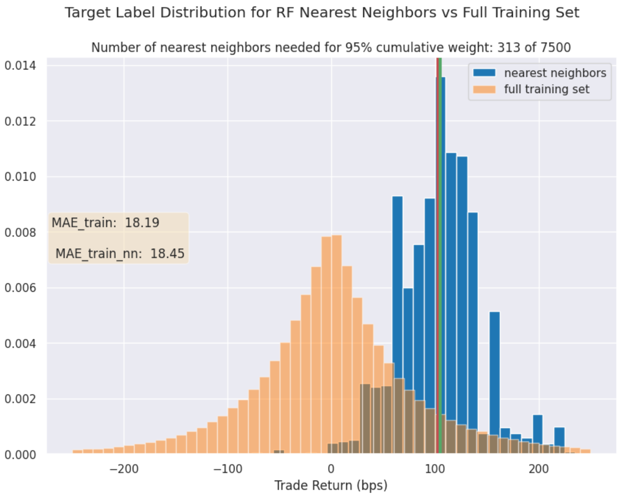


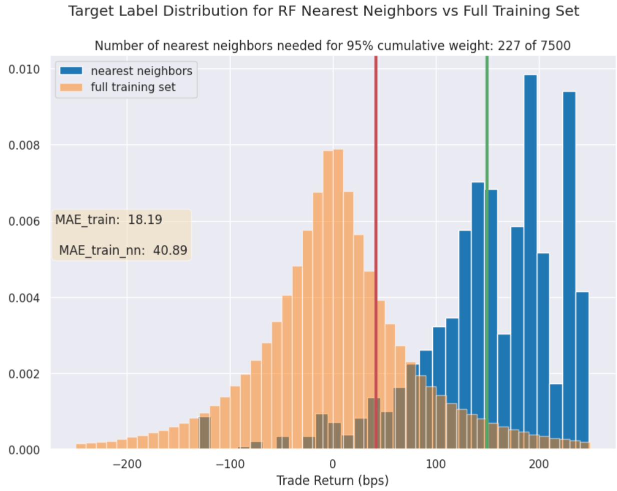
- For a given test point, the GAP proximities sum to 1
- Typically, how many nearest neighbor training points are needed in weighted KNN expansion to recover the RF prediction to good approximation (e.g., to 95% total weight)?
- Randomly sample test points and plot their cumulative GAP weight as a function of number of nearest neighbors
  - **Of the 7500 training points, only ~500 on average contribute significantly to the RF prediction**

Results – Example Explanations





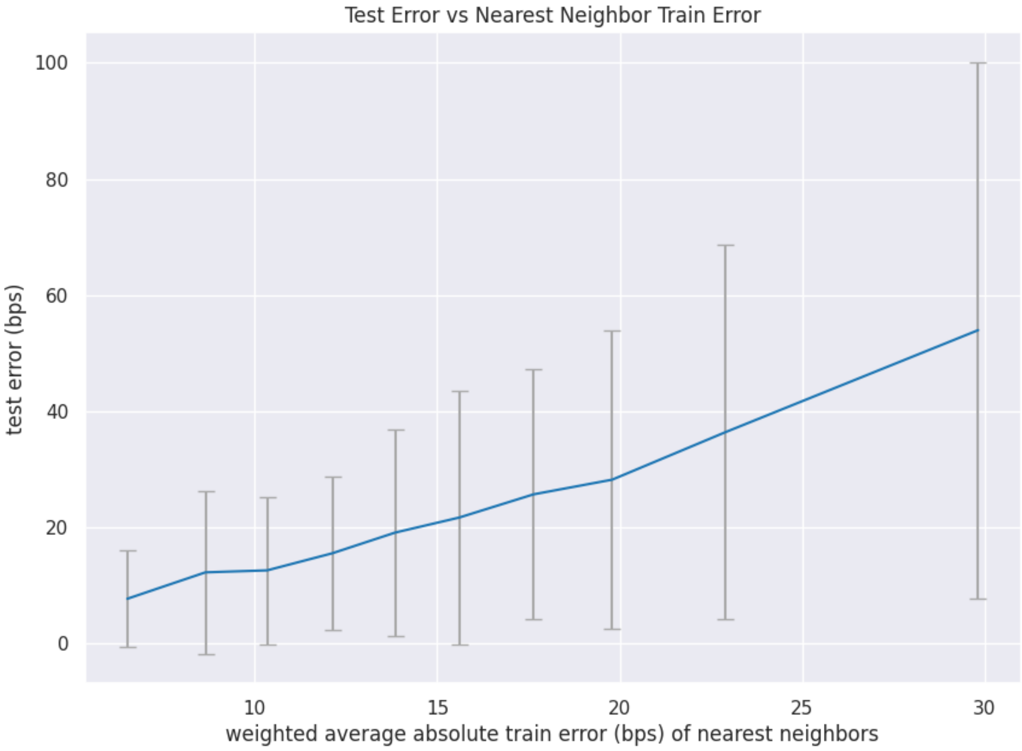




- Weighted distribution of nearest neighbors (blue histogram) is substantially more localized around the prediction than the full training set (orange histogram)
- **Good predictions (left)**
  - Realized return (red vertical line) and predicted return (green vertical line) are close
  - Proximity-weighted MAE of nearest neighbors is either less than or the same as the MAE for the full train set
  - Weighted distribution of nearest neighbors more sharply localized around prediction
- **Bad predictions (right)**
  - Realized and predicted return are far apart
  - Proximity-weighted MAE of nearest neighbors is substantially larger than the MAE for the full train set
    - This helps explain why the model was off
  - Weighted distribution of nearest neighbors more spread out than for successful predictions
- **Proximity-weighted distribution of target gives consistent estimate of conditional distribution of target variable p(y|x).**

Nearest Neighbor Training Error as an *Ex Ante* Confidence Score

- How does test set MAE vary as a function of the confidence score (given by proximity-weighted train set MAE of nearest neighbors)?
- To the right, we see that average test set MAE increases monotonically with the proximity-weighted train set MAE
- Likewise, so does the standard deviation of test set error
- On this basis, we can use the proximity-weighted train set MAE not only as an *ex post* explanation of out-of-sample performance, but also as an *ex ante* predictor of model performance out of sample



Conclusions

- In the context of tree-based models, proximity-based explanations complement methods like SHAP and LIME by giving instance-based rather than feature-based explanations
- Proximity-based explanations have the advantage **that they can explain not just model predictions but also model error**
  - Beyond explaining model error in a *post-hoc* fashion, **RF proximities give an *ex ante* measure of confidence in the model's predictions that is more local and granular than measures such as out-of-bag error or validation error**, in the sense that the confidence measure is specific to the test point in question and to neighborhood of feature space in which it lies
- An advantage of approaches like SHAP and LIME is that they are model-agnostic
- These points are illustrated in the context of an RF model used for corporate bond pricing
- Extensions (follow-up paper)
  - Extend instance-based explanations to RF classifier
  - Extend to GBMs
  - Use proximities to identify which test points are outliers relative to training set