

# ML5 – unsupervised learning and explainability

Machine Learning – Tools and applications for policy – Lecture 6

Iman van Lelyveld – Michiel Nijhuis

DNB Data Science Hub



## ML5 – unsupervised learning and explainability

---

1. Supervised versus unsupervised learning
2. What can we do with unsupervised learners?
  - K-means, t-SNE, DBSCAN, Gaussian mixtures
3. Can we transfer knowledge?
4. How to open the black box and explain results?

## ML5 – unsupervised learning and explainability

### Unsupervised Learning

- k*-Means clustering

- t-SNE

### Hierarchical Clustering

- DBSCAN

- Gaussian mixtures

### Transfer Learning

### Explainable AI



- K-means animation (Andrey Shabalin) ([link](#))
- K-means clustering (StatQuest) ([link](#))

## ML5 – unsupervised learning and explainability

### Unsupervised Learning

*k*-Means clustering

t-SNE

### Hierarchical Clustering

DBSCAN

Gaussian mixtures

### Transfer Learning

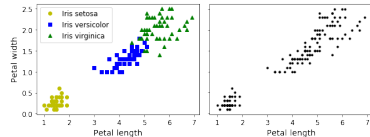
### Explainable AI



- The vast majority of data is **unlabeled** → enter **unsupervised learning**
- Flavors:
  - **Clustering**: group observations in similar groups for customer segmentation, recommender systems
  - **Anomaly detection**: what is “normal” so you can detect abnormal observations
  - **Density estimation**: what is the PDF of a DGP.  
Anomalies are probably in the low density areas

- The vast majority of data is **unlabeled** → enter **unsupervised learning**
- Flavors:
  - **Clustering**: group observations in similar groups for customer segmentation, recommender systems
  - **Anomaly detection**: what is “normal” so you can detect abnormal observations
  - **Density estimation**: what is the PDF of a DGP. Anomalies are probably in the low density areas

## Grouping with labels is easy

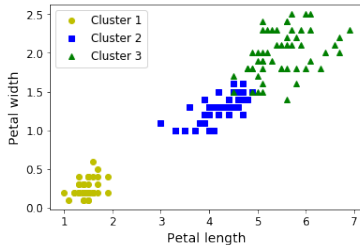


Source: Géron [Github](#)



- The vast majority of data is **unlabeled** → enter **unsupervised learning**
- Flavors:
  - **Clustering**: group observations in similar groups for customer segmentation, recommender systems
  - **Anomaly detection**: what is “normal” so you can detect abnormal observations
  - **Density estimation**: what is the PDF of a DGP. Anomalies are probably in the low density areas

A Gaussian mixture model (covered later) can separate these clusters pretty well



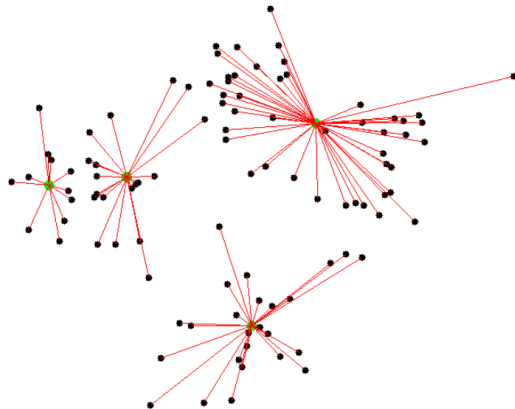
Source: Géron [Github](#)





- Goal: to find subgroups or clusters by partitioning dataset into distinct groups that are maximally “different” from one another.
  - Requires a definition of what is similar/different. This is often domain specific
- Types of clustering techniques
  - **k-means clustering**: requires decision for the number of clusters  $k$
  - **t-SNE**: non-linear PCA
  - **DBSCAN**: looks for “dense” areas in feature space
  - **Hierarchical clustering**: creates tree-like dendrogram that depicts all possible clusters of size 1 to  $n$ .
  - **Gaussian mixtures**: data is generated from an unknown mixture of several Gaussian distributions with unknown parameters
  - Agglomerative clustering, BIRCH, mean-shift, affinity propagation, spectral clustering

Starting with 4 left-most points. Click the picture to continue.



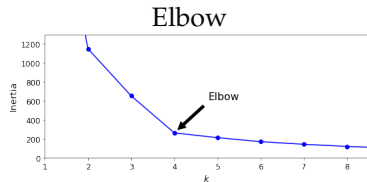
- **hard clustering** = assigning to a single cluster vs **soft clustering** = assigning a score
- **Inertia** is the performance metric: mean squared distance to the closest **centroid**

## Disadvantages

1. Guaranteed to converge (mostly quickly) but unclear if clustering is optimal
  - See plotting the **inertia attribute**. Or increase *n\_init*
  - Use **MiniBatchKMeans** estimator in **SKLearn**, which reduces computation time significantly with only slight worse quality (See **comparison**)
2. Not easy/impossible to spot visually in more than 3 dimensions
3. Requires **choosing the number of clusters**.

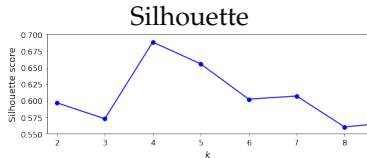


- Plot **inertia** over  $k$  and find **elbow**



Source: Géron [Github](#)

- Plot **inertia** over  $k$  and find **elbow**
- Plot **silhouette**
  - $(b-a) / \max(a,b)$  where  $a$  is the mean intra-cluster distance and  $b$  is the mean distance to the next cluster
  - Range: -1 to +1

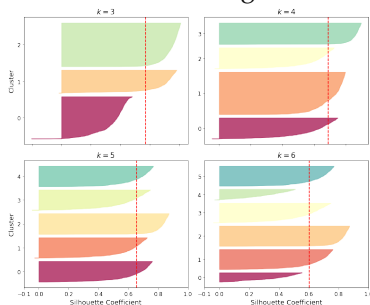


Source: Géron [Github](#)



- Plot **inertia** over  $k$  and find **elbow**
- Plot **silhouette**
  - $(b-a) / \max(a,b)$  where  $a$  is the mean intra-cluster distance and  $b$  is the mean distance to the next cluster
  - Range: -1 to +1
- Plot **distribution of silhouette scores**
- If you aim for similar sized groups where the observations all contribute roughly equally then  $k = 5$  seems OK

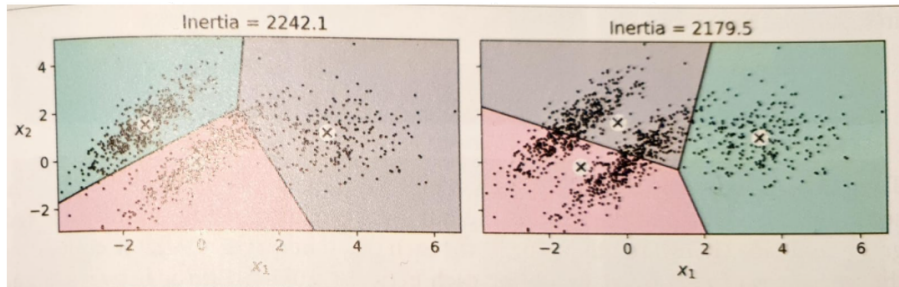
Silhouette diagram



Source: Géron [Github](#)

K-means does not behave well if:

- Clusters have varying sizes
- Different densities
- Nonspherical shapes



- **t-Distributed Stochastic Neighbor Embedding** (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data
- Difference between **PCA** and **t-SNE**: linear vs non-linear
- t-SNE calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. It then tries to optimize these two similarity measures using a cost function

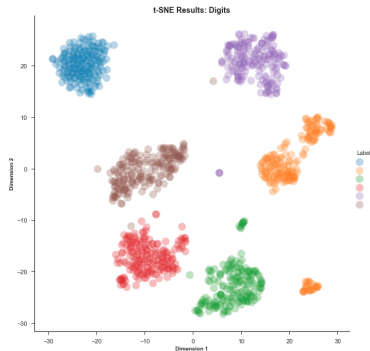


Source: Violante on [Medium](#)





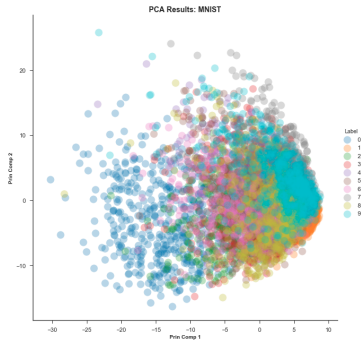
- **t-Distributed Stochastic Neighbor Embedding** (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data
- Difference between **PCA** and **t-SNE**: linear vs non-linear
- t-SNE calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. It then tries to optimize these two similarity measures using a cost function



Source: Violante on [Medium](#)

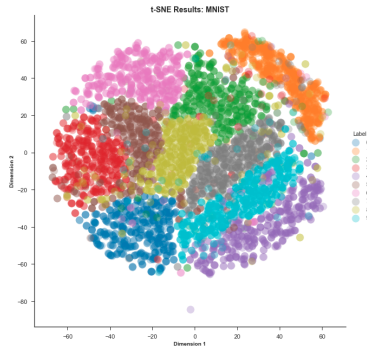


- **t-Distributed Stochastic Neighbor Embedding** (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data
- Difference between **PCA** and **t-SNE**: linear vs non-linear
- t-SNE calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. It then tries to optimize these two similarity measures using a cost function



Source: Violante on [Medium](#)

- **t-Distributed Stochastic Neighbor Embedding** (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data
- Difference between **PCA** and **t-SNE**: linear vs non-linear
- t-SNE calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. It then tries to optimize these two similarity measures using a cost function

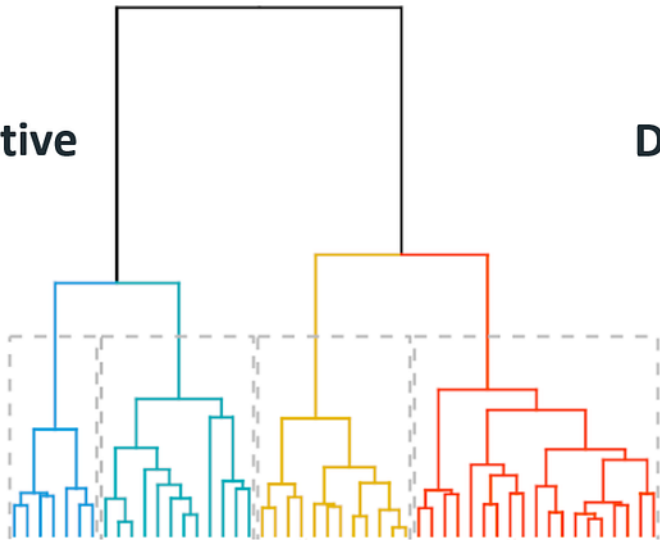


Source: Violante on [Medium](#)

- A drawback of k-means is the need to pick the total number of clusters a priori.
- **Agglomerative Clustering**: builds clusters one at a time in a bottom-up fashion. Initially, each of the data points is assumed to be a separate cluster. Then, the clusters that are closest to each other are merged to form a merged cluster. Repeat until all data points are merged into one single cluster. This method can help us to determine the hierarchy in which clusters were formed along with their similarity with each other. **Dendrograms** show this tree-like structure that shows the way in which all the clusters are formed.
- **Divisive Clustering**: top-down approach where all the data points are assigned to a single cluster. We split the data points into clusters based on their distance. These steps are followed until we have  $n$  clusters when  $n$  indicates the total number of data points chosen for clustering. Having a large value of  $n$  leads to having a higher number of clusters and higher computation costs. It is easier to visualize the results with the help of a dendrogram that also shows similarities between clusters that are used to divide them.

**Agglomerative**

**Divisive**



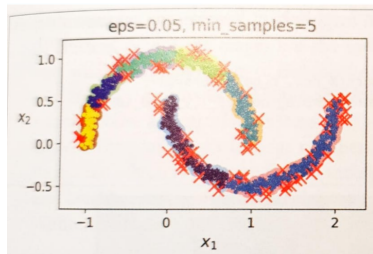
## Pros

- No requirement to specify the number of clusters a priori
- More intuitive as there is a hierarchy in the structure, and it is easier to interpret.

## Cons

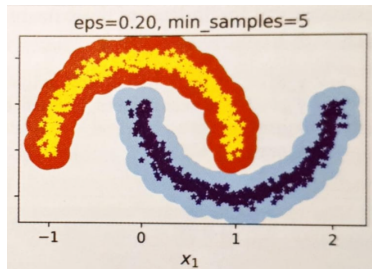
- Compute to run this algorithm is quite high. For large datasets, it can be better to look for alternative algorithms.
- Sensitive to outliers which affect the way in which clusters are formed or generated.
- Distance metric chosen used to split or merge clusters can have a significant impact

- Looks for “dense” areas in feature space
- Has just 2 hyperparameters:  $\epsilon$  and *min\_samples*
- Works well if dense areas are clearly separated by sparse areas



Source: Géron (2019)

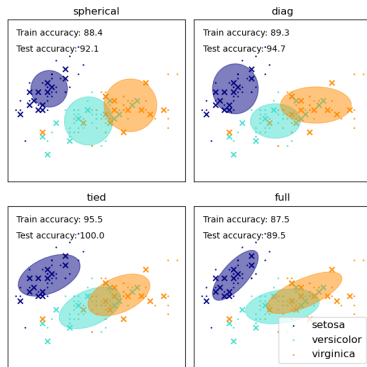
- Looks for “dense” areas in feature space
- Has just 2 hyperparameters:  $\epsilon$  and *min\_samples*
- Works well if dense areas are clearly separated by sparse areas



Source: Géron (2019)



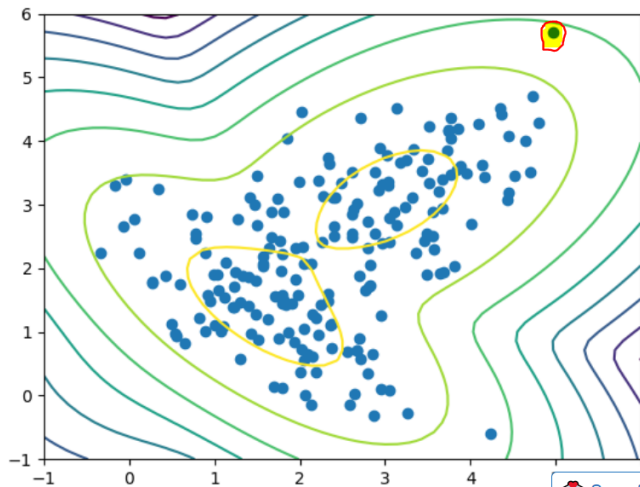
- Assumes data is generated from an unknown mixture of several Gaussian distributions with unknown parameters
- Expectation Maximization* (EM)
  - Similar to **k**-
  - Estimates not only the center but also size, shape, orientation and relative weight with soft assignments
- To reduce computational complexity adjust *covariance\_type*: “spherical”, “diag”, “tied” and “full” (default)



Source: Géron (2019)

- Number of clusters  $k$  is a hyperparameter (similar to K-means)
- **Inertia** or **silhouette** not well-defined
- **Bayesian Information Criterion** (BIC) and **Akaike Information Criterion** (AIC)
- Both BIC and AIC penalize models that have more parameters to learn (== more clusters) and reward models that fit the data well
- Plot BIC/AIC for an **elbow plot**





- **Principal Components Analysis (PCA)**
- **Fast-MCD** (minimum covariance determinant): variant of Gaussian mixture with a single distribution
- **Isolation forest**: Random Forest where each decision tree is grown randomly. At each node a random feature is used to split using a random threshold. Outliers tend to be split of relatively fast.
- **Local Outlier Factor (LOF)**: compares density with its neighbors' density
- **One-class SVM**: can we split observations from origin? Does not scale

## ML5 – unsupervised learning and explainability

### Unsupervised Learning

- k*-Means clustering

- t-SNE

### Hierarchical Clustering

- DBSCAN

- Gaussian mixtures

### Transfer Learning

### Explainable AI



- Knowledge from learning one task can be used for a different but related task
- E.g. by learning how to recognize trucks, a model could use this knowledge to recognize a passenger car



- We define a domain  $D$  and a task  $T$ , both for a **source** and a **target**.
- Domain  $D$  contains a feature space  $\chi$  and a marginal probability distribution  $P(X)$ , where  $X = x_1, \dots, x_n \in \chi : D = \{\chi, P(X)\}$ . (Pan & Yang (2010))
- Task  $T$  consists of label space  $\mathcal{Y}$  and an objective predictive function  $f(\cdot)$ . The task cannot be observed, but can only be learned from the training pairs  $x_i$  and  $y_i$ . Hereby,  $f(\cdot)$  can be used to **predict the label** of a new instance  $x$ , so  $T = \{\mathcal{Y}, P(\mathcal{Y}|X)\}$ .
- Using the knowledge of  $D_S$  and  $T_S$ , Transfer Learning tries to improve the learning of the **predictive function**  $f(\cdot)$  in  $D_T$ , where  $D_S \neq D_T$ , or  $T_S \neq T_T$ .

- Image recognition, cancer discovery and natural language processing.
- Useful in cases of data scarcity:
  - Extreme Out-of-the-Money options
  - Different stocks being affected by the same market
  - Credit data on MSMEs (Suryanto et al. [2022](#))
- Shows increased performance over regular learning. (He et al. [2019](#))





- Look at when to transfer, how to transfer, and what to transfer. (Lin and Jung 2017)
- When: in classification problems, brute-force transfer might cause a negative transfer, so do not **random-guess a transfer**, but consider if the performance is above **chance level**
- How: check if the training and test data are drawn from **similar distributions**, by checking the **Pearson's correlation coefficient**, for example
- What: apply **feature selection prior to transfer**, so irrelevant information isn't transferred

## ML5 – unsupervised learning and explainability

### Unsupervised Learning

*k*-Means clustering

t-SNE

### Hierarchical Clustering

DBSCAN

Gaussian mixtures

### Transfer Learning

### Explainable AI



- XAI methods can be of substantial help in revealing the decision rules behind, for instance, a complex Random Forest (RDF) model to forecast corporate default Cascarino et al. (2021).
  - default forecasts based on the RDF model exploit more of the available information set + greater importance to indicators with non-linear/non-monotonic relationships with the outcome variable
  - RDF forecasts display stability over time in the importance assigned to key predictors
  - In stress, liquidity ratios also give a relevant contribution to predictions

- Robustness



- Robustness
- Causality



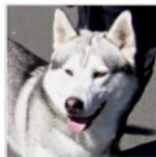
Predicted: wolf  
True: wolf



Predicted: husky  
True: husky



Predicted: wolf  
True: wolf



Predicted: husky  
True: husky



Predicted: wolf  
True: wolf



Predicted: wolf  
True: husky



- Robustness
- Causality



- Robustness
- Causality
- Fairness

RETAIL OCTOBER 11, 2018 / 1:04 AM / UPDATED 4 YEARS AGO

## Amazon scraps secret AI recruiting tool that showed bias against women

By Jeffrey Dastin

8 MIN READ



SAN FRANCISCO (Reuters) - Amazon.com Inc's [AMZN.O](#) machine-learning specialists uncovered a big problem: their new recruiting engine did not like women.



- Robustness
- Causality
- Fairness
- Privacy

Home › Tech News › AI Paper Summary › Apple Researchers Propose A Method For Reconstructing Training Data From Diverse Models

## Apple Researchers Propose A Method For Reconstructing Training Data From Diverse Machine Learning Models By Ensemble Inversion

By **Nitish Kumar** - November 27, 2021



Facebook



Twitter



LinkedIn



Reddit



Email



Print

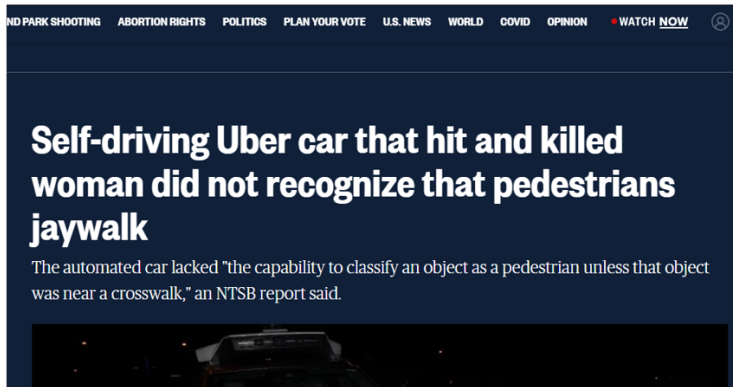
Model inversion (MI), where an adversary abuses access to a trained Machine Learning (ML) model in order to infer sensitive information about the model's original training data,



DataScience  
Hub

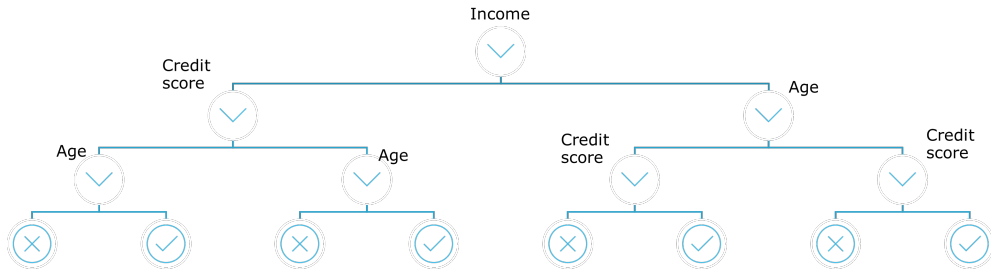


- Robustness
- Causality
- Fairness
- Privacy
- Trust

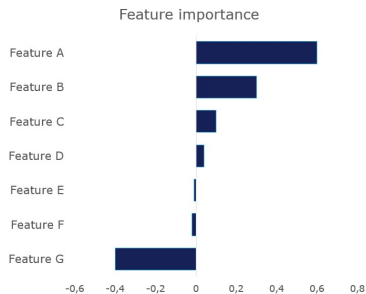


1. White box models
2. Global explainability
3. Local explainability

- Models which are familiar or easy to understand
  - e.g. linear regression or decision trees



- What variables are most important in our model?
- e.g. What factors are important for predicting a higher default rate?
- Pertubation ([Explain it like I'm 5 \(ELI5\)](#)), Partial dependence plots



### 3. Local explainability

32

- Why is a certain decision made?
- Why does loan X have a higher default rate?
  - LIME, Shapley values (SHAP), Counterfactuals



LIME ((Ribeiro et al. 2016), implemented in ELI5 is an algorithm to explain predictions of black-box estimators:

1. Generate a **fake dataset** based on the data analysed
2. Use **black-box estimator** to get **target values** in a generated dataset (e.g. class probabilities)
3. Train a **new white-box estimator**, using generated data + labels as training data
4. Explain the original example through weights of this white-box estimator instead
5. Prediction quality of a white-box classifier shows how well it approximates the black-box classifier. If the quality is low then don't trust explanation

- Pretend the AI model is a cooperative game
  - Payout is the prediction accuracy
  - Players are the parameters
- Calculate a fair way to distribute the payout over the players

$$\phi(\nu) = \frac{1}{|N|} \sum_{S \subseteq N \setminus \{i\}} \frac{n - |S| - 1}{|S|} [v(S \cup \{i\}) - v(S)]$$

1. Relative importance player  $i$  2. Number of players 3. Each coalition without player  $i$  4. Number of coalitions of this size without player  $i$  5. Additional payout of including  $i$  to the coalition













In this lecture we covered:

1. Some **unsupervised learners**
  - k-Means clustering, t-SNE, DBSCAN, Gaussian mixtures
2. The idea that you can transfer knowledge from one application to another
3. A first look at **explainable AI**
  - white box, global and local explainability

- 
-  Cascarino, G., Moscatelli, M., & Parlapiano, F. (2021). Explainable Artificial Intelligence: Interpreting default forecasting models based on Machine Learning. Working Paper.
-  Géron, A. (2019). Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly. Retrieved April 21, 2019, from <https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>
-  He, Q.-Q., Pang, P. C.-I., & Si, Y.-W. (2019). Transfer Learning for Financial Time Series Forecasting. PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference 24–36.
-  Lin, Y.-P., & Jung, T.-P. (2017). Improving EEG-Based Emotion Classification Using Conditional Transfer Learning. Frontiers in Human Neuroscience, 11, 334.
-  Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning [Conference Name: IEEE Transactions on Knowledge and Data Engineering]. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1



- 
-  Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier [arXiv:1602.04938 [cs, stat]].
  -  Suryanto, H., Mahidadia, A., Bain, M., Guan, C., & Guan, A. (2022). Credit Risk Modeling Using Transfer Learning and Domain Adaptation. *Frontiers in Artificial Intelligence*, 5. Retrieved November 16, 2023, from <https://www.frontiersin.org/articles/10.3389/frai.2022.868232>

