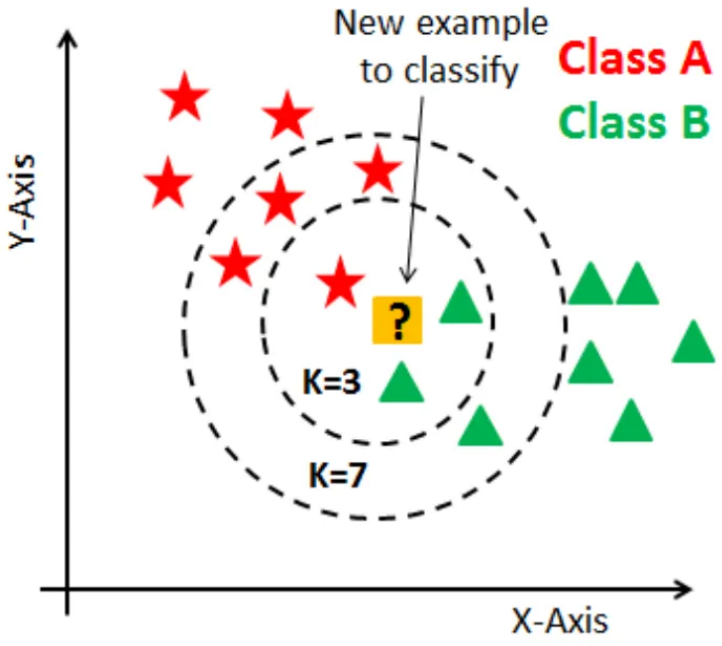
10 - Prática: Lidando com Dados do Mundo Real (II)

Descrição da Atividade

Section 6: More Data Mining and Machine Learning Techniques

1. K-Nearest-Neighbors: Concepts

- Classify new data points based on Distance to know data.



- KNN defines some distance metric between the items in your dataset, and find the K closest items.
- USe those items to predict some property of a test item, by having them somehow "vote" on it.

2. Using KNN to predict a rating for a movie

- Predicting the rating of *Toy Story* by analyzing movies with similar themes.
    1. Import Rating Data.
    2. Group everything by movie ID.
    3. Compute the total number of ratings and the average rating for every movie.
    4. Create a new DataFrame that contains the normalized number of ratings. (A value of 0 means nobody rated it, and a value of 1 will mean it's the most popular movie there is).
    5. Import Genre Data.
    6. Create a dictionary called movieDict = there are 19 fields, each corresponding to a specific genre.
    7. Define a function to compute the distance(KNN).
    8. Sort the movies by distance and print the KNN.
    9. Average rating of the 10 nearest neighbors to Toy Story.
    10. How does this compare to Toy Story's actual average rating?
        a. Predict = 3.344. Actual = 3.878. Not a bad result.

3.Dimensionality Reduction: Principal Component Analysis (PCA)

- PCA is a dimensionality reduction technique, to deal with huge number of dimensions.
    - Example: In recommending movies the ratings vector for each movie
- Dimensionality Reduction = Distill higher-dimensional data down to a smaller number of dimension.
    - Preserving as much of the variance in the data as possible.
- K-Means Clustering = is an example of dimensionality reduction algorithm.

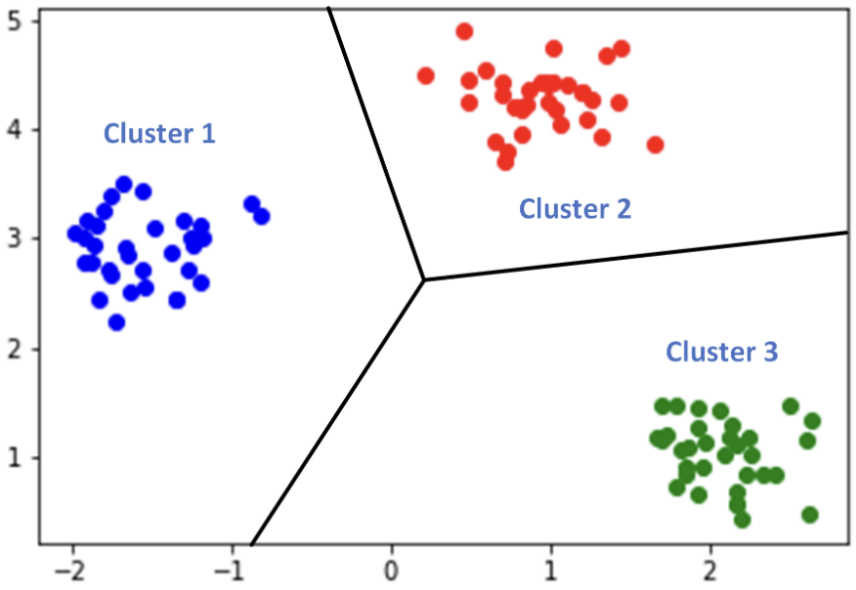
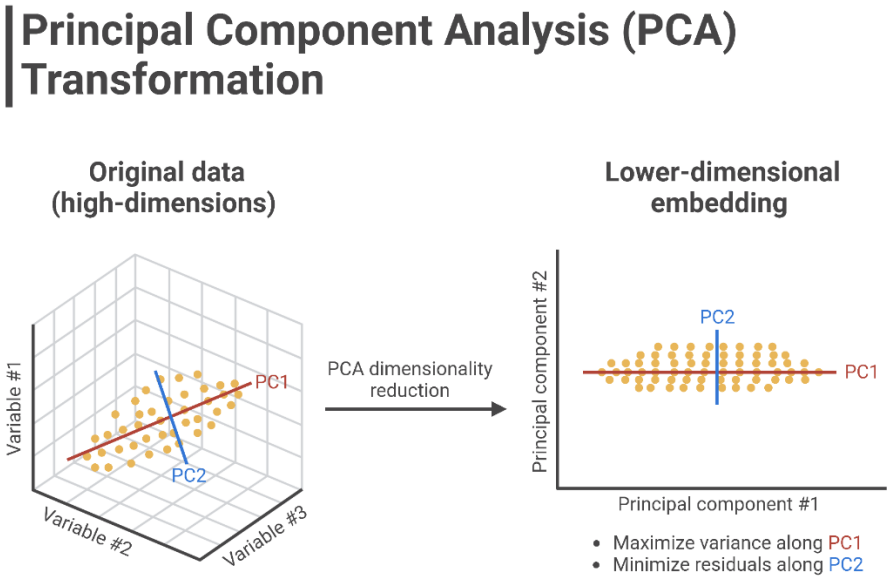


Fig.1. An Example Of Data Clustering

- Principal Component Analysis.
  - Really useful for things like image compression and facial recognition.
  - A popular implementation is called Singular Value Decomposition(SVD).



4.[Activity] PCA Example with the Iris data set

1. **Import Iris Data Set.**

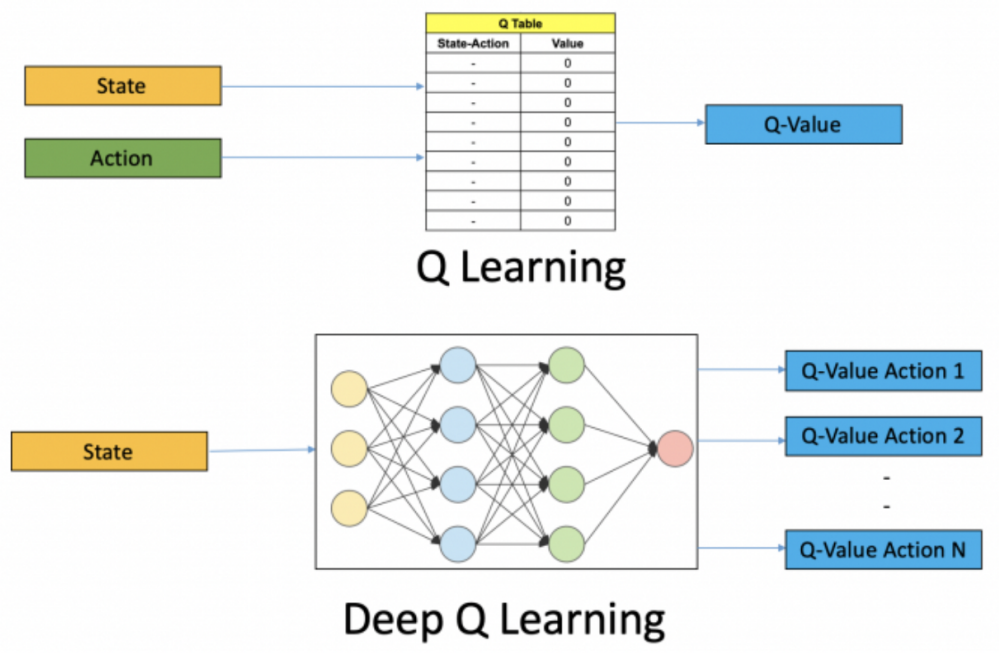
2. **Distill our 4D data set down to 2D.**
3. **See how much information we've managed to preserve.**
4. **Plot 2D Graph.**

5.Data Warehousing Overview: ETL and ELT

- Data Warehousing:
  - A Large, centralized database that contains information from many sources.
- ETL and ELT: refer to how data gets into a data warehouse.
- ETL: Extract, Transform, Load.
- ELT: Extract, Load, Transform (big data).

6.Reinforcement Learning

- RL: Some sort of agent that “explores” some space.
  - Example: Pac-Man, Cat & Mouse game.
- Q-Learning: A specific implementation of reinforcement learning.
  - Actions: Semi-randomly explore different choices of movement.
  - States: Given different conditions.
  - For each state/action, a Reward or Penalty is given.
  - Use those Q values to inform its future choices.

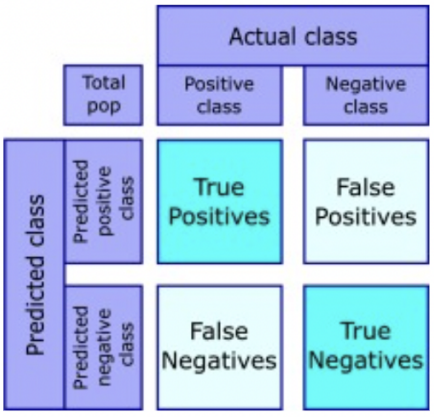


- Fancy/Math Words.
  - MDPs (Markov Decision Processes) = a mathematical framework for modeling Decision Making, in situations where outcomes are partly Random and partly Under the Control.
  - Our Q values are described as a reward function Pa(s,s').
  - Discrete time stochastic control process = is the same just a fancy term.
  - Dynamic Programming = solving a complex problem by breaking it down into a simpler subproblems.

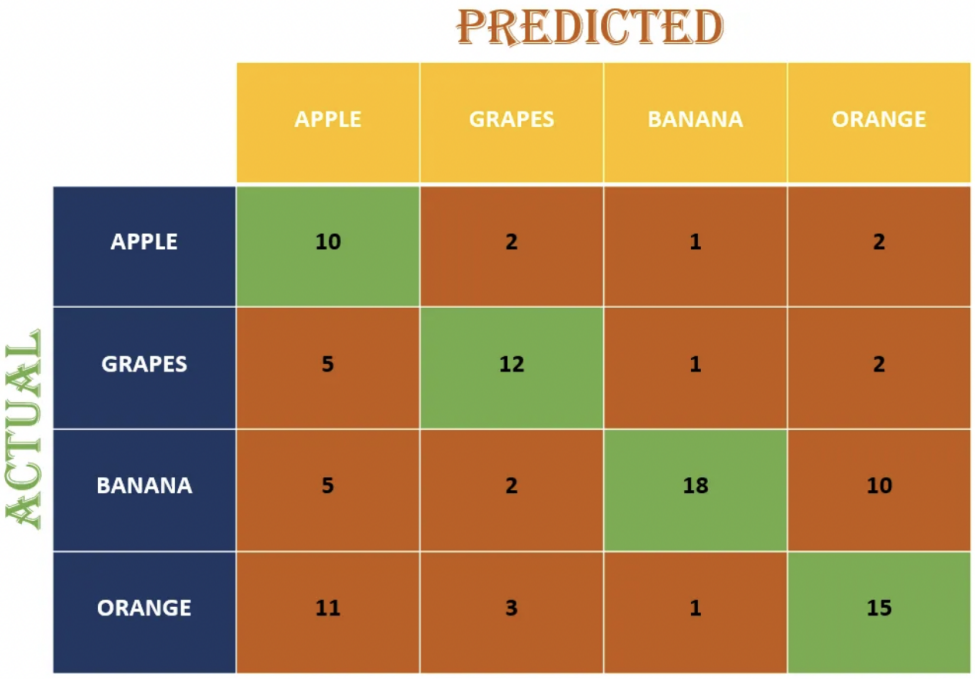
7.[Activity] Reinforcement Learning & Q-Learning with Gym

8.Understanding a Confusion Matrix

- A test for a rare disease can be 99.9% accurate by just guessing no all the time.
- We need to understand true positives and true negative, as well as false positives and false negatives.
- A confusion matrix show this.



- Multi-class confusion matrix.

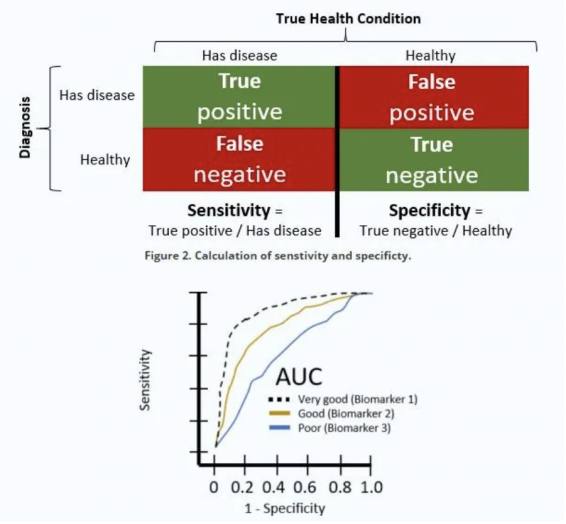


4 x4 Confusion matrix for fruit classifier

9.Measuring Classifiers (Precision, Recall, F1, ROC, AUC)

- **Precision (FP).**
  - TP/TP+**FP.**
  - AKA Correct Positives.
  - Percent of relevant results.
  - Good when you care a lot about false positives. Ex: drug testing, medical screening.

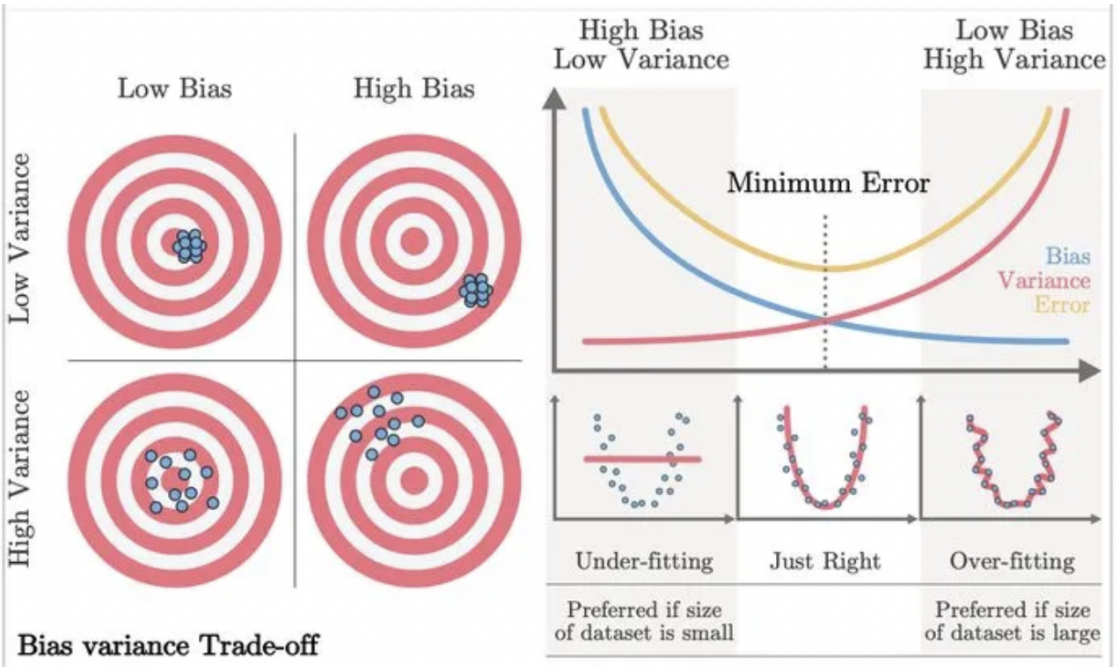
- **Recall (FN)**
  - TP/TP+**FN.**
  - AKA Sensitivity, True Positive rate, Completeness.
  - % of positives correctly predicted.
  - Good when you care a lot about false negative. ex: fraud detection.
- **Other metrics.**
  - **Specificity** = TN/ TN + FP = True Negative rate.
  - **F1 Score** = 2TP/ 2TP + FP + FN. Harmonic mean of precision and sensitivity, when you care about **precision and recall.**
  - RMSE: root mean squared error. Only cares about **right and wrong answers.**
- **ROC Curve**
  - The more its bent toward the upper-left the better.
- **AUC - Area Under the Curve.**



Seção 7: Dealing with Real-World Data

1.Bias/Variance Tradeoff

- Bias = how far removed are from real answer.
- Variance = how scattered are from real answer.



- Error is what we care about and want to reduce. Error = Bias2 + Variance.
  - It's what we want to minimize, not bias or variance specifically.

2.[Activity] K-Fold-Cross-Validation to avoid Overfitting

- One way to further protect against overfitting is k-fold cross validation.
  1. Split your data into K randomly-assigned segments.
  2. Reserve one segment as your test data.
  3. Train on the combined remaining K-1 segments and measure their performance against the test set.
  4. Repeat for each segment.
  5. Take the average of the K r-squared scores.

3. Data Cleaning and Normalization

- The reality:
  - Outliers.
  - Missing Data.
  - Malicious Data.
  - Erroneous Data.
  - Irrelevant Data.

    - Inconsistent Data.
    - Formatting.
- Always look at your Data!

4.[Activity] Cleaning web log data

5.Normalizing Numerical Data

- Some models may not perform well when different attributes are on different scales. Examplo: Ages 0 - 100, Money 0-Billions
- Bias in the attributes can also be a problem
- Most data mining and ML techniques work fine with raw, un-normalized data
    - Remember to re-scale when done

6.[Activity] Detecting Outliers

7. Feature Engineering and the Curse of Dimensionality

- Applying our knowledge to create better features to train our models
- Feature Engineering is selecting the features most relevant to the problem
- Curse of dimensionality:
    - too many features can be a problem
    - every feature is a new dimension
    - To distill many features into fewer features, we can use unsupervised dimensionality reduction techniques (PCA, K-Means)

8.Imputation Techniques for Missing Data

- Mean Replacement
    - Median may be a better choice when outliers are present
    - Only works on column level
    - Can’t use on categorical features
    - Not very accurate

```
import panda as pd
masses_data = pd.read_csv('')
masses_data.head()

mean_inputed = masses_data.fillna(masses_data.mean())
mean_inputed.head()
```

- Dropping
  - If not many rows contain missing data
  - It’s never going to be the best approach
- Machine Learning
  - KNN: Find K nearest (most similar) rows and average their values
    - Assumes numerical data, not categorical
  - Deep Learning:
    - Build a ML model to impute data for your ML. Works with categorical data really well, but its complicated.
  - Regression
    - Find Linear or non-linear relationship between the missing feature and the others.
    - Most advanced technique: MICE - Multiple Imputation by Chained Equations
- Get More Data

9.Handling Unbalanced Data: Oversampling, Undersampling, and SMOTE

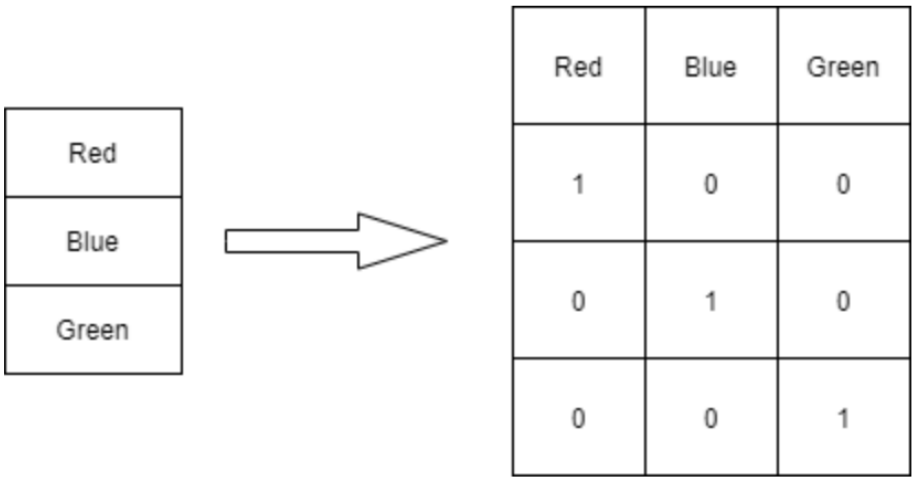
- What it's?
  - Large discrepancy between
  - ex: Fraud is rare, and most rows will not be not-fraud
  - Mainly a problem with a Neural Network
- Oversampling
  - Duplicate samples from the minority class

    - Can be done at random
- Undersampling
    - Remove negative samples
    - It’s usually not the right answer
- Smote
    - Synthetic Minority Over-sampling Technique
    - Artificially generate new samples of the minority class using nearest neighbors
        - run KNN of each sample of the minority class
        - Create a nem sample from the KNN result
        - Better than just oversampling
- Adjusting Thresholds
    - If you have too many false positives, simply increase that threshold
    - Reduce false positives, but could result in more false negatives

10.Binning, Transforming, Encoding, Scaling, and Shuffling

- Binning
    - Bucket together based on ranges of values. Ex: estimated ages of people, 20..., 30...
    - Quantile binning categorize data by their place in the data distribution.
    - Transformis numeric data to ordinal data
    - Especially useful when there is uncertainty in the measurements
- Transforming
    - Feature Data with an exponential trend. Ex: Youtube recommendations
- Encoding
    - Very common in Deep Learning
    - Transforming data into some new representation required by the model
    - One-hot encoding

        - Create buckets for every category, the bucket for your category has a 1, all other have 0



One Hot Encoding

- Scaling/ Normalization
    - Normally distributed around 0 (most neural nets)
        - Otherwise features with larger magnitudes will have more weight than they should.
    - Scikit Learn has a preprocessor module that helps (MinMaxScaler)
    - Remember to scale your results backup
- Shuffling
    - Many algorithms benefit from shuffling their training data
    - Otherwise they may learn from residual signals in the training data resulting from the order in wich ther were collected

Conclusão

In summary, advanced data mining and machine learning techniques are essential for optimizing models and improving outcomes. Section 6 highlights methods like K-Nearest Neighbors (KNN) for classification and PCA for dimensionality reduction, as well as Reinforcement Learning, which uses

rewards to train agents. Evaluation tools such as the Confusion Matrix and metrics like Precision and Recall are crucial for assessing model performance.

Section 7 addresses real-world data challenges, focusing on the balance between bias and variance for accuracy. Techniques like k-fold cross-validation help prevent overfitting, while data cleaning and normalization manage outliers and inconsistencies. The section emphasizes feature engineering and methods like SMOTE, binning, encoding, and scaling for preparing effective models. Overall, these strategies enhance data-driven decision-making.

Referências

Curso: Data Science and Machine Learning. Disponível em: https://www.udemy.com/course/data-science-and-machine-learning-with-python-hands-on/learn/lecture/15090172#overview. Acessado em: 23/09/2024.

Artigo: **K-Nearest Neighbors(KNN): Entendendo o seu funcionamento e o construindo do zero**. Disponível em <https://share.atelie.software/k-nearest-neighbors-knn-entendo-o-seu-funcionamento-e-o-construindo-do-zero-a21b022acd6f> Acessado em: 18/09/2024

Artigo: **Confusion Matrix: Performance Evaluator of Classifier**. Disponível em <https://faun.pub/confusion-matrix-performance-evaluator-of-classifier-ac60325c88bb> Acessado em: 18/09/2024

Artigo: **What is the AUC — ROC Curve?.** Disponível em <https://medium.com/computer-architecture-club/what-is-the-auc-roc-curve-47fbdcbf7a4a> Acessado em: 19/08/2024

Artigo: **Bias Variance tradeoff in Machine Learning**. Disponível em <https://medium.com/@sujathamudadla1213/bias-variance-tradeoff-in-machine-learning-a1856b55f6a9> Acessado em: 19/08/2024

Artigo: **Different types of Encoding**. Disponível em <https://ai-ml-analytics.com/encoding/> Acessado em: 23/09/2024