Sumary M3 Salome

Software comparison:

• R/ R-stuido is used mostly for statistical inference and testing

• libraries for statistics, making it ideal for tasks where deep statistical analysis is required

• environment for data manipulation, plotting, and statistical computation with specialized statistical libraries

• R's visualization tools are especially powerful when doing exploratory data analysis

• lacks the robust deep learning and AI ecosystem

• Python is more powerful for machine learning

• flexibility and ease of use in machine learning, AI, and deep learning.

• libraries like scikit-learn (for traditional machine learning), TensorFlow and PyTorch (for deep learning), and Keras have established Python as the go-to language for modern AI applications.

• handling large datasets and integrating with other systems (like web services or databases) makes it highly suitable for machine learning projects that go beyond pure analytics

Macro areas of machine learning algorithms: Supervised vs unsupervised machine learning

1. Supervised Machine Learning:

trained on labeled data, where both the input (features) and the output (target) are known, used to learn a mapping from inputs to outputs that can be applied to unseen (future) data.

Model performance can be evaluated using metrics such as accuracy, precision, recall, or root mean squared error (for regression).

• Regression models

• Decision trees

• Random forests

To measure and evaluate accuracy and precision:

Difference between accuracy and precision:

• Accuracy takes into account predicitions from all classes, to measure how many are correct and how many are not

• Precision looks at a particular class only (eg. Class 1 only)

• As you have multiple classes, you have one accuracy and as many precision values as you have classes

Generalization:

To measure model performance in an unbiassed way, we need to use different data to test the model than the data that the model was trained on.

For this we use the 'train-test' split

• Information leakage: train and test data set should not have correlated observations. Case of aircrafts at airport, if one flight is delayed, the next one from this aircraft is most likely as well. So splitting it you should be careful

• With time series data, don't randomize into test and training set by observation from all the years, instead use the years to do the split ( do not use default shuffle = TRUE for the split)

• Time series data should not be shuffled and randomly split, time series data requires careful consideration to avoid data leakage and to reflect the temporal structure

• Stationary data, you can typically use more flexible splitting methods

§ Stationarity: stationarity refers to a property of a dataset where the statistical characteristics of the time series (such as mean, variance, and autocorrelation) do not change over time

• Non-stationary data has time-dependent characteristics, such as trends, seasonality, or varying variance, making it more challenging to model.

§ Sequential Train-Test Split is critical, Randomly splitting the data would lead to data leakage

§ Rolling Window: Train on a fixed window size (e.g., last 5 years) and make predictions for the next period (e.g., next 1 year), then slide the window forward.

Source of error: Performance of a model is evaluated by its error, which is broken into two sources:

Folowing two components capture different aspects of a model's ability to generalize to new data, understanding the trade-off between bias and variance is crucial

• Variance: model's sensitivity to fluctuations --> high variance occurs when model is too complex and fits the training data too closely, capturing noise along with the true signal, leads to overfitting the model

• Bias: (systematic) distance from the ground truth --> high bias occurs when model is too simple and unable to capture the underlying patterns in the data, leads to underfitting the model

bias-variance trade-off: is a fundamental concept in machine learning, describing the tension between a model's ability to fit the training data and its ability to generalize to new data

Cross-Validation:

• involves splitting the dataset into multiple subsets and training the model on some subsets while validating it on others

• helps in getting a reliable estimate of the model’s performance and helps avoid overfitting.

Here you split the data set into:

• training set: learns patterns and relationships from this data.

• validation set: used to tune hyperparameters and evaluate model performance during training, can be used to select the best model configuration

• test set: used to provide an unbiased evaluation of the final model’s performance

Capacity of a model: how much information can the model ingest (number of functions that it can fit)

• Linear model: low capacity as just lineaar functions can be incorperated

• Polynomial models: higher capacity

• Neural network, random forests and and trees can fit many different functions and thus have a high capacity

Working with data sets:

In machine learning models, categorical variables generally need to be converted into numeric representations because most machine learning algorithms work with numerical inputs.

Algorithms like decision trees, random forests, and deep learning models require numerical data to perform mathematical operations during the learning process.

• Label Encoding: assign a unique integer

• Ordinal Encoding: where there is a natural order (e.g., low, medium, high), categories are assigned numeric values that respect this order.

Certain models are able to deal with NA's and certain models are not

Most machine learning models cannot handle missing values (NA values) directly, and you will typically need to deal with them during data preprocessing.

If you feed a model with missing data, it may result in errors or poor model performance

• Remove Missing Data (Dropping NA values):

• Imputation

• Mean imputation

• K-Nearest Neighbors (KNN) Imputation

• Advanced methods such as regression imputation

scikit-learn module

extensive documentation with python examples https://scikit-learn.org/stable/user\_guide.html

module contains many different machine learning methods

Performance measure

• Assess models by its RMSE:

• Calculate the RMSE based on your training set ( 80% split)

• Calculate the RMSE based on your test set: as the test rmse is much larger than the training msre, than means the model is not suitable to generalize the data

• The RMSE from both training and test set should be similar (where the difference between the two with respect to the scale of the y-axis matters)

§ Train rmse = 3, test rmse = 5, y scale of 0-50, the errors can be seen as similar

§ Train rmse = 3, test rmse = 5, y scale of 0-10, the model is overfitted to the training data set and not suitable for generalizing data

Before assessing ML models, it helps to think about a baseline to have a reference accuracy value.

Such a baseline could be as simple as the sample mean of the outcome variable or the RMSE of a simple linear model.

Any complex model aims to improve the baseline

ML techniques:

1. Logistic regression

logistic regression models the probability that Y belongs to a particular category.

So the logistic regression predicts a probability value between 0 and 1 for an outcome Y.

The logistic regression belongs to the linear models and is mostly used for binary classification, where the goal is to predict one of two possible outcomes.

It is generally used for binary classification problems, but there are extensions like multinomial logistic regression that exist for more than two classes.

Assumptions:

• works best when the dataset is sufficiently large

• No multicollinearity

• Independance of observations

Extention: multiclass classification problems

Multinomial logistic regression: directly handles multiple classes without breaking the problem into multiple binary classifiers.

It is used to compute the probabilities of each class, the model tries to estimate the parameters (weights) that maximize the likelihood of the correct class for each training example.

As data points are plotted, the boundary lines will be linear between classes, as it is a linear model

2. Decision Tree

A decision tree is a supervised machine learning algorithm used for both classification and regression tasks.

The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

The tree selects a feature that best separates the data based on a splitting criterion. Then the dataset is split into subsets at each node according to the selected feature.

A metric called the Gini impurity index can be used to quantify how messy or clean a dataset is, especially useful when we use decision trees to classify the subsets.

The gini index measures how "pure" a node is, so how mixed the classes in a node are. If all observations belong to the same class the gini impurity will be 0, if classes are perfectly mixed, the gini impurity will be 1.

Assume a data set with Gini index of 0.650. As it is split into two nodes A with index 0.400 and B with 0.100, we gain as the sum of A and B is lower and thus better than at the initial node.

The closer to 0, the better is therefore the classification at the node. The sum of the Gini Impurity score at the nodes should be lower than the Gini coefficient at the node above.

As it does not improve, no further splits should be done.

Once the tree is built, it is used to predict outcomes by navigating through the nodes based on input features

Benefits: capture non-linear dependencies, and require small amount of data, no need for data preprocessing (normalization and scaling of the data),

Provides insights into which feature are most important in determining the outcome

Trees can easily handle qualitative predictors without the need to

create dummy variables

Disadvantages: likely to overfit, bias toward dominant features, not optimal for continuous variables, can be non-robust as changes in the data set might easily change final estimated tree

Tree based methods:

The goal is to segment the data set into different regions (classified regions) to make a prediction for a given observation.

- Decision trees for regression problems:

• A regression tree is fit to the data, where a series of splitting rules is applied based on the features of the data set

• Leaving the data set with distinct non-overlapping regions

• To predict the outcome of an observation, the tree is used to assign it to a specific region, of which the mean is taken as the predicted value

• Resulting tree might be too complex leading to overfitting such that test set performance is poor

• With a smaller tree, the variance can be reduced at the cost of a larger bias, as the number of regions is reduced. Only continue with the splitting until a threshold is reached to which the error gets reduced

- Decision trees for classification problems:

• Similar to the regression tree, but instead of a quantitative response the classification tree is used for a qualitative response.

• In the regression tree, the mean of the region an observation is assigned to is used for the prediction, for the classification problem the observation is assigned to the most occuring class within a region

• The impurity of a region is measured as the fraction of training observations that do not belong to the class identified as the most common within that region

If there is a highly non-linear and complex relationship between the features and the response then decision trees may outperform classical linear (baseline) approaches

Random forest

Decision tree is never used alone, it will be applied as many trees build a random forest

A random forest is an ensemble of decision trees (usually 500-1000 trees) that is used for prediction and classification.

The algorithm first generates multiple random subsets from the original data set (bootstrap samples).

These data sets are created by randomly selecting observations with replacement from the dataset, meaning each time a data point is chosen it is taken from the

Full data set, leading to the possibility of a data point to occure multiple times within a subset

Number of splits or depth should maximally be number of variables/features in the data

Grid Search Cross Validation to find the hyperparamethers

hyperparamethers : parameters that are not learned from the data but set by the user before training

At each split within a tree, a random subset of features is used to ensure that not a single feature dominates the predictions across all trees.

After building multiple decision trees, the Random Forest aggregates the predictions.

- For classification: Each tree makes a prediction for the input data, and the class with the most votes (majority vote) across all the trees is the final prediction. The benefit of Random Forest is that by counting the “votes” of each member tree, the prediction tends to be much more stable than the prediction from any single decision tree.

- For regression: The predictions from all the trees are averaged to provide the final prediction.

It is this averaging across trees that makes Random Forest so successful at reducing the high variance of a decision tree

Random forest accoring to the book:

A number of decision trees is built based on the training sample. However, at every split a random sample of features is chosen as possible "candidates" from the ful set of features.

The split is then made based on one of the features

The sample of features that are cosidered at a split is approximately equal to the square root of the total number of predictors.

Without this reduction of considered features it might be that one very strong feature will use this one in the split, such that the trees all look alike (correlation of the trees)

By forcing each split to consider only a subset of the features, the trees can be decorrelated such that the predicition becomes more reliable and that the variance is reduced

(with averaging correlated results lead to a much worse of a reduction in variance compared to uncorelated averaging)

A Feature Importance Plot is a visualization that ranks and displays the importance or contribution of input features (or variables) in a predictive model. It helps in understanding which features are most influential in predicting the target variable and which are less so. This can guide decision-making around feature selection, model interpretation, and improvement.A Feature Importance Plot is a visualization that ranks and displays the importance or contribution of input features (or variables) in a predictive model. It helps in understanding which features are most influential in predicting the target variable and which are less so. This can guide decision-making around feature selection, model interpretation, and improvement.

Can be calculated using several methods, one of which is Permutation importance: Measures how performance decreases when the values of a feature are randomly shuffled.

While this is used for example Xgboost or logistic model, for random forest it is not necessary.

For Random Forest models, an other tree-based type of feature importance is stored in the model attribute. This is much faster to plot than the permutation feature importance.

To have more trust in the result of a random forest, the permuation importance can be done to see if the results of feature importance aligns

Boosting:

Here trees are grown sequentially: meaning that each tree is grown using information from previously grown trees.The construction of each tree depends strongly on the trees that have already been grown.

Thereby the boosting approach learns slowly.

After each tree, more focus is placed on the data points that were misclassified (or had the most error). This forces the new tree to improve where the previous ones failed.

XGBoost

2. Unsupervised ML: cluster observation in groups to understand common characteristics, explorative analysis to discover unknown subgroups in the data, ways to measure the quality of the clusters/clustering in unsupervised ML, explore patterns in data

trained on data that is unlabeled, meaning the output / y variable is not known. The goal is to identify patterns, structures, or relationships within the data

Unsupervised learning techniques can be applied to:

- Explorary data analysis: with only a few features, the data can be visualized in simple plots

- Understanding the data: highlight data structure and outliers

- Noise reduction: reduce noice by dimensionality reduction -> remove features that are providing information

- Data compression: model fitting time is reduced with lower features without losing important information

Non-linear methods: t-Sne and UMAP

• Clustering

• Anomaly detection

• Principal component analysis PCA

Clustering:

Clustering techniques divide the set of data into group of points having common features

K-means clustering:

each group gets assigned a cluster center, as a mean representative point

cluster centres are chosen in order to minimize the following loss function:

From <https://colab.research.google.com/github/neworldemancer/DSF5/blob/master/Course\_3.ipynb#scrollTo=wOaFNbi1effq>

From <https://colab.research.google.com/github/neworldemancer/DSF5/blob/master/Course\_3.ipynb#scrollTo=5XsYvuoUeffq>

3. Reinforcement learning:

There is a goal to reach or task to accomplish such as winning a chess game, model explores possibilities to reach this goal and is guided by a reward function

Normalize the data features to have 0 mean standard deviation 1 to make sure that the magniture of the variables does not impact the distance calculated in clustering