CORE Skills

Delivering Data Science
In Resources & Energy

Machine Learning I

Day 8

Fundamental concepts and supervised techniques

Dr Leonardo Portes dos Santos & Dr Thomas Steml

Senior Research Fellow ARC ITTC Data Analytics for Resources and Environments/Center for Exploration Targeting

Senior Lecturer ARC ITRH

Thomas.Stemler@uwa.edu.au
Leonardo.portesdossantos@uwa.edu.au

A CORE partnership

Program Timeline

1 day Leading Data Scientists Executive Program	Data Science Springboard Technical Preparatory		Introduction to Data Projects	Data Analysis			Data & Communication Sandbox	Data Fusion and Machine Learning	Data Fusion and Machine Learning	Data Fusion Sandbox	Special Data Types - Time-series Data	Special Data Types - Natural Language Processing and Text Mining	Special Data Types - Spatial Data	Capstone Project Development & Presentation	Capstone Propeller
1/2-1 Day*	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15
Enabling your people's data science capability in context of the 15 day program	Introduction to the program tools & set up	Introduction to the program tools & set up	Zero to Data Science in a day	Getting to know the program tools - data munging and exploratory data analysis	Simple predictions - regression and statistical model building	Multivariate analysis and model building	Effective data storytelling - communicating results to non-technical audiences	Pros and cons of commonly used statistical and machine learning techniques I	Pros and cons of commonly used statistical and machine learning techniques II	Sandbox - Consolidate approaches covered and test on datasets	The 4th dimension and predictions	Finding needles in wordstacks	Spatial analytics and predictions	Pitching Capstone Projects	Project Review Day

Plan of the day

AWST	AEST	Agenda	Educator
08:00	**10:00**	**Q&A, Preparing Jupyter Notebook**	
08:15	10:15	The Machine Learning Landscape	Thomas
09:45	*11:45*	*Morning tea*	
10:00	12:00	Supervised techniques	Leonardo
11:30	*13:30*	*Lunch*	
12:15	14:15	Capstone Project Update/Share	Tamryn/All
12:45	14:45	Evaluating the ML model	Thomas
13:45	*15:45*	*Afternoon tea*	
14:00	16:00	More about ML	Leonardo
15:15	17:15	Closeout – Reflections, Takeaways	
15:25	17:25	Menti Feedback	Tamryn
15:30	**17:30**	**Close**	

Aims & Learning Outcomes – Day 8

Aims

1. Introduce fundamental concepts of supervised ML and what are the crucial steps in an end-to-end machine learning strategy from data.
2. Explore traditional supervised classification techniques and the evaluation metrics for error analysis.
3. Introduce open-source toolboxes and packages that can be applied to mining/energy data.

Learning Outcomes

1. Understand the key steps to develop a machine learning investigation from data.
2. Understand how to evaluate, validate and prevent problems such as overfitting and underfitting.
3. Understand which techniques are best suited for a ML problem and how to design the learning model.
4. Understand the advantages and disadvantages for the techniques to select the correct type of model given available data.

Day 8 The Machine Learning Landscape

Day 8 What is Machine Learning?

Day 8 What is Machine Learning?

- Arthur Samuel (1959) addressed the Machine Learning as a field of study that gives computers the ability to learn without being explicitly programmed.

- Machine learning is the science (art) of programming computers so they can **learn from data**. Data or examples that the system is using to learn are called “training set”, each training example is called “training instance or sample”. (*Aurelien Geron)*

- A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. (*Tom Mitchell)*

Learn from experience

Learn from data

Day 8 Artificial Intelligence vs Machine Learning vs Data Mining

Day 8 Artificial Intelligence vs Machine Learning vs Data Mining

- DM: applying techniques to dig into large amounts of data to discover patterns that were not immediately apparent.

- ML: goes beyond DM and provides the system the ability to build a trained model that can predict and compare new data or situations based on the training set.

- AI: goes beyond ML and provides the system the ability to take decisions so the model can create associations among events and situations without ever having seen such patterns or data before.

Day 8 Why is Machine Learning great?

Day 8 Why is Machine Learning great?

- Traditional approaches are based on writing rules while ML is based on training from examples and data.
  - Explicit rules and instructions are replaced by finding patterns in the data.
- ML algorithms can find solutions for problems that either are too complex for traditional approaches.
- Learn and adapt to changes and new data.
- ML can help humans learn to get insights about complex problems and large amounts of data as they can be inspected to see what they have learned.

"If a typical person can do a mental task with less than one second of thought, we can probably automate it using AI either now or in the near future." — **Andrew Ng**

Day 8 WorkFlow of a Typical ML Project

Traditional approach

ML approach

Source:

Day 8 WorkFlow of a Typical ML Project

Source:

Day 8 Some types of ML models

- Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a prediction model:
  - **Instance-based learning:** The new instances must be identical to the training instances or very similar (requires similarity measure). *K-nearest neighbour algorithm*
  - **Model-based learning:** Using a model of examples to generalise the prediction of new instances. *Support Vector Machine*

Source: Hands on machine learning with Scikit-learn and TensorFlow

Day 8 Some types of ML models

- Whether or not they can learn incrementally from a stream of incoming data:
  - **Batch learning (offline learning):** Time consuming and requires a lot of computing resources.
  - **Online learning:** Feed new data instances sequentially, fast and cheap. A big challenge is that if bad data fed to the system, the performance gradually declines.

Source: Hands on machine learning with Scikit-learn and TensorFlow

Day 8 Some types of ML models

- Whether or not they are trained with human supervision:
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning: The agent (the learning system) observes the environment to get rewards or penalties in return to learn by itself the best strategy (policy).

Source: Hands on machine learning with Scikit-learn and TensorFlow

Day 8 Supervised vs Unsupervised Learning

- **Supervised learning:** the training data we feed to the algorithm includes the desired solutions (**called labels**). Typical supervised tasks:
  - Classification
  - Regression

- Most important algorithms:
  - K-Nearest Neighbours (K-NN)
  - Linear regression
  - Logistic regression
  - Support Vector Machines (SVMs)
  - Decision trees, Random forests, Gradient Boosting Trees
  - Neural networks

- **Unsupervised learning:** the training data is unlabeled (doesn't include a desired solution), the system tries to find structure in the data without using labels. Typical unsupervised tasks:
  - Clustering
  - Dimensionality reduction

- Most important algorithms:
  - K-means clustering
  - Expectation Maximization
  - Gaussian Mixture Models (GMM)
  - Hierarchical cluster analysis (HCA)
  - Principle Component Analysis (PCA)
  - Kernel PCA
  - Locally linear embedding

https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Day 8 ML in industry: some examples

Examples of classification/regression tasks:

Day 8 ML in industry: some examples

Examples of classification/regression tasks:
From industry partners in CORE Skills Pilot:

- Used linear regression to check for unaccounted for total gas and whether it is correlated to maximum temperature or wind speed.

- Used geochemical data to predict domains for ore/material characterisation. Need to look at if variation in classes unbalances predictions (ore oversampled relative to other lithologies?) geochemical data was able to select clusters as we had a lot of data. Trimmed X fields with missing values - 12 fields we retained including from and to downhole intercepts. Some fields correlated with each other.

- Iron ore fluctuations of tonnage yield - predict from logging and geochemical data, missing data and log transform %, create ML pipeline to take account of the transformation, reduce the features were important to go from 200 to 20 - and used supervised learning model to reduce variables which were correlated with each other.

- Calcination and superfine - learned how to use temperature, 30 days of energy usage, used iron ore example from week 4 - labelled as good/bad operations then applied KNN - very encouraging initial results but not a lot of data in small test - was able to replicate the experts opinion - with rules (kind of a feature extraction).

Thomas and Leonardo (UWA)

Day 8 Some limitations and challenges in ML

Day 8 Some limitations and challenges in ML

- **Lack of good data**: Most of ML models requires sufficient amount of data for efficient training and better generalisation.
- **Non-representative or lack of training data (labeled data)**: it is crucial that the training data be representative of new data. Sampling method can be flawed (**sampling bias**).
- **Poor quality data:** full of errors, outliers or noise, such data need to be cleaned before use.
- **Irrelevant features:** ML requires a good set of relevant features to make appropriate associations.
- **Interpretability and complexity**. Some models have too many hyper-parameters to tune.
- Tendency to use ML is a ”**black-box**.”

Day 8 Some limitations and challenges in ML

Statistical fit

- **Over-fitting the training data:** the model performs well in the training data but does not generalise well.

- **Under-fitting the training data:** the model is too simple to learn the underlying structure of the data.

ML general workflow

What are the main steps of the ML workflow?

ML general workflow

Day 8 Exercises

Scenario 1 (rock type problems):

- **Part 1:** Your company has just taken over a new operating mine and you don't trust the old operator's classifications or rock models. You have a bunch of photographs of core and a pile of core trays. The asset manager has asked you to take a look at the photographs and determine roughly how many different types of rock you've got.
  - Questions: what sort of problem is this? What features could you extract from the data?

- **Part 2:** You've taken a subset of exemplar rock types from your clusters and validated these with your geoscience function at head office who have given you some labels for each of the rock types. What approaches could you take to leverage this information to predict rock types across all the photographed core.

- **Part 3:** You've been able to send some rocks off for assay - how would you predict the composition of the remaining rocks that you didn't assay.

Day 8 Exercises

Scenario 2 (predictive maintenance problems):

You are working in the production section of an asset, and your boss has been lured into investing in a whizz-bang new vibration sensor for predictive maintenance. Your job is to work out whether you'll get any useful information out of the data.

- **Part 1:** What sorts of features could you extract from a vibration sensor? How could you use these features for maintenance purposes? What kind of approaches or algorithms would you use?

- **Part 2:** You have vibration data from failing machinery and machinery that is operating normally. What sort of problem is this? How would you approach it?

- **Part 3:** Your maintenance manager wants to be able to plan the replacement and maintenance work based on likely time to failure. What data could you collect to measure this? What sort of prediction problem is it? Could you design an experiment to determine how accurate your predictions are?

Day 8 Supervised Machine Learning: General Steps

Picture adapted from: Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap 1.

Thomas and Leonardo (UWA)

Day 8 Principal References

Scikit-learn: http://scikit-learn.org

Get prepared:
https://github.com/core-skills/08-machine-learning

Day 8 Supervised techniques: K-NN, Random Forest and SVM

Day 8 K-Nearest Neighbors (K-NN)

Use for:

- Classification and regression

Instance-based learning (model is not explicitly learned)**:**

- Idea: similar things are close to each other.
- Based on measures of similarity: a **distance metric** defines the distance between the instances in a feature space.
- Usually based on the **Minkowski distance** between two data instances $\mathbf{x}_1$ and $\mathbf{x}_2$:

$$Minkowski(\mathbf{x_1}, \mathbf{x_2}) = \left(\sum_{\mathbf{i}=1}^{\mathbf{m}} abs(\mathbf{x_1}[\mathbf{i}] - \mathbf{x_2}[\mathbf{i}])^{\mathbf{p}} \right)^{\frac{1}{\mathbf{p}}}$$

- The parameter *p* defines the behaviour of the metric. If *p = 1* we have the Manhattan distance, if *p = 2* we have the Euclidean distance.

Source: Fundamentals of Machine Learning for Predictive Data Analytics, Cap. 5

Day 8 K-Nearest Neighbors (Cont.)

Finding the nearest neighbor (K=1)

However, sensitive to noise in the data.

Decision boundary updated.

- Feature space is partitioned into a Voronoi tesselation: local partitions.
- Implicitly, a global prediction boundary is determined by aggregating the local partitions within the feature space.

Source: Fundamentals of Machine Learning for Predictive Data Analytics, Cap. 5

Day 8 K-Nearest Neighbors (Cont.)

- Decrease the dependency on individual instances (K > 1).
- Majority vote in the set of **K nearest neighbors**.

$$M(\mathbf{x}_q) = \arg\max_{c \in C} \sum_{i=1}^{k} \delta(c_i, c)$$

Day 8 K-Nearest Neighbors (Cont.)

Algorithm accounts neighbors that are too far.

- High values of K -> tendency towards the majority class, therefore do not work well for **imbalanced datasets**.

Source: Fundamentals of Machine Learning for Predictive Data Analytics, Cap. 5

Day 8 K-Nearest Neighbors (Cont.)

Data cleaning is very important in a ML project!

Distance weighted K-NN

- Weight the contribution of each neighbor w.r.t. the distance between the query and the neighbor:
  - Close neighbors -> higher weights (more relevance)
  - Distant neighbors -> less relevance.

$$M(\mathbf{x}_q) = \arg\max_{c \in C} \sum_{i=1}^{k} w_i \delta(c_i, c)$$

$$w_i = \frac{1}{d(\mathbf{x}_q, \mathbf{x}_i)^2}$$

Source: Fundamentals of Machine Learning for Predictive Data Analytics, Cap. 5

Day 8 Aspects of K-NN

Pros:

- Simple and intuitive. Update the model is easy.
- Can be used for semi-labelled data (samples together share same labels).
- Easy to extend to multiclass classification.

Cons:

- Memory-intensive.
- Expensive testing or prediction.
- Tradeoff: Small values of K -> risk of overfitting
  Higher values of K >- risk of underfitting
- Require a meaningful distance metric.
- Sensitive to skewed datasets (even using variations).
- Course of dimensionality -> usually dimensionality reduction techniques to help.

Source: https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

Day 8 Aspects of K-NN (Cont.)

- No assumptions about structure of the data.
- Retrieval of similar items – but no learning/knowledge.
- Works well when the density of the feature space in any point is fairly high.
- Sufficient domain knowledge should be available – this helps to define a meaningful distance metric.
- Noise and outliers may have negative effect.
- **Feature normalization** is important -> the larger the scale the larger the influence of the feature.

Day 8 Aspects of K-NN (Cont.)

What about feature importance?

- Irrelevant features increase costs without affecting accuracy.
- Misleading features increase costs and decrease accuracy.

Attribute-weighted K-NN

$$dist_w(x, y) = \sum_{i=1}^{m} w_i (x_i - y_i)^2$$

How to find the weights?

- You have an expert to advice you.
- Or, rely on ML!

Optimized data structures and implementations:

- Approximate Nearest Neighbor (k-d trees).
- Locality sensitive hashing (LSH) – for higher dimensions.

Exercise

- Open am1-iron-ore-dataset.ipynb and go through the the exercises related to the KNN classification.

Day 8 Decision Trees

A dataset that represents the characters in the *Guess Who* game.

Man	Long Hair	Glasses	Name
Yes	No	Yes	Brian
Yes	No	No	John
No	Yes	No	Aphra
No	No	No	Aoife

Someone picked **Brian**.
What should you ask first?

1. Is it a man?
2. Does the person wear glasses?

Sources: Fundamentals of Machine Learning for Predictive Data Analytics, Cap. 4; Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 6

Day 8 Decision Trees (Cont.)

"Does the person wear glasses?"

$$\frac{1+2+3+3}{4}=2.25$$

That's the average number of questions you have to ask per game.

Sources: Fundamentals of Machine Learning for Predictive Data Analytics, Cap. 4; Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 6

Day 8 Decision Trees (Cont.)

"Q1: Is it a man?"

$$\frac{2+2+2+2}{4}=2$$

That's the average number of questions you have to ask per game.

"On average, an answer to Q1 is more Informative that an answer to Q2."

Sources: Fundamentals of Machine Learning for Predictive Data Analytics, Cap. 4; Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 6

Day 8 Decision Trees (Cont.)

Used for:

- Classification and regression.

Information-based learning:

- *Informativeness* of a descriptive feature: split the dataset into branches that maximize the information gain, reducing the node's impurity.

- Find out which feature is more informative to ask questions about, consider the effects of different answers to these questions (how the domain will be split and the likelihood of each answer).

Sources: Fundamentals of Machine Learning for Predictive Data Analytics, Cap. 4; Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 6

Day 8 Decision Trees (Cont.)

Sources: Fundamentals of Machine Learning for Predictive Data Analytics, Cap. 4; Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 6

Day 8 Aspects of Decision Trees

Pros:

- Works well for complex datasets.
- Require very little data preparation (even scaling is not necessary).
- Example of a white box approach: decisions are intuitive.
- Make few assumptions about the structure of the data.

Day 8 Aspects of Decision Trees

Cons:

- Rely on a good tuning of restrictions (hyperparameters) to avoid overfitting – otherwise trees can just memorize the training data. Pruning can also be used.
- Sensitive to training set rotation. Option: use PCA for a better orientation of the training data.
- Sensitive to small variations in the training set.

Day 8 Ensemble Learning

Idea: Combine the predictions of many predictors to get a better final prediction.

Source: Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 7.

Day 8 Ensemble Learning (Cont.)

Bagging (Bootstrap aggregating) **and pasting:**

- Train the models on different random subsets of the training set.
- Individual predictors have high bias, but in overall aggregation produces lower bias and lower variance when compared to training one predictor.

Source: Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 7.

Day 8 Ensemble Learning (Cont.)

The ensemble's prediction tends to generalize better than a single predictor.

Bagging or pasting?

- Bagging introduces more diversity: slightly higher bias but lower variance.
- Bagging tends to produce better results.

Other approaches: Boosting (AdaBoost, Gradient Boosting), Stacking.

Source: Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 7.

Day 8 Random Forests

An ensemble of Decision Trees:

- Most commonly trained with the bagging method.
- Searches for the most informative feature in a (random) subset of the training set.
- Tree diversity flavours higher bias for a lower variance.

Feature Importance

- Most important features are likely to be near the root.
- Average depth at which a specific feature appears across all trees.
- `Feature_importance_variable.`

Source: Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 7.

Day 8 Aspects of Random Forests

Pros:

- Among the most powerful ML algorithms.
- Learn nonlinearities in the training set structure.
- Robust to outliers.

Cons:

- In each node, the training algorithm compares all features in all samples. It can be slow for larger datasets.
- Attention to the regularization hyperparameters.
- Ensemble makes prediction less intuitive.

Day 8 Aspects of Random Forests (Cont.)

- Promise technique when there are many discontinuities in the data structure (orthogonal features) – features showing almost no linear dependencies.

Identification of geological rocktypes

Source: Harvey, A. S., and G. Fotopoulos. "GEOLOGICAL MAPPING USING MACHINE LEARNING ALGORITHMS." International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 41 (2016).

Day 8 Random Forests - Exercise

Exercise

- Open am1-iron-ore-dataset.ipynb and go through the the exercises related to the Random Forests classification.

Day 8 Support Vector Machines (SVMs)

Used for:

- Classification and regression.

Large margin classification:

- It maximizes the distance to the nearest points relative to both classes.

What that means?

- More robust to classification errors, better generalization.

Day 8 Support Vector Machines (Cont.)

Hard margin classification:

- Imposes all samples are correctly classified.
- Sensitive to outliers.
- Requires a linear separable solution.

What would be the correct SVM output?

First the correct classification, then maximize the margin.

Day 8 Support Vector Machines (Cont.)

Soft margin classification:

- More flexible model.
- Balance between maximizing the margin and allowing some misclassifications.
- SVM are robust to outliers by mediating well this compromise.

How SVM control this balance?

- Hyperparameter C can play a role if the model is overfitting: the larger the value of C the more intricate or restricted the decision boundary is.

Source: Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 5.

Day 8 Support Vector Machines (Cont.)

Nonlinear classification?

$$x_2 = (x_1)^2$$

Polynomial features

<u>Kernel trick!</u>

Polynomial features

Source: Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 5.

Day 8 Support Vector Machines (Cont.)

Similarity features

Similarity function: Gaussian Radial Basis Function (RBF) with $\gamma = 0.3$

Hyperparameter γ

- Define how far the influence of a single training example reaches.
- Low values led to smoother boundaries.

$$\gamma = \frac{1}{2\sigma^2}$$

$$\phi_\gamma(\mathbf{x}, \ell) = \exp\left(-\gamma \| \mathbf{x} - \ell \|^2\right)$$

$$x_2 = \exp\left(-0.3 \times 1^2\right) \approx 0.74$$

$$x_3 = \exp\left(-0.3 \times 2^2\right) \approx 0.30$$

Source: Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 5.

Day 8 Support Vector Machines (Cont.)

Gaussian RBF Kernel

Source: Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 5.

Day 8 Support Vector Machines (Cont.)

Regression

- Reverse the objective - fit data samples <u>within</u> the margin.

Source: Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 5.

Day 8 Aspects of SVMs

Pros:

- Works well for complex small to medium-size datasets.
- Works well in high-dimensional feature spaces.
- Works reasonable well where number of dimensions is greater than number of data instances.
- Estimation of the decision boundary is conducted on a subset of points – the support vectors.
- Kernel trick allows the specification of complex decision functions.
- Generally works well on sparse data.

Day 8 Aspects of SVMs (Cont.)

Cons:

- Interpretability.

- Sensitiveness to feature scales.
- Slow for hundreds of thousands of instances.
- Regularization is essential when the number of instances is much lower then the number of features.
- Rely on the concept of distance – it has to be meaningful.
- Trickier to tune. Rule of thumb: try first linear, then Gaussian RBF and then polynomial kernel.

Exercise

- Open am1-iron-ore-dataset.ipynb and go through the the exercises related to the SVMs classification.

Day 8 Evaluating the ML models

Day 8 The bias x variance Tradeoff

Underfitting

Overfitting

Motivation

- Creating Machine Learning models is easy. However, creating ***good*** Machine Learning models is not that easy.

Motivation

- Creating Machine Learning models is easy. However, creating ***good*** Machine Learning models is not that easy.

- The only way to know how a model will generlise to new cases is to actually try it out on new cases, NOT practical.
- Evaluating a model is a core part of building an effective machine learning model.
- There are different evaluation metrics for different kind of problems.
- The choice of the evaluation metric is completely depends on the type of the model and the problem.

Day 8 How to set up your data?

- **Training set:** The subset of the data used to fit the model.
- **Development set:** (also known as hold-out cross-validation set): The subset of the data that is not used in training, but it will be used to guide choices in the learning process, e.g., fine-tune of hyperparameters.
- **Test set:** The subset of the data that is not involved in the training process - unbiased estimation of the generalization error.
- Common choices used for many years: 80%-20%-20%, 70%-30%.
- With the area of big-data: 90%-10%-10%, or even 98%-1%-1%.

Day 8 The learning curve

Plot the model performance and the number of training samples increases in time.

Which curves represents overfitting and which one represents underfitting?

Day 8 The learning curve

Underfitting
More training samples won't help.

Overfitting
More training samples might help

Day 8 K Fold cross-validation

- Split the training set into K complementary subsets. Each model is trained and validated using a different combination of such subsets.

- **Stratified K-Fold Cross Validation for imbalanced datasets:** each fold contains approximately the same proportion of each class.

Day 8 Performance Measures

Regression

$$y_p^{(i)} = h\left(\mathbf{x}^{(i)}\right)$$

- **Root Mean Square Error (RMSE)**

$$RMSE(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_p^{(i)} - y^{(i)}\right)^2}$$

- **Mean Absolute Error (MAE):** sometimes used when there are many outliers.

$$MAE(\mathbf{X}, h) = \frac{1}{m}\sum_{i=1}^{m}\left|y_p^{(i)} - y^{(i)}\right|$$

Day 8 Performance Measures (Cont.)

Classification

- Accuracy (trickier for imbalanced datasets)
- Confusion matrix

		TRUE CLASS	
		POSITIVE	NEGATIVE
PREDICTED	POSITIVE	TP	FP
	NEGATIVE	FN	TN

Example

		TRUE CLASS	
		POSITIVE	NEGATIVE
PREDICTED	POSITIVE	4344	1307
	NEGATIVE	1077	53272

Accuracy over 95% !

Day 8 Performance Measures (Cont.)

Precision

- Accuracy of positive predictions

$$precision = \frac{TP}{TP + FP}$$

Recall

- Ratio of correct positive predictions

$$recall = \frac{TP}{TP + FN}$$

F1-score

- Harmonic mean of prediction and recall
- High F1 score if both precision and recall are high.

$$1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = 2\,x\,\frac{precision\ x\ recall}{precision + recall} = \frac{TP}{TP + \frac{FN + FP}{2}}$$

	Actual = Yes	Actual = No
Predicted = Yes	TP	FP
Predicted = No	FN	TN

Example:

	Actual = Yes	Actual = No
Predicted = Yes	4344	1307
Predicted = No	1077	53272

Precision = 77%

Recall = 79%

F1-score = 78%

Day 8 Performance Measures (Cont.)

Precision/Recall tradeoff

- Increasing precision reduces the recall and vise versa.
- Ways to select a good precision/recall tradeoff

Day 8 Performance Measures (Cont.)

AUC classifier 1 = 0.96

AUC classifier 2 = 0.99

AUC Random Classifier = 0.5

The receiver operating characteristic (ROC) curve

- Common tool used to compare classifiers.
- It plots the true positive rate x false positive rate
- Area under the curve (AUC)

Day 8 More about ML: Regularization, toolboxes and further readings

Day 8 Directions for overfit/underfit

- **High bias:** performance on the training set.
  - Try a larger (or new!) set of features.
  - Try high variance ML models (K-NN, SVM, RF, NN).

- **High variance:** performance on the dev set.
  - Get more training data.
  - Try smaller set of features.
  - Try to decrease the complexity of the model.
  - K-fold cross validation.
  - Try regularization.

Day 8 Regularization

Penalize the parameters of the model (degrees of freedom) in order to produce a simpler model.

- Tradeoff between increasing the bias and decreasing the variance.

Linear regression model prediction

$$y_p = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

n is the number of features

θ_j is the jth model parameter (feature weight)

MSE cost function for a Linear regression model

$$MSE\left(\mathbf{X}, h_{(\theta)}\right) = \frac{1}{m} \sum_{i=1}^{m} \left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2$$

Source: Hands on Machine Learning with Scikit-Learn and TensorFlow, Cap. 3.

Day 8 Regularization (Cont.)

Penalize the parameters of the model (degrees of freedom) in order to produce a simpler model.

- Tradeoff between increasing the bias and decreasing the variance.

Regularized linear models

$$J(\theta) = MSE(\theta) + \lambda \frac{1}{2} \sum_{i=1}^{n} \theta_1^2$$

Ridge Regression

(Tikhonov regularization)

Fit the model keeping the model weights as small as possible.

$$J(\theta) = MSE(\theta) + \lambda \sum_{i=1}^{n} |\theta_i|$$

Lasso Regression

(uses norm)

Tends to eliminate the weights of the less important features.

- Sparse models.
- Feature selection.

Day 8 Regularization (Cont.)

$$J(\theta) = MSE(\theta) + \lambda \frac{1}{2} \sum_{i=1}^{n} \theta_1^2$$

defines how much to regularize?

–> no regularization

too big

good

too small

- **λ too big:** all weights tend to zero, result is a line through the data's mean.
- **λ too small:** small penalties and therefore overfitting can still occur.

Day 8 Some toolboxes and open libraries

- **Scikit-learn:** machine learning in python.
- **StatsModels:** Python for statistical models.
- **Imbalanced-learn:** ML for imbalanced datasets in python.
- **Shogun:** machine learning library that supports many languages.
- **Spark MLlib:** for scalable machine learning applications.
- **H2O:** For fraud and tend predictions.
- **GoLearn:** To build machine learning in Go language.
- **DeepLearn.js:** Hardware accelerate machine intelligence for javaScript.

Day 8 Further reading

- Medium website (nice tutorials for practical ML).

  https://medium.com/topic/artificial-intelligence
- Towards Data Science:

  https://towardsdatascience.com
- Oracle + Data Science:

  https://www.datascience.com/blog
- KDNuggets:

  https://www.kdnuggets.com
- Kaggle:

  http://blog.kaggle.com
- Stack Exchange:

  https://stats.stackexchange.com

Day 8 Takeaways

- Understanding the problem is the most crucial step.
- Training set should be representative of new cases to be generalized.
- And should contain relevant features.
- Devote a time to explore the structure of the data.
- Data matters more than algorithms for complex problems1.
- Noise in your data can affect learning process, enhancing the probability of overfitting.
- Simple models can give plausible results and should be tested first.
- If the "test set" contributed to any aspect during the learning process, it is not a "test set", and assessment of future outcomes might be compromised.
- If you data set is biased, your learning algorithm will produce a biased outcome as well (sampling bias problem).

1 - Halevy, A., Norvig, P., & Pereira, F. (2009). The unreasonable effectiveness of data. IEEE Intelligent Systems, 24(2), 8-12.

Day 8 Takeaways (Cont.)

- What are your takeaways?

- **Homework:**
  - Story and video class from HBR on Nokia's company-wide ML approach https://hbr.org/2018/10/the-chairman-of-nokia-on-ensuring-every-employee-has-a-basic-understanding-of-machine-learning-including-him

- **Next week:**
  - Artificial Neural Networks
  - Deep Learning
  - Working with images

1 - Halevy, A., Norvig, P., & Pereira, F. (2009). The unreasonable effectiveness of data. IEEE Intelligent Systems, 24(2), 8-12.

CORE Skills

A CORE partnership

COREHUB.COM.AU/SKILLS