**DS Grad Programme 5 (Machine Learning – Part One )**

Machine Learning - Introduction

ML – The automated detection of meaningful patterns in data

Main types of ML

* Classical ML – Simple data, clear features
* Ensembles – Focus on quality given high variability within the data
* Neural networks / Deep Learning – Complex data, unclear features, want a miracle
* Reinforcement learning – No data, but we have an environment to interact with

Supervised vs Unsupervised

* Supervised learning – Used to build predictive models
  + Classification
  + Regression
* Unsupervised learning – Used to build descriptive models (to inform supervised)
  + Clustering – similarity of locations in feature space
  + Dimensionality Reduction – enables generalisation of embedded information
  + Association – find things that appear together often

Classical statistics vs Machine Learning

* Statistics
  + Assume a link between input and output
  + Then apply restrictive assumptions on this relationship between X and Y
    - Assumptions made relating to the normality of distributions, p-values, homoskedacity etc
    - p-values are only applicable if statistical assumptions have been applied and checked (eg. normality of distributions)
* Machine Learning
  + Assume a link between input and output
  + Don’t make assumptions about the link between X and Y
    - No assumptions made about the normality of distributions, p-values, homoskedacity etc
      * p-values are meaningless, as you have not applied the assumptions underpinning them
    - Gives us more freedom to find more complex relationships
    - We then use other techniques to quantify the uncertainty created due to not making standardised statistical assumptions
      * ie. Loss functions
        + This is our empirical process for risk management
        + Computed using a gradient descent methodology which helps to calculate the minimal value of the loss function

Bias/Variance

* Variance – The degree of difference between the different estimates, depending on using different data
  + Low variance = high generalisability
  + High variance = low generalisability
* Bias – Degree of difference between your results and the ground truth
  + Low bias = high validity
  + High bias = low validity
* Aim for a model that trades off variance vs bias
  + Model that is flexible enough to minimise bias, but not so flexible that it is demonstrates high variance by being too sensitive to different data
* Under/Over-fitting
  + Under-fitting – low bias, high variance
  + Over-fitting – high bias, low variance

Pre-Processing in Machine Learning

Missing data

* Can be missing for a reason – important to consider the reason before proceeding with the following options
  + Deletion of rows/columns containing missing data
  + Imputation using mean/median/mode/regression (how?)
    - Could also be done using ML models to estimate missing values eg. KNN
  + Interpolation
    - Fill gaps using the surrounding values
      * Padding – Fill in missing data with the previous value (useful for time series)
      * Linear interpolation – Use a linear function to find the new value between the two surrounding values

Noisy data

* Check datatypes
  + Convert if needed eg. One hot encoding / categorical encoding
* Deal with outliers
  + Can use binning to help with this, to put them into equal frequency bins
  + Can use clustering to detect and remove outliers
  + Can smooth the data by fitting the data point into a regression function

Data transformation

* Feature encoding – categorical into numerical
  + Categorical (should be inherent ordering)
  + One hot encoding (also called dummy encoding)
* Feature decoding – Numerical into categorical
  + Discretization – create a discrete set of variables by binning (eg. age categories)
  + Binarization – Create a threshold to create a Boolean outcome (eg. Adult/child)
* Feature scaling – Different features using different scales
  + Simple feature scaling
    - Normalisation
      * Standardisation (Z-score Normalization) – StandardScalar()
        + Subtracts the mean and divides by the SD
        + Scales to ensure all features have a mean of 0 and a SD of 1, to fit the distribution to the normal distribution
      * Min Max Scaling (Normalization) – MinMaxScalar()
        + Subtracts the minimum value and divides by the range, to scale down features to within 0-1.
        + Scales to standardise the Euclidean distance from the centre of the data space
    - Both of these are sensitive to outliers, as all data points are included
  + More complex feature scaling
    - Robust scaling
      * Subtracts the median and divides by the interquartile range
      * This makes it more robust to outliers
  + Which algorithms require scaling?
    - Do need scaling
      * Algorithms that take some form of distance measurement between the different data points into account
      * Eg. k-means clustering, k-nearest neighbours, PCA
    - Don’t need scaling
      * Algorithms that don’t use distance measures between the data points in their calculations
      * Eg. Naïve Bayes, Linear models, Tree based models
  + Which to use?
    - Standardisation
      * Good when the data is normally distributed
    - Min Max Scaling
      * When the data is not normally distributed, or you are unsure of the distribution
      * Useful when algorithms use distances between data points to make calculations
    - Robust scaling
      * When you need to deal with outliers
  + When to scale the data?
    - Do this after data splitting to prevent data leakage
      * Ensures independence of the training set
      * If you scaled it before splitting, then it would be learning different relationships between the variables that reflected the scaled values, rather than the real values

Data reduction

* Aim to retain as much data as possible
* Reducing data can help:
  + accuracy, interpretability and speed of computation
  + to reduce overfitting

Feature Selection

* EDA – Helps you identify which features are going to be most relevant to the problem
* Formal methods
  + Embedded method
    - Methods for feature selection embedded within the models themselves
    - Examples
      * Lasso/Ridge Regression
  + Filter method
    - Relevance of features determined by the statistical properties of the features themselves
      * Distributions/skew of variables
      * Correlation between variable and DV
        + Use scatter plots for continuous variables
        + Use stacked box plots for categorical variables
    - Examples
      * ANOVA
      * Pearson’s Correlation
      * Variance thresholding
      * Feature importance measures
  + Wrapper method
    - Evaluate the performance of the model by using a subset of features
      * Take variables out systematically and test the model again and again to look for differences in performance to see the effect of different combinations of features
    - Examples – different ways of selecting different combinations of features
      * Forward selection
      * Backward selection
      * Stepwise selection

Dimensionality Reduction (unsupervised learning)

* Most common techniques
  + PCA (Principal Component Analysis)
  + SVD (Singular Value Decomposition)
  + T-SNE (t-distributed Stochastic Neighbour Embedding)
* These can be evaluated to see how much information is retained

Data partitioning

* Shuffle data first to ensure both sets are representative
* 80/20 most common split
* Validation set – can be created as a subset of the training data
  + Can be used for tuning your hyperparameters

Regression

* Regularisation
  + How to balance bias/variance trade-off in regression
  + This helps deal with the tendency towards overfitting (low bias) within regression models
* Regression techniques prioritise the limiting of bias (getting as close to the ground truth as possible)
  + As a result have the potential for high variance (lack of generalisability, delivering differing performances on unseen data).
  + This is particularly the case when the data exhibits multi-collinearity (highly correlated IVs) or when there is a particularly high number of features relative to observations (high model complexity).
* Regularisation addresses this issue by adding a penalty term to the regression calculation to add in some bias (thereby lowering variance, as bias and variance are counteractive forces)
  + This will likely improve the loss function metrics eg. MSE (Mean Squared Error) score on the test/validation dataset.
* Ridge regression and Lasso regression create this penalty term in different ways.
* Ridge Regression
  + Ridge regression shrinks the impact of the coefficients (“m” gradient component for each individual variable in the form y = mx + c) towards zero via the penalty term.
    - This is particularly impactful on those regressions with a very steep slope.
    - This leads to lower co-efficient values and a limited impact by those high slope coefficients, thereby limiting the variance that was previously caused by the higher coefficients.
* Lasso Regression
  + Lasso regression takes a similar approach to Ridge Regression, however it also allows for the shrinkage of the coefficient values to 0.
  + This makes it additionally useful for feature selection, as it allows you to identify features that will have no application in real life (lack of generalisability), as there is no linear relation between the feature and the DV itself once the variance has been increased (via the penalty term) to reflect the likely variance in real world data.

Quality Assurance

There are a number of different ways that we can attempt to improve the quality of our models

Volume of Data

* More data
  + Larger samples can be beneficial
    - More unprocessed data eg. Surveys
    - More labelled data
    - Use both of these together in semi-supervised learning
* More features
  + Can collect them ourselves (primary research)
  + Create them by combining existing features
  + Can join them, by linking to existing data elsewhere
  + Key question – is there evidence that more features will improve model performance?
* Improving a model
  + Improve accuracy (predictive power)
  + Improve confidence in model (greater certainty about predictions)
  + Improve robustness (generalisability)
* Data Quality
  + Additional data can be uninformative (though may help with representativeness) or harmful (multicollinearity)
  + Quality of data is key
    - Accurate labelling
    - Informative features
    - Representative samples
    - Limited missing data

Simplicity vs Complexity

* A model is usually comprehendible if:
  + Number of coefficients/output values of the model is not too large
  + Number of features is not too large
  + Learning algorithm is well understood
  + Features are interpretable (no dimensionality reduction)
* Simple models
  + Benefits
    1. Easier to interpret, audit and explain predictions
    2. Can be more robust (generalisable) and less likely to overfit
    3. Can perform well with less training data
    4. Can often approximate complex relationships well
  + Limitations
    1. May miss more complex relationships between the data, causing weaker performance
* Complexity is created by:
  + Model choice
  + Format of the data
* Must seek a balance between simplicity and complexity
  + Defining this balance will be done dependent on the context and in conjunction with stakeholders
  + Solutions
    1. Set a minimum predictive performance and find the simplest model that reaches this threshold
       - This threshold needs to be established in conjunction with the business area
       - What is the tolerance for risk and incorrect predictions?
    2. Find the best performing model that meets simplicity constraints given
       - Are there hard limits on the complexity of the model?
         * Is full understanding required?
         * Who needs to understand the model?
         * Is the model being used for inference or prediction?
         * Do predictions have to be linked to features directly or indirectly?
  + Issues with implementing these solutions
    1. Constraints may exist in relation to the amount of data available
       - Small data sets + complexity = overfitting
    2. If there have been no constraints set on simplicity it is still essential to document what the simplicity constraints were
    3. May be no easy way to establish a minimum predictive performance threshold
       - If the model is used in a non-critical context, then it might be that wrong predictions are more tolerated and simplicity can be prioritised
* Practical considerations regarding simplicity/complexity
  + Start simple
  + Increase complexity, comparing to previous simpler models (or vice versa)
  + Must document decisions on modelling requirements
  + Final model must meet modelling requirements on performance and simplicity
  + Consider what types of algorithms have been used in similar fields before – they may reflect the nature of the problem more effectively

Specification Gaming

* The model will revert to the most efficient way to solve the problem, which can cause problems
* Greater risk of specification gaming when:
  + Models can’t be explained simply
  + Training data is highly complex
  + Algorithms transform data into multiple dimensions that we can’t understand
  + Objective we’re trying to maximise is over-simplified
    - Eg. Maximise speed with a walking robot that then falls over so that it utilises the speed that is provided by gravity
* Manifests itself as a form of over-fitting
  + The model is fitting to a different pattern than you intended it to!
* Avoiding specification gaming
  + Specify unambiguous goals – clearly aligned with modelling goals
  + Exploration of counterfactual relationships that are possible to learn, using domain knowledge
  + Representative training data
  + Robust data preprocessing
    - Standardising data to make all sample comparable
    - Exploration of latent variables
    - Handling of confounding variables
  + Methods for checking for specification gaming:
    - Fully unseen test data to evaulate generalisability
    - Understandable models / explainable decisions