How to evaluate Algorithms/approaches

Models’ Evaluation
Key factors which will help you to decide which algorithm to use:

• If the relationship between dependent & independent variable is well approximated by a linear model, linear regression will outperform tree based model.

• If there is a high non-linearity & complex relationship between dependent & independent variables, a tree model will outperform a classical regression method.
Key factors which will help you to decide which algorithm to use:

• If you need to build a model which is easy to explain to people, a decision tree model will always do better than a linear model. Decision tree models are even simpler to interpret than linear regression!

• Have you ever tried using linear regression on a categorical dependent variable? Don’t even try! Because you won’t be appreciated for getting extremely low values of adjusted $R^2$ and $F$ statistic.
How to teach machine if it is a ball?

Source: https://www.youtube.com/watch?v=nj_hChhSrOI
Let’s test the machine

it is a ball

Test Passed
Let’s test the machine

it is a ball
Let’s test the machine

This is Underfitting

Test Failed
How to overcome underfitting?

• Find more features
• Try high variance machine learning models
  • Decision Trees
  • K-NN (Different from K-Means)
  • SVM (Another classification and regression algorithm)
How will you teach the machine in this situation

- round shape
- you cannot eat ball
- you can play with ball
your machine recognition is good now

these are all balls

these are not balls
what if you teach machine with too many features from given data?

- round shape
- you cannot eat ball
- you can play with ball
- ball has stitch
- greater than 70mm in diameter
what if you teach machine with too many features from given data?

- round shape
- you cannot eat ball
- you can play with ball
- ball has stitch
- greater than 70mm in diameter
your machine recognition is worse

these are all balls

these are not balls
your machine recognition is worse

these are all balls

these are not balls
your machine had high variances

This is Overfitting
bias and variance

Bias measures how far off in general these models’ predictions are from the correct value.

Variance is how much the predictions for a given point vary between different realizations.

Reference from: http://scott.fortmann-roe.com/docs/BiasVariance.html
bias, variance tradeoff
How to overcome overfitting? (cross validation)

**K-Folds Cross Validation**

In K-Folds Cross Validation we split our data into k different subsets (or folds). We use k-1 subsets to train our data and leave the last subset (or the last fold) as test data. We then average the model against each of the folds and then finalize our model. After that we test it against the test set.

Some other types of Cross Validation

• Holdout Method
  • 50 to 90 % for training and 50 to 10% for testing

• Leave one out CV (K = N)
  • Take each data instance for testing and remaining for training.
  • Computationally expensive

• Bootstrap Methods
  • Bagging

• K-Cross Validation
  • Generally value of K from 5 to 10
Summarizing Overfitting and Underfitting

• How to find if the model is underfitting?
  • If you have high bias.
  • To detect high bias: If high error on both the training and test sets

• How to overcome underfitting?
  • Find more features
  • Try high variance machine learning models
    • Decision Trees
    • K-NN
    • SVM

• How to find overfitting?
  • Test error is much higher than training error

• How to overcome overfitting?
  • K-cross validation

More reading: https://datascience.stackexchange.com/questions/361/when-is-a-model-underfitted
Metrics for Evaluation of Algos.

• **Regression Metrics**
  • Mean Absolute Error.
  • Mean Squared Error (Covered in the Lecture 3)
  • R^2.

• **Classification Metrics**
  • Classification Accuracy.
  • Logarithmic Loss.
  • Area Under ROC Curve.
  • Confusion Matrix.
  • Classification Report (set of more metrics).

[https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d](https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d)
Regression Metrics

• Mean Absolute Error
  • Sum of the absolute differences between predictions and actual values.
  • It gives an idea of how wrong the predictions are.
  • Measure gives an idea of the magnitude of the error, but no idea of the direction (e.g. over or under predicting).

• R^2
  • R^2 (or R Squared) metric provides an indication of the goodness of fit of a set of predictions to the actual values.
  • In statistical literature, this measure is called the coefficient of determination.
  • This is a value between 0 (no-fit) and 1 (perfect fit).
Classification Metrics

- The confusion matrix is a handy presentation of the accuracy of a model with two or more classes.
Classification (Metrics) Report

• **Accuracy:** Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.
  • Accuracy = (TP+TN)/(TP+FP+FN+TN)

• **Precision:** Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.
  • Precision = TP/(TP+FP)

• **Recall:** Recall is the ratio of correctly predicted positive observations to the all observations in actual class.
  • Recall = TP/(TP+FN)

• **F1 Score:** F1 Score is the weighted average of Precision and Recall.
  • F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Metrics of success

Precision = \frac{tp}{tp + fp}

Recall = \frac{tp}{tp + fn}
Classification Metrics (ROC, AUC, .. )

• Area under ROC ('receiver operator characteristics') Curve (or AUC for short) is a performance metric for binary or multi (with some modifications) classification problems

• ROC can be broken down into sensitivity and specificity.
  • Sensitivity: True positive rate also called the recall. It is the number instances from the positive (first) class that actually predicted correctly.
  • Specificity is also called the true negative rate. Is the number of instances from the negative class (second) class that were actually predicted correctly.

• A binary classification problem is really a trade-off between sensitivity and specificity.

Additional sources:
http://www.navan.name/roc/
https://www.youtube.com/watch?v=OAl6eAyP-yo
Summarising: Metrics of success

**sensitivity, recall, hit rate, or true positive rate (TPR)**

\[ TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \]

**specificity or true negative rate (TNR)**

\[ TNR = \frac{TN}{N} = \frac{TN}{TN + FP} \]

**precision or positive predictive value (PPV)**

\[ PPV = \frac{TP}{TP + FP} \]

**negative predictive value (NPV)**

\[ NPV = \frac{TN}{TN + FN} \]

**miss rate or false negative rate (FNR)**

\[ FNR = \frac{FN}{P} = \frac{FN}{FN + TP} = 1 - TPR \]

**fall-out or false positive rate (FPR)**

\[ FPR = \frac{FP}{N} = \frac{FP}{FP + TN} = 1 - TNR \]

**false discovery rate (FDR)**

\[ FDR = \frac{FP}{FP + TP} = 1 - PPV \]

**false omission rate (FOR)**

\[ FOR = \frac{FN}{FN + TN} = 1 - NPV \]

**accuracy (ACC)**

\[ ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN} \]

**F1 score**

is the harmonic mean of precision and sensitivity

\[ F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN} \]
Classifiers: If the training data is correct?

• Accuracy: for evaluating classification models.

• Accuracy = \frac{Correct \ predictions}{Total \ # \ of \ predictions} *

*: Testing data

• Accuracy w.r.t training data:
  • (Corrupt) Data: **Noise** in the data
  • minor(ity) class / Imbalanced data
Noise: Reasons of data corruption

• **Implicit errors:**
  • Introduced by measurement tools
  • Examples:
    • Sensors not working correctly.
    • Restriction on websites (not all users have set the fields public)

• **Random errors:**
  • Experts entering the data
  • Examples:
    • Document digitalization process.
    • Incorrect data entry by an operators.
# Noise Types [2]

- Class Noise
- Attribute Noise

<table>
<thead>
<tr>
<th>Attribute 1 [0-1]</th>
<th>Attribute 2 (Red, Green, Blue)</th>
<th>Class/Labels [+/-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>Red</td>
<td>-</td>
</tr>
<tr>
<td>0.25</td>
<td>Red</td>
<td>+</td>
</tr>
<tr>
<td>0.25</td>
<td>Blue</td>
<td>%</td>
</tr>
<tr>
<td>1.02</td>
<td>Green</td>
<td>+</td>
</tr>
<tr>
<td>0.75</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>0.35</td>
<td>Green</td>
<td>+</td>
</tr>
</tbody>
</table>
## Noise Types

<table>
<thead>
<tr>
<th>Attribute 1 [0-1]</th>
<th>Attribute 2 (Red, Green, Blue)</th>
<th>Class/Labels [+/-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>Red</td>
<td>-</td>
</tr>
<tr>
<td>0.25</td>
<td>Red</td>
<td>+</td>
</tr>
<tr>
<td>0.25</td>
<td>Blue</td>
<td>%</td>
</tr>
<tr>
<td>1.02</td>
<td>Green</td>
<td>+</td>
</tr>
<tr>
<td>0.75</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>0.35</td>
<td>Green</td>
<td>+</td>
</tr>
</tbody>
</table>

- **Class Noise** [2]
  - **Contradictory:** For same set of attributes different labels
  - Mislabeled
Noise Types

- **Class Noise**
  - **Contradictory**
  - **Mislabeled**: Using out of class labels.

<table>
<thead>
<tr>
<th>Attribute 1 [0-1]</th>
<th>Attribute 2 (Red, Green, Blue)</th>
<th>Class/Labels [+/-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>Red</td>
<td>-</td>
</tr>
<tr>
<td>0.25</td>
<td>Red</td>
<td>+</td>
</tr>
<tr>
<td>0.25</td>
<td>Blue</td>
<td>%</td>
</tr>
<tr>
<td>1.02</td>
<td>Green</td>
<td>+</td>
</tr>
<tr>
<td>0.75</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>0.35</td>
<td>Green</td>
<td>+</td>
</tr>
</tbody>
</table>
# Noise Types

- **Attribute Noise**
  - *Erroneous Noise*: Value of attribute is out of range.
- **Missing Values**

<table>
<thead>
<tr>
<th>Attribute 1 [0-1]</th>
<th>Attribute 2 (Red, Green, Blue)</th>
<th>Class/Labels [+/-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>Red</td>
<td>-</td>
</tr>
<tr>
<td>0.25</td>
<td>Red</td>
<td>+</td>
</tr>
<tr>
<td>0.25</td>
<td>Blue</td>
<td>%</td>
</tr>
<tr>
<td><strong>1.02</strong></td>
<td>Green</td>
<td>+</td>
</tr>
<tr>
<td>0.75</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>0.35</td>
<td>Green</td>
<td>+</td>
</tr>
</tbody>
</table>
## Noise Types

<table>
<thead>
<tr>
<th>Attribute 1 [0-1]</th>
<th>Attribute 2 (Red, Green, Blue)</th>
<th>Class/Labels [+/-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>Red</td>
<td>-</td>
</tr>
<tr>
<td>0.25</td>
<td>Red</td>
<td>+</td>
</tr>
<tr>
<td>0.25</td>
<td>Blue</td>
<td>%</td>
</tr>
<tr>
<td>1.02</td>
<td>Green</td>
<td>+</td>
</tr>
<tr>
<td>0.75</td>
<td>[Redacted]</td>
<td>-</td>
</tr>
<tr>
<td>0.35</td>
<td>Green</td>
<td>+</td>
</tr>
</tbody>
</table>

- **Attribute Noise**
- **Erroneous Noise**
- **Missing Values**: Data is missing for an attribute.
Minority Class (or Imbalanced data)

- Consider a binary classification problem
  - 80% instances are labeled with class-1
  - Remaining 20% instances are labeled with Class-2.
- Imbalance class: Often expected

<table>
<thead>
<tr>
<th>Attribute 1 [0-1]</th>
<th>Attribute 2 (Red, Green, Blue)</th>
<th>Class (+/-)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>Red</td>
<td>+</td>
</tr>
<tr>
<td>0.35</td>
<td>Green</td>
<td>+</td>
</tr>
<tr>
<td>0.45</td>
<td>Blue</td>
<td>+</td>
</tr>
<tr>
<td>0.20</td>
<td>Red</td>
<td>+</td>
</tr>
<tr>
<td>0.89</td>
<td>Green</td>
<td>-</td>
</tr>
</tbody>
</table>
Minority classes: Examples datasets

• Fraudulent transactions:
  • Vast majority of the transactions will be in the “Not-Fraud” class
  • Very small minority will be in the “Fraud” class.

• Customer churn:
  • Majority of customers stay with the service (the “No-Churn” class)
  • Small minority cancel their subscription (the “Churn” class).
How to take care of Noise and Imbalanced data

Noise/Imbalanced data: How to handle it?

• Data level
  • Pre-processing of the datasets: remove the noise [Noise problem]
  • Oversampling: Generate synthetic samples [Imbalanced class problem]
    • SMOTE: Synthetic Minority Over-sampling Technique
  • Undersampling: Ignoring the majority class [Imbalanced class problem]
    • Random undersampling: Random removal from majority class instances.
    • Informed undersampling: by finding out the distribution of data first and selectively pick points to be thrown away

• Algorithmic level: Adaptation of the algorithms
Noise/Imbalanced data: How to handle it?

• Data level

• Algorithmic level: Adaptation of the algorithms
  • Tomek Links: Removing instances which are near to borderline [Noise].
  • Cost-sensitive classifier:
    • Additional cost on the model for making misclassification on the minority class during training.
    • Penalties can bias the model towards paying attention to minority class.
    • Example: Modified (proximal) SVM

• Multiple Classifier Systems:
  • Ensemble Learning as an interesting solution to learn from imbalanced data.
  • Example: Random Forest

Some more reading: https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/
References:

• How to determine the quality of a multiclass classifier: https://stats.stackexchange.com/questions/44261/how-to-determine-the-quality-of-a-multiclass-classifier

Demo time!

https://courses.cs.ut.ee/2018/bda/spring/Main/Practice