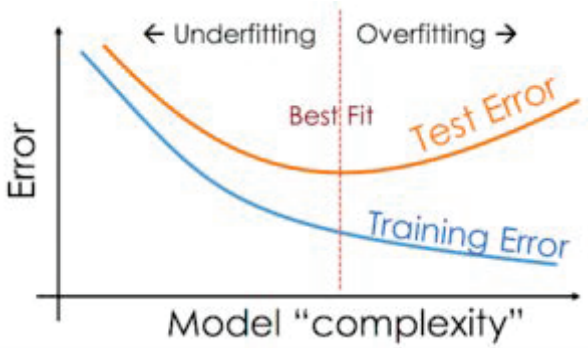
14. EVALUATION OF SUPERVISED ML ALGORITHMS

- We studied the following classification/regression models
  - Perceptron (classification)
  - Linear and Polynomial Regression (regression)
  - Logistic Regression (classification)
  - Artificial Neural Networks (classification and regression)
  - K-Nearest Neighbors (classification and regression)
  - Decision Trees / Random Forests (classification and regression)
  - Support Vector Machines (classification and regression)
  - Naïve Bayes (classification)
- **Three phases in the machine learning process**
  1. training: building the model
  2. validation and selection: evaluating and comparing different models which may represent the same broad type of an algorithm (different K in K-NN) or a completely different algorithm (K-NN and Naïve Bayes)
  3. testing and assessment: final, last-chance estimate of how the machine learning model will perform in the real world



https://vitalflux.com/overfitting-underfitting-concepts-interview-questions/

CLASSIFICATION

There are many ways to evaluate which learning algorithm performs better, including cross-validation, confusion matrix, accuracy and other measures, ROC (Receiver Operating Characteristic) curves, AUC (Area Under the Curve), Precision-Recall (PR) curves, etc.

Confusion Matrix

The confusion matrix is a square matrix whose size is the same as the number of different output labels. In sklearn, the rows correspond to the observed or actual labels and the columns correspond to the labels predicted by the algorithm. In the case of binary classification, the confusion matrix is given by:

	predicted negative	predicted positive
observed negative	true negative (TN)	false positive (FP)
observed positive	false negative (FN)	true positive (TP)

- **Precision:** for each predicted class, what percentage of your predictions are correct?

$$\text{Precision (positive class)} = \frac{\text{TP}}{\text{TP + FP}}$$

$$\text{Precision (negative class)} = \frac{\text{TN}}{\text{TN + FN}}$$

- **Recall:** for each observed class, what percentage of your predictions are correct?

$$\text{Recall (positive class)} = \text{Sensitivity} = \text{True Positive Rate (TPR)} = \frac{\text{TP}}{\text{TP + FN}}$$

$$\text{Recall (negative class)} = \text{Specificity} = \text{True Negative Rate (TNR)} = \frac{\text{TN}}{\text{TN + FP}}$$

- **Accuracy:** what is the percentage of correct predictions?

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- F_1**-score:** is defined, for each class, as the harmonic mean of the recall and the precision

$$F_1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

- F_β**-score** is defined for each class by

$$F_\beta = \frac{(\beta^2 + 1)(\text{Recall} \times \text{Precision})}{\beta^2(\text{Recall}) + \text{Precision}}.$$

Note that if the data is unbalanced, the accuracy measure may not be adequate.

In addition, depending on the problem, sometimes we might want to increase precision, recall, and F_β score. For example, in spam detection we may want to reduce FP so the appropriate measure would be precision (of the positive class), while in detecting cancer in patients we want to reduce FN and we should use recall (of the positive class). When both FP and FN are important, we need to use both precision and recall, and in that situation F_β score is suitable with an appropriate value of β. If FP and FN are equally important, we choose $\beta = 1$ and in that case F_1 score is the harmonic mean. If FP has more impact than FN, we reduce β and choose $\beta \in (0, 1)$, and if FN has more impact than FP, then we increase β and use $\beta \in (1, 10)$, for example.

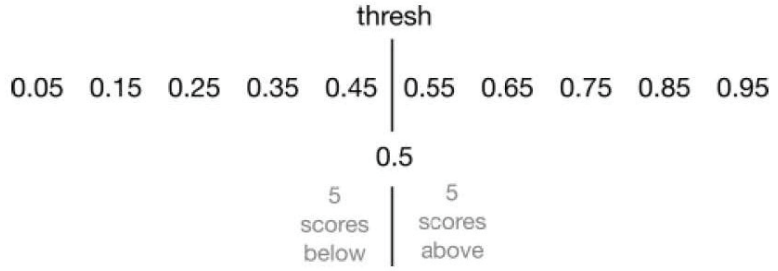
ROC Curve, AUC, and Precision–Recall Curve

These three evaluation methods are used in binary classification.

- **Receiver Operator Characteristic (ROC) Curve**

  Most of the time when we apply a classifier to a data observation, the classifier generates probabilities of each class label for that observation. These probabilities indicate how confidently we can predict a label to that observation after comparing these probabilities with a specified *threshold value*.

  For a given threshold, we create a confusion matrix. If we change the threshold, the predicted labels might change and the confusion matrix might change. For example, if we lower the threshold value, more of our observations will be predicted as positive.



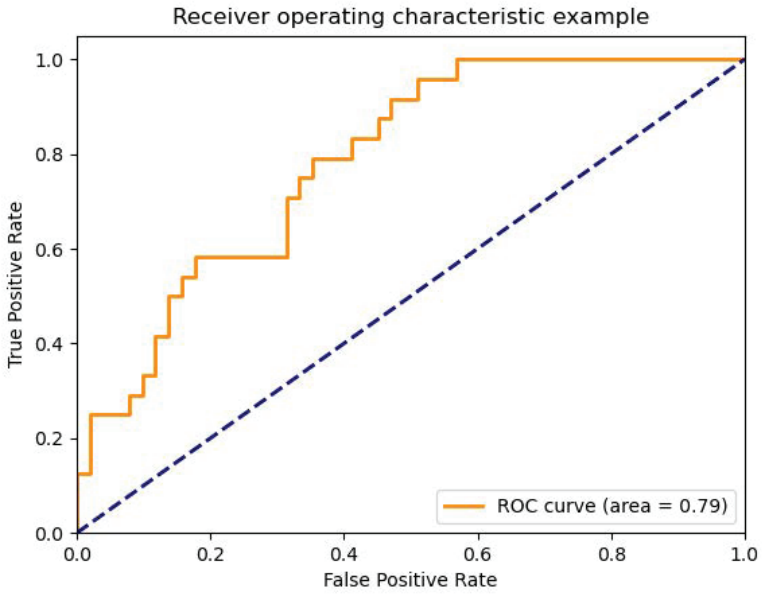
  Instead of comparing confusion matrices for all the different choices of the threshold, we draw the ROC graph that provides a simple way to summarize the model performance. The horizontal axis depicts

$$\text{False Positive Rate (FPR)} = 1 - \text{Specificity} = \frac{FP}{FP + TN}$$

  and the vertical axis depicts

$$\text{True Positive Rate (TPR)} = \text{Sensitivity} = \frac{TP}{TP + FN}.$$

  Note that as move the threshold from right to left, the number of positive predictions is increasing. Thus, both TPR and FPR will increase and we move along the ROC curve in the direction from the bottom left to the top right corner.



https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html

The points on the ROC curve that correspond to optimal threshold values are those where the TPR is the highest and FPR is smallest. We choose the threshold value depending on how many false positives we are willing to accept. A good model should always have ROC curve above the line TPR=FPR (which is what a random prediction would be).

- **Area Under the ROC Curve (AUC)**

  The AUC makes it easy to compare one ROC curve to another (which in turns means comparing one algorithm to another, such as a logistic regression to a random forest).

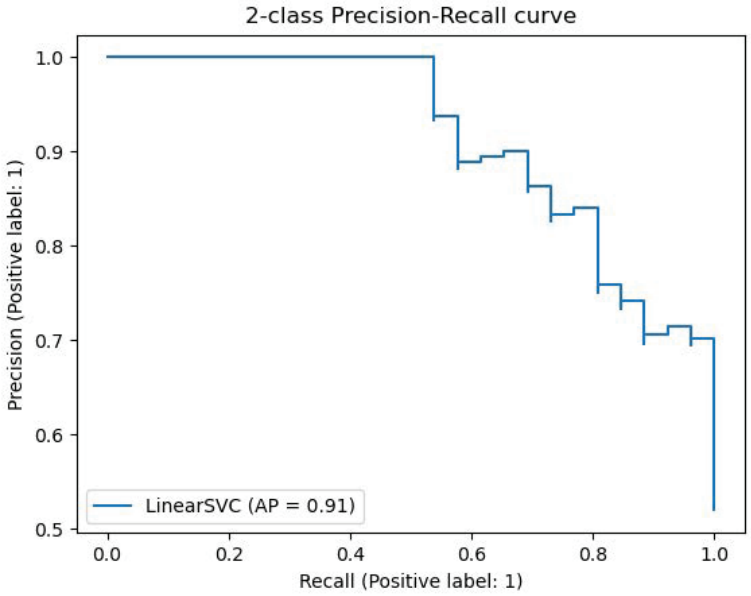
  The AUC is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

  - If AUC = 1, the classifier is able to perfectly distinguish between all the positive and the negative class points correctly.
  - If $0.5 < \text{AUC} < 1$, there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values.
  - If AUC = 0.5, the classifier is not able to distinguish between positive and negative class points meaning either the classifier is predicting random class or constant class for all the data points.
  - If AUC = 0, the classifier would be predicting all negatives as positives and all positives as negatives.

  In summary, ROC curves make it easy to identify the best threshold, in addition to domain expertise, and the AUC help us decide which model is better.

- **Precision–Recall (PR) Curve**

  The PR curve is particularly used in case of imbalanced datasets to evaluate classifier's performance. It depicts precision on the vertical and recall on the horizontal axis. A good classifier will produce a PR curve that is close to the upper right corner.



https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

Python codes:

Lecture_14_ROC_AUC.ipynb

REGRESSION

In regression problems, the target variable is continuous numerical. Given the data set, let y_i denote the actual or observed values and let $\hat{y}_i$ denote the predicted values.

Features $X_1, X_2, \ldots, X_d$	Actual value y_i	Predicted value $\hat{y}_i$	Prediction error $y_i - \hat{y}_i$
...	47	45	2
...	31	31	0
...	35	35	0
...	28	40	-12

The quality of a model is related to how well its predictions match up against actual values. Some of the evaluation metrics for regression problems are the following.

- **Mean Absolute Error (MAE)** is the mean or average of the absolute values of the individual prediction errors and it tells us how big of an error we can expect on average.

$$\boxed{MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|}$$

  For the above example,

$$\begin{aligned} MAE &= \frac{1}{4}\left(|47-45| + |31-31| + |35-35| + |28-40|\right) \\ &= \frac{1}{4}\left(|2| + |0| + |0| + |-12|\right) \\ &= \frac{1}{4}\left(2 + 0 + 0 + 12\right) \\ &= 3.5 \end{aligned}$$

  The main advantage of MAE is that it is in the same unit as the output variable. The disadvantage is that MAE is not a differentiable function so we have to apply various optimizers like Gradient Descent to minimize it.

- **Mean Squared Error (MSE)** is the mean of the squared prediction errors.

$$\boxed{MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

  For the above example,

$$\begin{aligned} MSE &= \frac{1}{4}\left((47-45)^2 + (31-31)^2 + (35-35)^2 + (28-40)^2\right) \\ &= \frac{1}{4}\left((2)^2 + (0)^2 + (0)^2 + (-12)^2\right) \\ &= \frac{1}{4}\left(4 + 0 + 0 + 144\right) \\ &= 37 \end{aligned}$$

Because the MSE contains squares, its units do not match that of the original target. The effect of the square term is most apparent with the presence of outliers in the data: while each residual in MAE contributes proportionally to the total error, the error grows quadratically in MSE. This ultimately means that outliers in our data will contribute to much higher total error in the MSE than they would in the MAE, and the model will be penalized more for making predictions that differ greatly from the corresponding actual values.

- **Root Mean Squared Error (RMSE)** is the square root of the MSE.

$$\boxed{RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}$$

By squaring the errors before we calculate their mean and then taking the square root of the mean, the RMSE gives more weight to the large but infrequent errors. That is why we compare RMSE and MAE to determine whether there are large and infrequent errors. The larger the difference between RMSE and MAE the more inconsistent the error size. For the above example,

$$RMSE = \sqrt{37} = 6.083$$

- **R Squared Score (R^2)** describes the proportion of variation in the target variable that is due to variation in the feature variables.

$$\boxed{R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} = 1 - \frac{SSE}{SST}}$$

If the regression model is "perfect", SSE is zero, and $R^2 = 1$.

For the above example,

$$\begin{aligned} R^2 &= 1 - \frac{(47-45)^2 + (31-31)^2 + (35-35)^2 + (28-40)^2}{(47-35.25)^2 + (31-35.25)^2 + (35-35.25)^2 + (28-35.25)^2} \\ &= 1 - \frac{148}{208.75} \\ &= 0.29 \end{aligned}$$

The most common interpretation of R^2 is how well the regression model fits the observed data. In our example, R^2 of 0.29 reveals that 29% of the data fits the regression model. R^2 value is between 0 to 1 and a larger value indicates a better fit between predictions and actual values.

R^2 is a good measure to determine how well the model fits the dependent variable; however, it does not take into consideration the overfitting problem. If the regression model has many independent variables, because the model is too complicated, it may fit very well the training data but it might perform badly on the test data. The disadvantage of R^2 is that by adding new features, the value of R^2 will increase or remain the same. The problem is when we add an irrelevant feature in the data set and to control this situation we define the **Adjusted R Squared Score**

$$\boxed{R^2_{adjusted} = 1 + (R^2 - 1)\frac{n-1}{n-d+1}}$$

where n is the number of data points in our dataset, d the number of features, and R^2 is the R-squared value determined by the model.

If R^2 does not increase significantly when we add a new feature, then the value of adjusted R^2 will actually decrease. On the other hand, if we see a significant increase in R^2 value when we add a new feature, then the adjusted R^2 value will also increase.

Adjusted R^2 is helpful when the dataset contains a lot of features and we need to choose the most effective features to train the model.

Differences among the regression evaluation metrics

- MSE and RMSE penalize the large prediction errors compared to MAE. However, RMSE is widely used to evaluate the performance of the regression model with other models as it has the same units as the dependent variable.
- MAE is more robust to data with outliers.
- RMSE is a differentiable function that makes it easy to perform mathematical operations in comparison to a non-differentiable function MAE. Therefore, in many models, RMSE is used as a default metric for calculating Loss Function despite being harder to interpret than MAE.
- The lower value of MAE, MSE, and RMSE implies higher accuracy of a regression model.
- R^2 & Adjusted R^2 are used for explaining how well the features in the model explain the variability in the target variable.
- A higher value of R^2 is considered desirable.
- R^2 always increases with the addition of new features which might lead to the addition of the redundant variables in our model. However, the adjusted R^2 solves this problem. Adjusted R^2 takes into account the number of predictor variables and its value decreases if the increase in R^2 by the additional variable isn't significant enough.