UNIT - IV

Model Validation in Classification : Cross Validation - Holdout Method, K-Fold, Stratified K-Fold, Leave-One-Out Cross Validation. Bias-Variance tradeoff, Regularization , Overfitting, Underfitting. **Ensemble**

Methods: Boosting, Bagging, Random Forest.

Supervised Machine Learning: Model Validation, a Step by Step Approach

Model validation is the process of evaluating a trained model on test data set. This provides the generalization ability of a trained model. Here I provide a step by step approach to complete first iteration of model validation in minutes.

The basic recipe for applying a supervised machine learning model are:

Choose a class of model

Choose model hyper parametersFit the model to the training data
Use the model to predict labels for new data

From Python Data Science Handbook by ***Jake VanderPlas***

Jake VanderPlas, gives the process of model validation in four simple and clear steps. There is also a whole process needed before we even get to his first step. Like fetching all the information we need from the data to make a good judgement for choosing a class model. Also providing finishing touches to confirm the results after. I will get into depth about these steps and break it down further.

- Data cleansing and wrangling.
- Split the data into training and test data sets.
- Define the metrics for which model is getting optimized.
- Get quick initial metrics estimate.
- Feature engineering to optimize the metrics. (*Skip this during first pass).*
- Data pre-processing.
- Feature selection.
- Model selection.
- Model validation.
- Interpret the results.
- Get the best model and check it against test data set.

I will be using data set from **UCI Machine Learning Repository**. Data set is from the *Blood Transfusion Service Center* in Hsin-Chu City in Taiwan. This is a classification problem. The idea behind this extends to regression problem as well

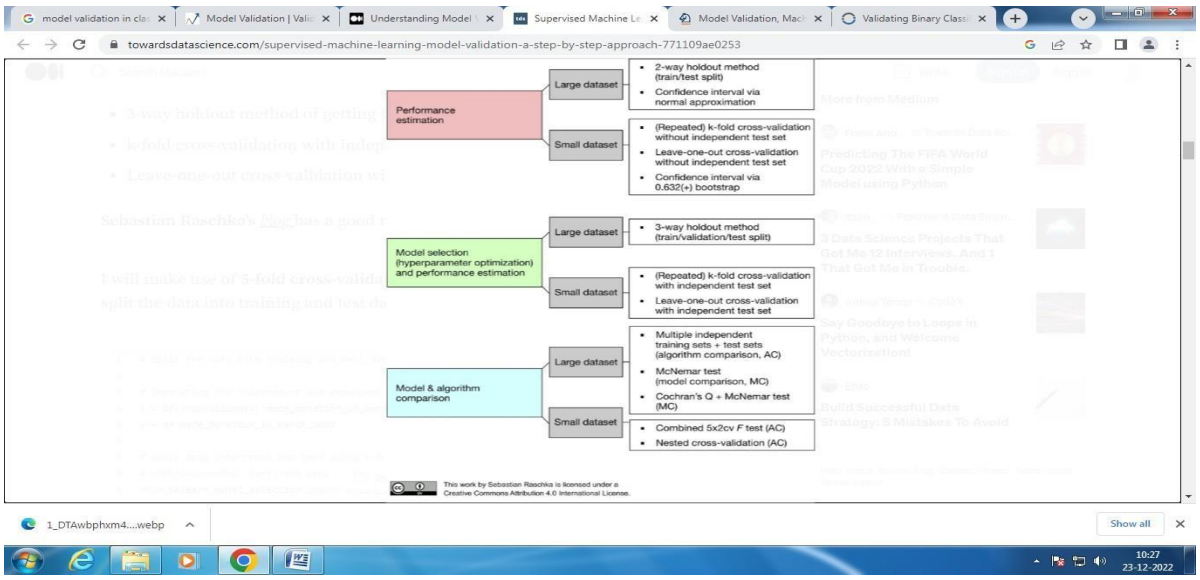
Data cleansing and wrangling.

Blood Transfusion Service Center Data Set is a clean data set. This will not be the case for most other data sets. So this is the step to inspect and clean up the data for example handling missing values…

Split the data into training and test data sets.

There are many ways to get the training and test data sets for model validation like:

- 3-way holdout method of getting training, validation and test data sets.
- k-fold cross-validation with independent test data set.
- Leave-one-out cross-validation with independent test data set.



The main idea behind this step is to get the baseline estimate of metrics which is being optimized. This baseline will work as reference in further steps of model validation. There are several ways to get the baseline estimate for classification problem. I am using the majority class for prediction. The baseline accuracy score is approximately 77%.

```
# Get
quick initial metrics
estimate.
# Using simple pandas value counts method
print(y_train.value_counts(normalize=True))
# Using sklearn accuracy_score
import numpy as np
from sklearn.metrics import accuracy_score
```

```
majority_class = y_train.mode()[0]
prediction = np.full(shape=y_train.shape,
fill_value=majority_class)
accuracy_score(y_train, prediction)
```

Feature engineering to optimize the metrics.

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Feature engineering is fundamental to the application of machine learning, and is both difficult and expensive.

From Wikipedia

This means identifying the relationships between independent and dependent features. This is with the help of graphs like pair plots or correlation matrix. Then the identified relationships we can add as polynomial or interaction features.

Feature engineering step is the point of entry for successive iterations. This is a critical step and plays a greater rolein predictions as compared to model validation.

As a quick solution we can throw in some polynomial features using *PolynomialFeatures in sci-kit learn*.

Domain knowledge on the problem in hand will be of great use for feature engineering. This is a bigger topic in itselfand requires extensive investment of time and resource.

Data pre-processing.

Data pre-processing converts features into format that is more suitable for the estimators. In general, machine learning model prefer standardization of the data set. I will make use of **RobustScaler** for our example.

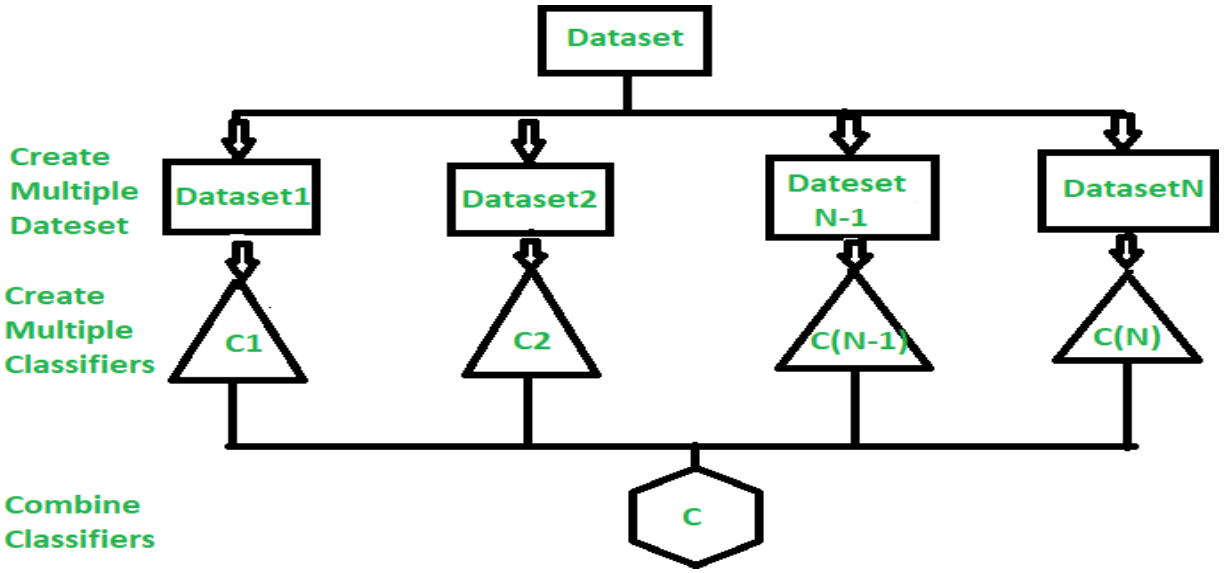
Refer to sci-kit learn's *Preprocessing data* section for detailed information.

Feature selection.

Feature selection or dimensionality reduction on data sets helps to

- Either to improve models' accuracy scoresor
- To boost their performance on very high-dimensional data sets.

I will use **SelectKBest**, univariate feature selection method. The scoring function used for classification andregression problems will vary.



Why do ensembles work?

Dietterich(2002) showed that ensembles overcome three problems –

- **Statistical Problem –**

The Statistical Problem arises when the hypothesis space is too large for the amount of available data. Hence, there are many hypotheses with the same accuracy on the data and the learning algorithm chooses only one of them! There is a risk that the accuracy of the chosen hypothesis is low on unseen data!

- **Computational Problem –**

The Computational Problem arises when the learning algorithm cannot guarantees finding the best hypothesis.

- **Representational Problem –**

The Representational Problem arises when the hypothesis space does not contain any good approximation of thetarget class(es).

Main Challenge for Developing Ensemble Models?

The main challenge is not to obtain highly accurate base models, but rather to obtain base models which make different kinds of errors. For example, if ensembles are used for classification, high accuracies can be accomplished if different base models misclassify different training examples, even if the base classifier accuracy is low.

Methods for Independently Constructing Ensembles –

- *Majority Vote*
- *Bagging and Random Forest*

- *Randomness Injection*
- *Feature-Selection Ensembles*
- *Error-Correcting Output Coding*

Methods for Coordinated Construction of Ensembles

UNIT – V

Unsupervised Learning : Clustering-K-means, K-Modes, K-Prototypes, Gaussian