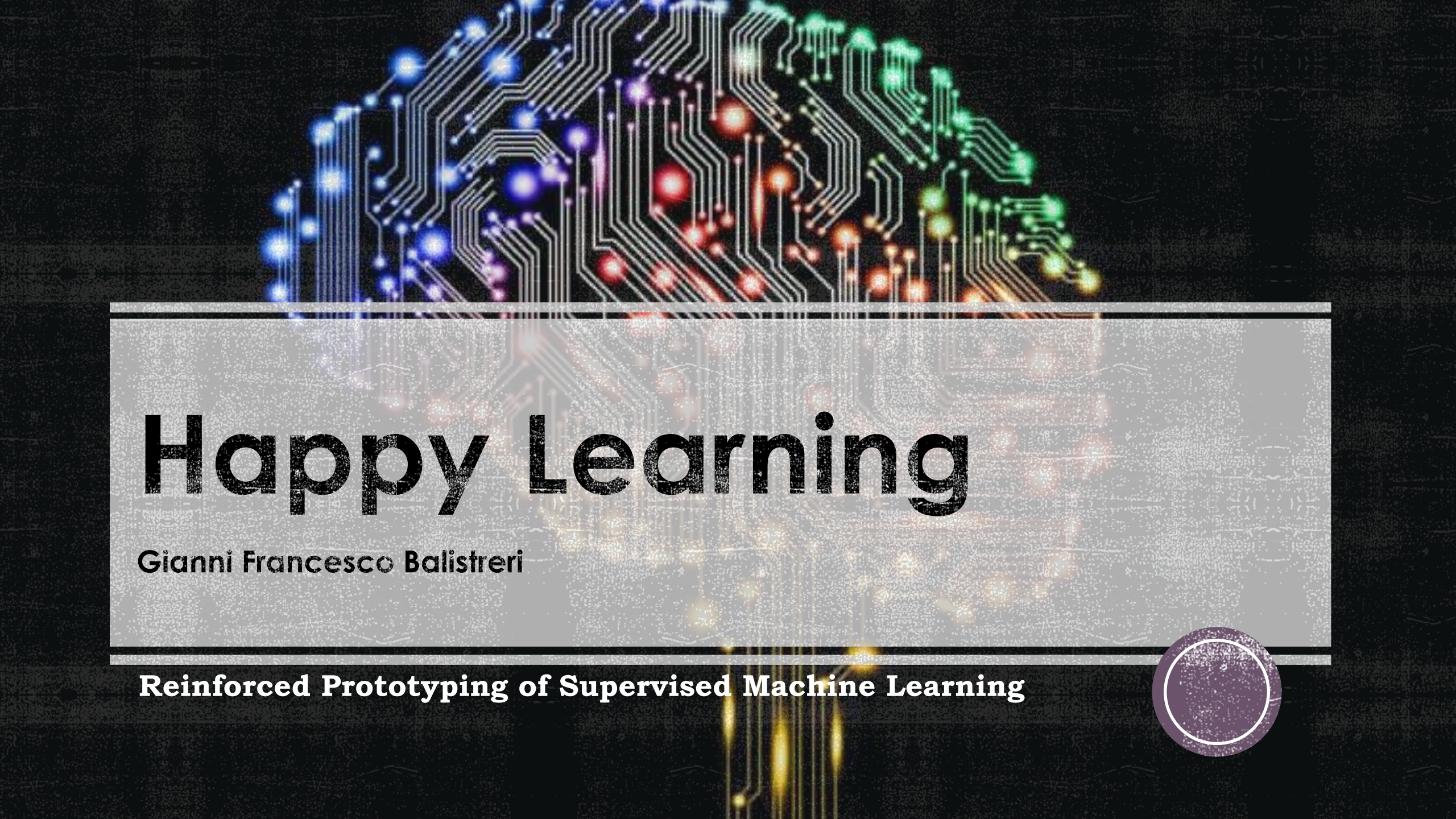


Happy Learning

Gianni Francesco Balistreri

Reinforced Prototyping of Supervised Machine Learning

Content

- Purpose
- Fully Autonomous Framework
- FeatureEngineer Module:
  - Concept
  - Analytical interpretation
  - Analytical based data processing
- FeatureLearning Module: Concept
- FeatureSelector Module: Concept
- FeatureTournament Module:
  - Concept
  - Framework
- Genetic Module:
  - Concept
  - Reinforcement Learning Architecture
  - Optimizing Feature Engineering
  - Optimizing Machine Learning Models
  - Supervised Machine Learning Metric Score (SML): Concept
  - Supervised Machine Learning Metric Score (SML): Calculation
- DataMiner Module: Concept

Purpose

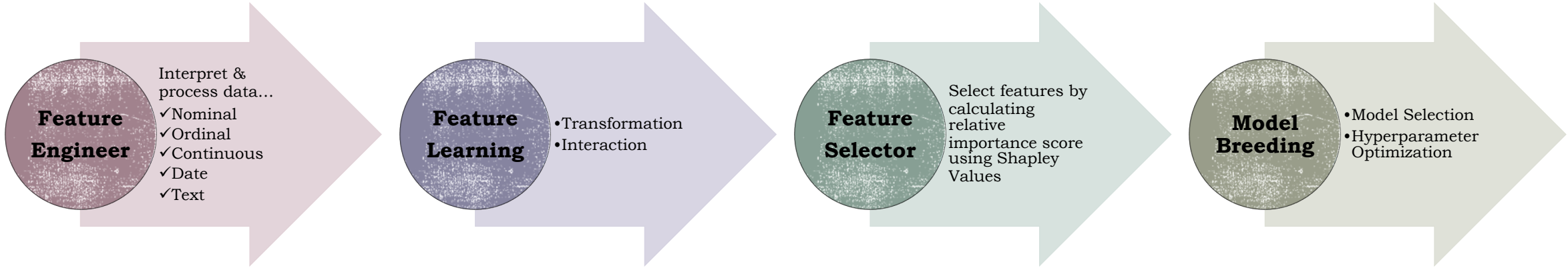
- Efficient prototype developing of supervised machine learning models in two different ways using structured data set …

(1) **Fully autonomous:**

- Interpret feature analytically
- Prepare data set using common processing methods
- Generate new features using feature engineering methods, based on the measurement level of each feature (categorical, ordinal, continuous, date, text)
- Learn reinforced which features should be engineered using evolutional theorie concept called Genetic Algorithm
- Evaluate relative feature importance by calculating Shapley scores using a tournament framework
- Breed or optimize proper supervised machine learning model using Genetic Algorithm

(2) **Partially autonomous** using one or more of the above processing steps

Fully Autonomous Framework: Overview

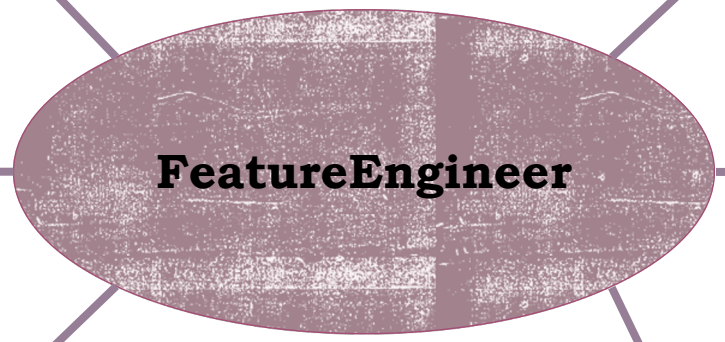


You can run this framework using the **DataMiner** module !

FeatureEngineer Module:

One ‘pickable‘ Object

Handling invalid values properly:
- Missing data (Multiple Imputation)
- Infinit (Replace by „next valid“ value)



Analytical interpretation

Common Processing Methods:
- Import data set from file or db
- Subset data
- Merge data sets
- …

Analytical based processing:
- Categorical (nominal / ordinal)
- Continuous
- Date
- Text

Process tracking:
- Re-Generate features for predictions

FeatureEngineer: Concept

- The concept of the FeatureEngineer module is to process data smartly
  - This means …
    - ✓ … to process features accordingly to their measurement level (nominal, ordinal, date, etc.)
      - There is an internal decorator to ensure that each feature is properly processed
    - ✓ … to pre-process data set by common methods like importing, merging, subsetting, etc.
    - ✓ … to track every processing action
    - ✓ … to capture internal results like transformation objects, name or value mapping
    - ✓ … to re-generate features for predictions in the same way as they was processed in training
    - ✓ … the hole object can be saved as pickle file
    - ✓ … it is equiped with several misc methods like „notepad“ in which you can write & read comments about the data set for example
    - ✓ … it can be used in development and production environment both
    - ✓ … it can be used in an reinforcement learning (especially GA) framework. Therefore it has an internal actor-critic framework implemented

Analytical Interpretation

- Feature segmentation based on the data:
  (1) Categorical features (nominal scaled)
  (2) Semi-Continuous (ordinal scaled)
  (3) Continuous
  (4) Date
  (5) Text or ID
    i. Phrases (text containing natural language)
    ii. Enumeration (text containing enumerated categories only)
    iii. URL / Email (text containing url or email address only)
    iv. ID

- Analytical interpretation is required for intelligent and efficient reinforced feature engineering of mixed data sets

Analytical based data processing

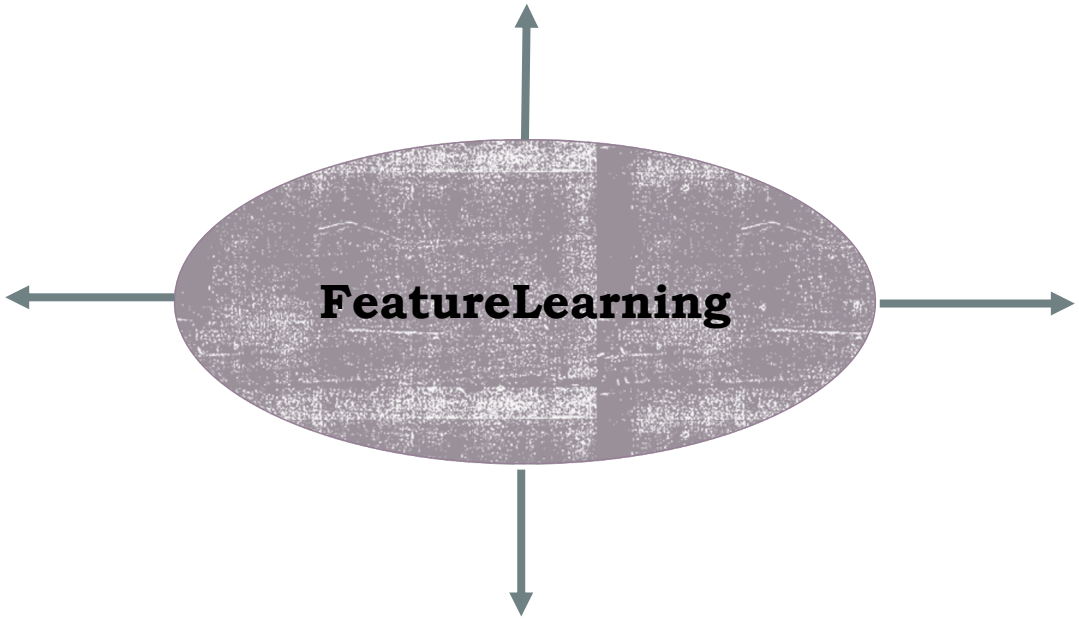
- For developing supervised machine learning models you have to process data by using proper methods
- Methods to process continuous & semi-continuous data:
  - ✓ Transformation of a single feature (Normalizing, Standardizing, Scaling, …)
  - ✓ Interaction of more than one feature
  - ✓ Binning (discretize continuous data into bins supervised and unsupervised)
- Methods to process categorical (nominal) data:
  - ✓ One-Hot-Encoding
  - ✓ „One-Hot-Merging“ (merge one-hot-encoded features)
- Since machine learning algorithms can handle numeric data only you have to process non-numeric raw data in a certain way:
  - ➢ Methods to process datetime data:
    - ✓ Interaction (time difference between dates)
    - ✓ Discretizing (extract parts of dates like year, month, week, etc.)
  - ➢ Methods to process text data:
    - ✓ Counting (length, words, sentences, special characters, special occurances, etc.)
    - ✓ Semantic (NER, POS, Tree Dependencies, Noun Chunks, Emojis)
    - ✓ Enumeration / URL / Email detection (check whether text content can be interpreted as categories)
    - ✓ Similarity (TF-IDF, Clustering)

FeatureLearning Module

Model development on the fly using Genetic Algorithm

Engineer Categorical Features

- Interaction (One-Hot-Merging)



Feature Engineering take place only for the fittest data set (features)

Engineer (Semi-) Continuous Features

- Transformation
- Interaction

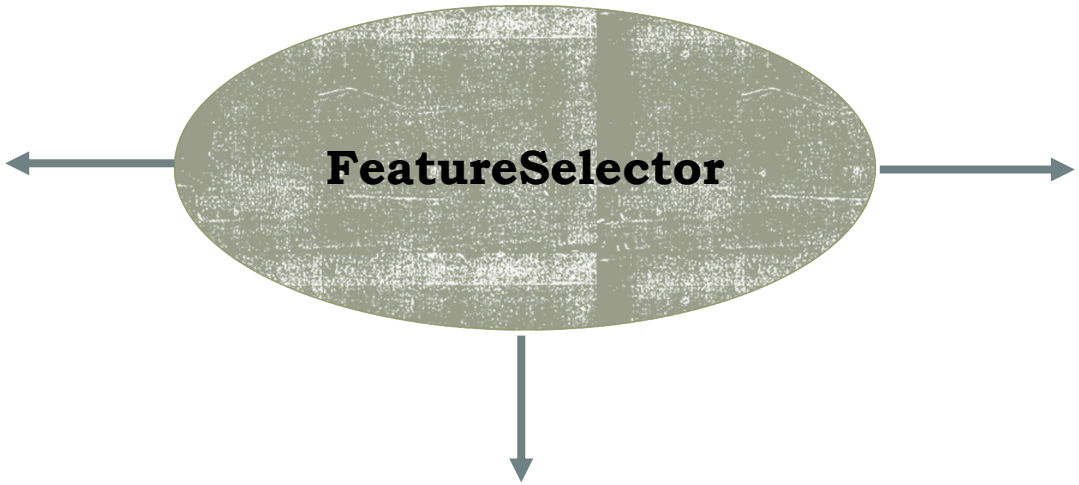
FeatureLearning: Concept

- The concept of the FeatureLearning module to efficiently generate „deep“ features using the hole action space in respect to the measurement level of the data
  - This means …
    - ✓ … the learning process depends on the analytical interpretation of the data:
      - continuous and semi-continuous data are processed in the same framework:
        - Model config: XXX
      - the one-hot-encoded (categorical) features are processed in an other framework using „One-Hot-Merging“
        - Model config: XXX
    - ✓ … to generate „deep“ features efficiently, it uses the Genetic Algorithm framework to optimize feature engineering by using only the fittest features for mutation
    - ✓ … the underlying machine learning model, used in GA, is generated reinforced (on the fly) by using the GA framework for optimizing model and model parameters

FeatureSelector Module

Relative Importance of Features using Shapley Values

- FeatureTournament: Calculating Shapley Values using tournament framework



Select top-n features automatically based on their relative feature importance

Other feature selection methods:

- Decision Tree (RF, GBDT, XGB)
- Lasso Regression

FeatureSelector: Concept

- The concept of the FeatureSelector module is to efficiently run different kind of feature selection frameworks and automatically extract top-n predictors

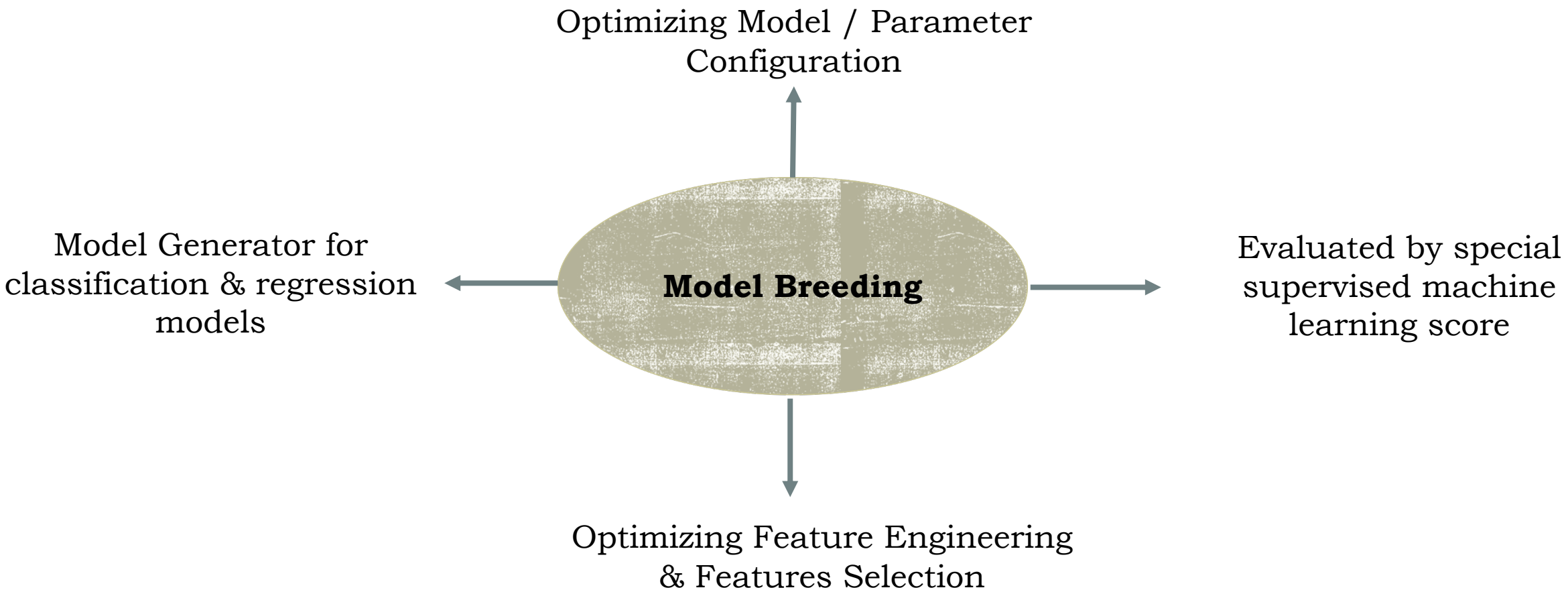
- Feature Selection Framework:
  (1) Shapley Value Approximation (Feature Tournament)
  (2) Decision Tree Algorithm (Random Forest, Gradient Boosting Decision Tree)
  (3) Lasso Regression (for continuous data only)

FeatureTournament: Concept

- The concept of the FeatureTournament module is to efficiently calculate relative importance score for each feature in data set, in respect to the specified target feature using an approximation of the Shapley Value concept
- Applied Shapley Value Approximation (Shapley Additive Explanation):
  - based on supervised machine learning model that have a feature importance measurement integrated:
    - ✓ Decision Tree (Random Forest / Gradient Boosting Decision Tree)
    - ✓ Lasso Regression
  - ✓ framework is designed as a tournament
  - the algorithm works as follows:
    (1) **Penalty**: ensures that predictors without any information for clarification of the target feature are excluded from the tournament
    (2) **Tournament**: calculate scoring by evaluating both fitness of model and fitness of features regarding feature importance measurement, inherited by the ml algorithm itself

FeatureTournament: Framework

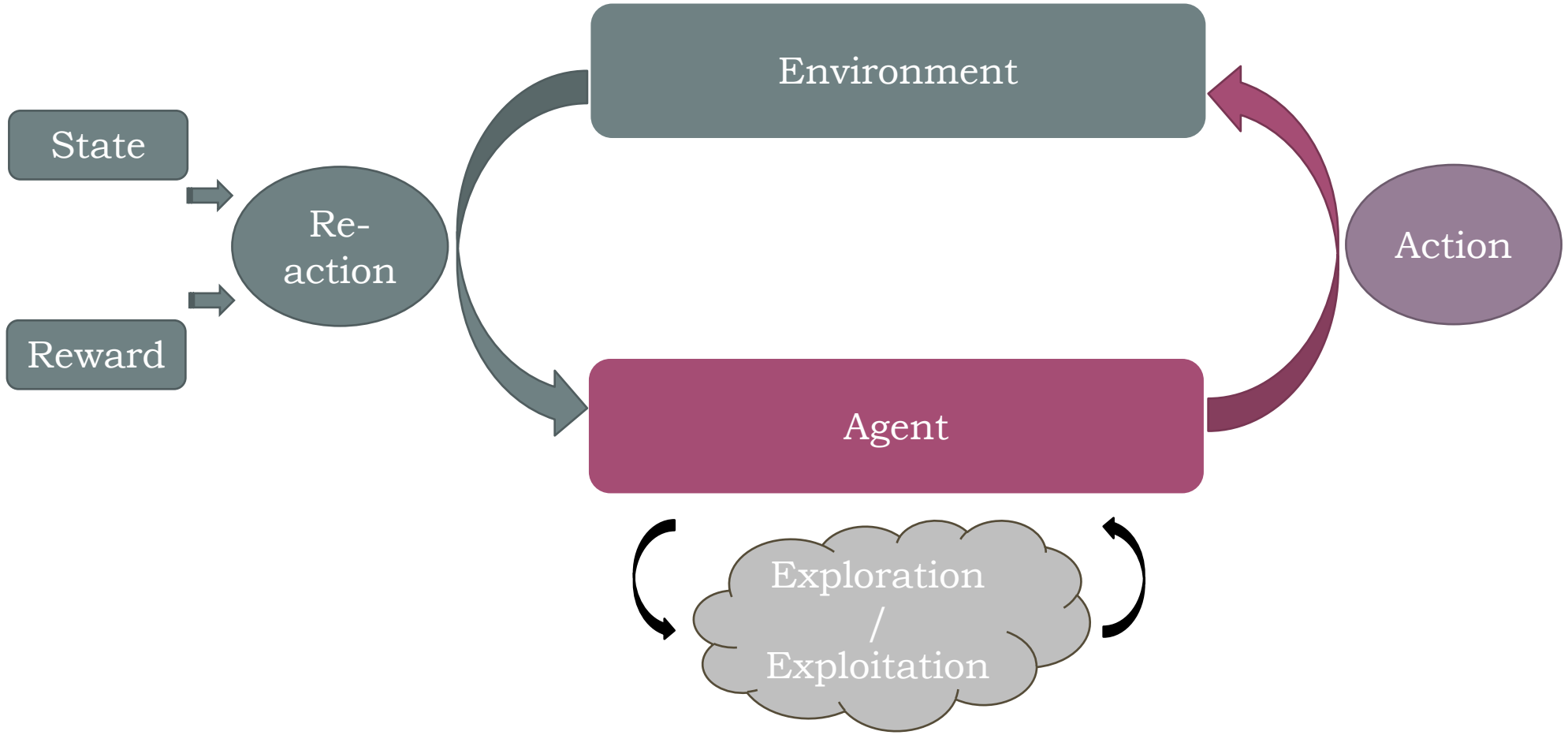
GeneticAlgorithm Module



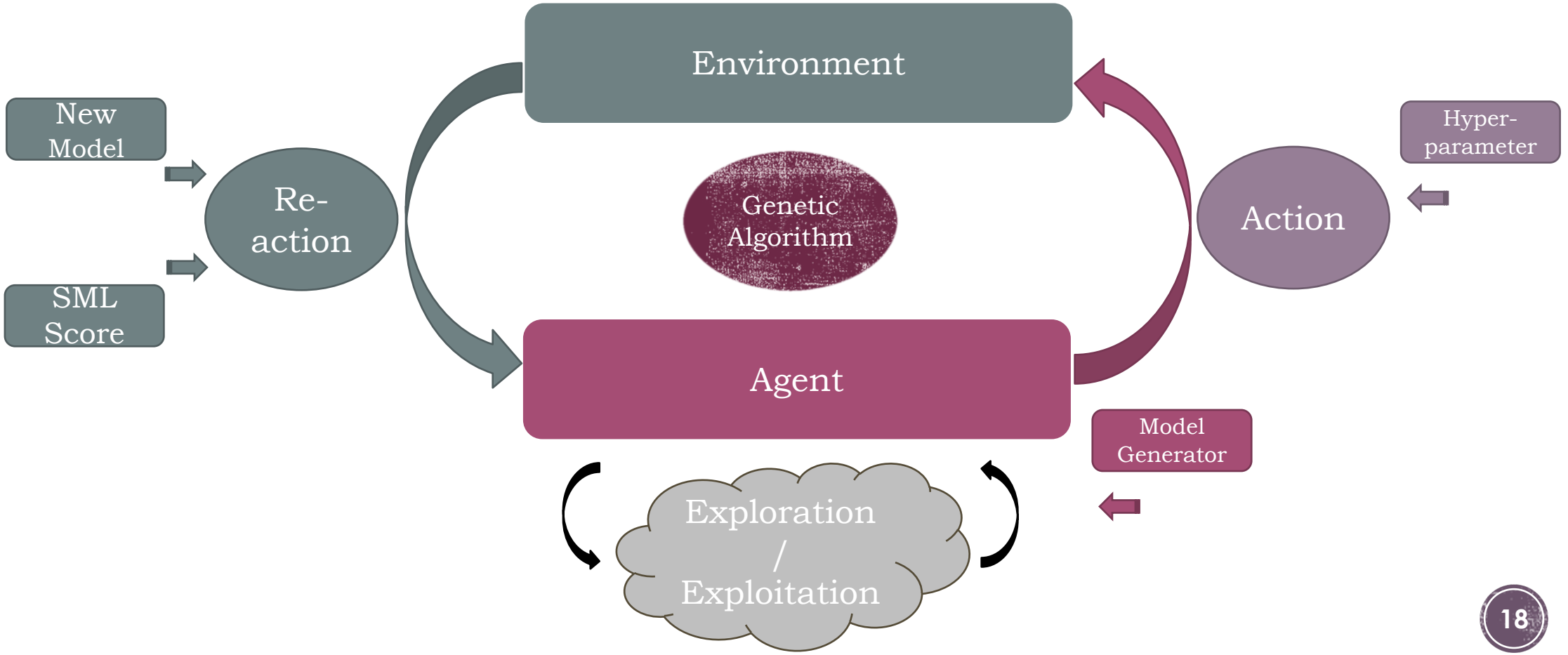
Genetic: Concept

- The concept of the Genetic module is to evolve properly trained supervised machine learning models for …
  (1) Prediction
  (2) Feature Selection
  (3) Feature Engineering
  (4) Data Sampling

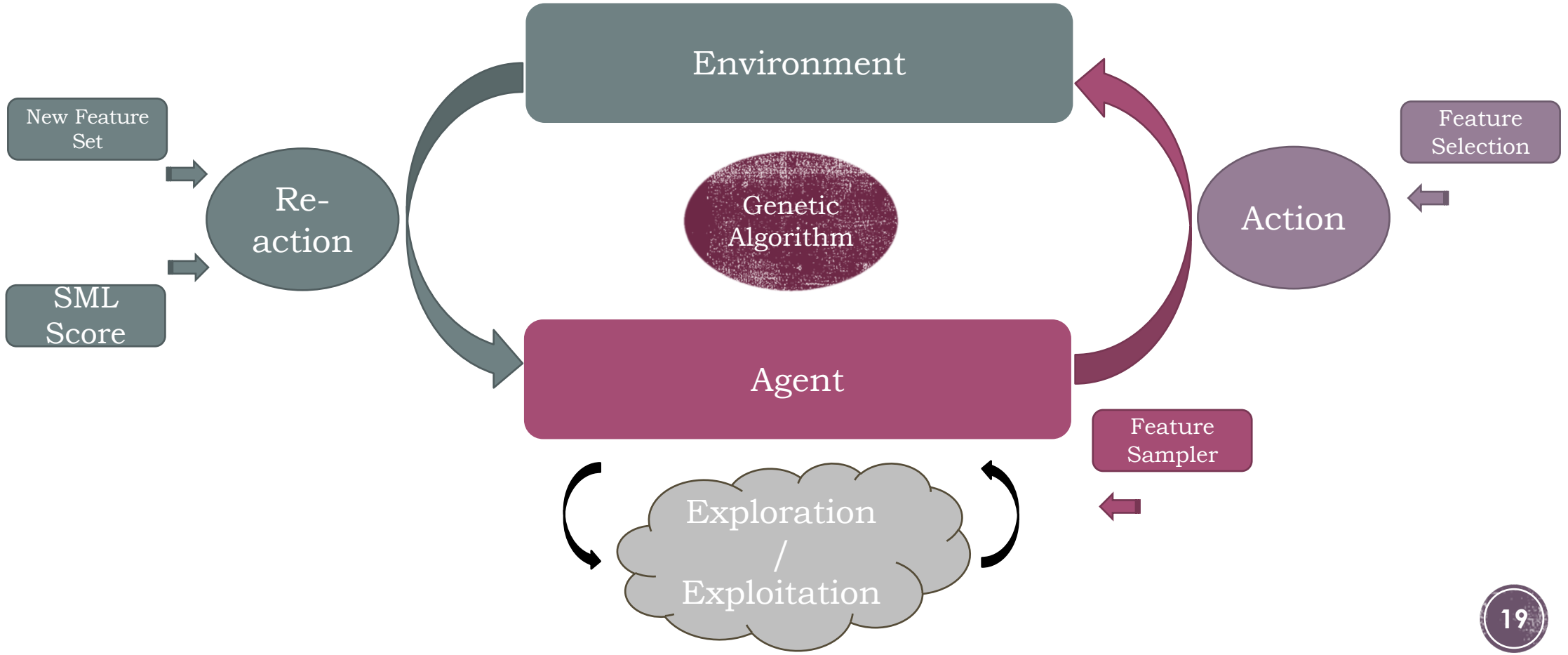
Reinforcement Learning:



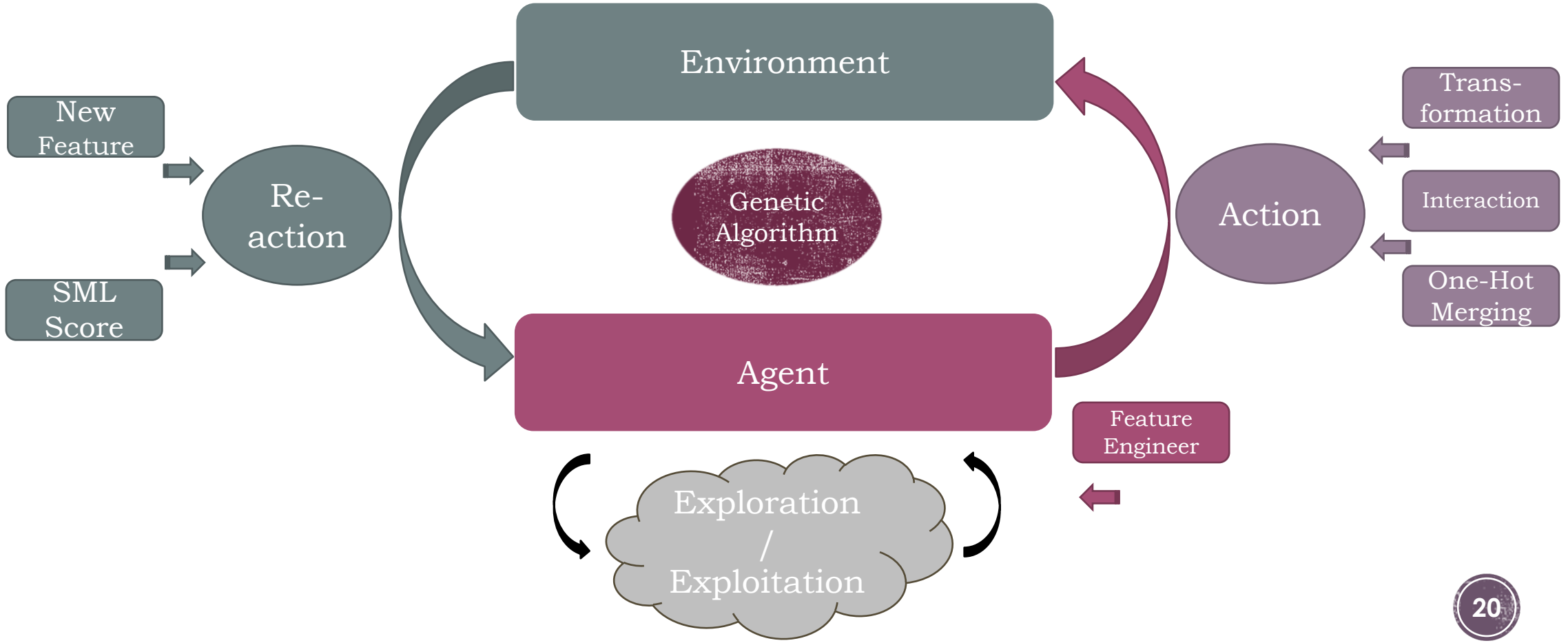
Genetic Algorithm: Model Breeding



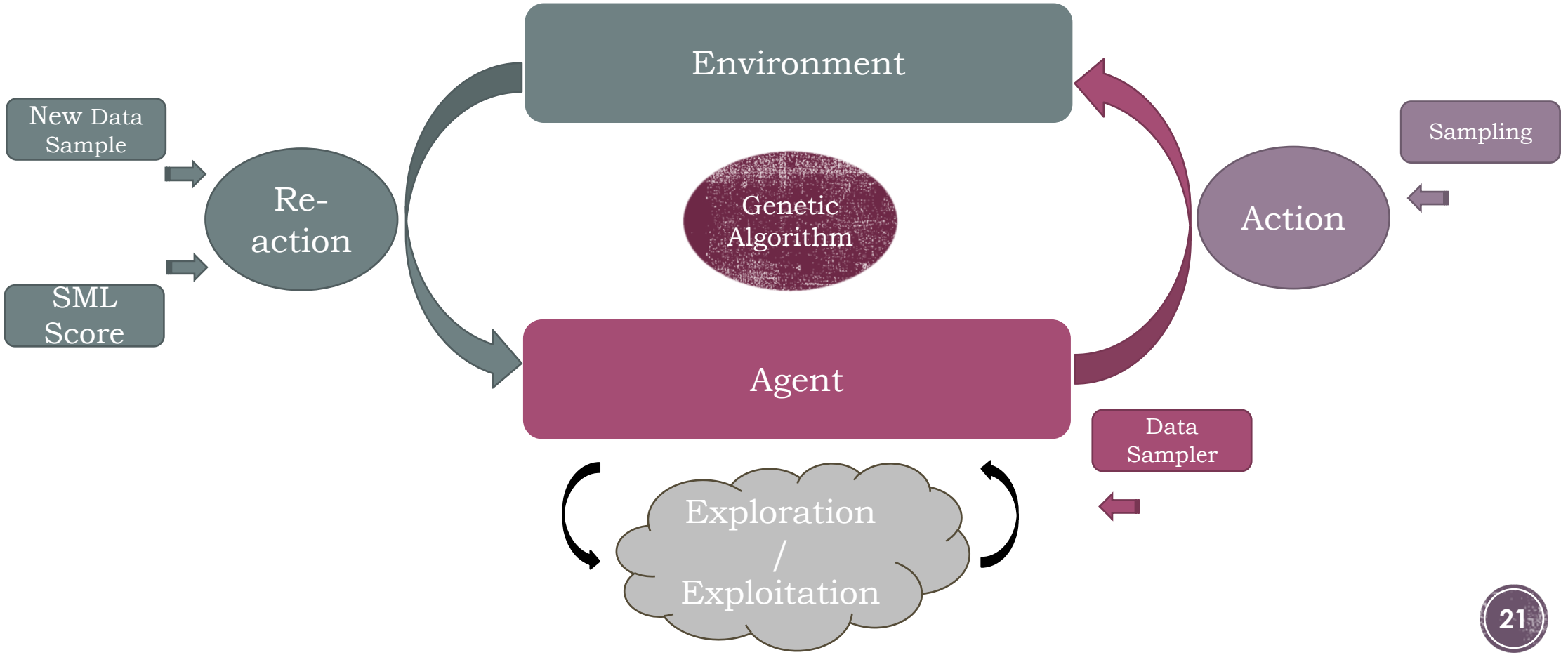
Genetic Algorithm: Feature Selection



Genetic Algorithm: Feature Engineering



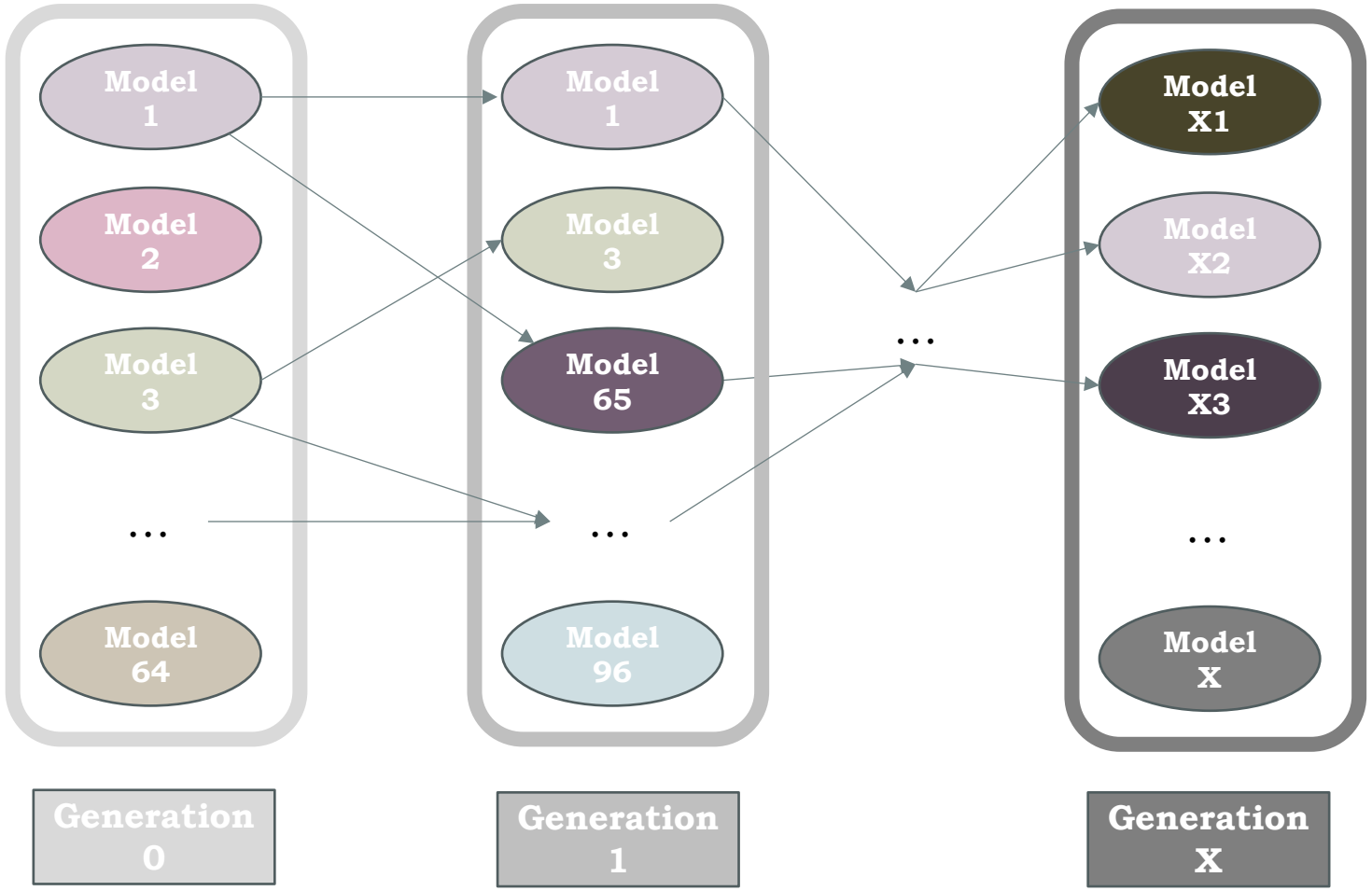
Genetic Algorithm: Data Sampling



Genetic Algorithm (GA)

- The algorithm works as follows …
  (1) Populate generation 0 by a fixed number of randomly drawn individuals
  (2) Train and evaluate machine learning models
  (3) Determine the fittest individuals of the population based on their fitness score
  (4) Mutate genes of the weaker individuals based on the gene pool of the fittest
  (5) Select the fittest individual after breeding generation X

Evolution



- Genes (hyperparameter, feature, etc.) of the fittest individuals (best half of the population) are inherited from generation to generation
- Mutation takes place on a random sample of the gene pool
  - In some cases the mutation process changes the hole gene pool

ModelGenerator

- The Genetic Algorithm inherently uses supervised learning algorithm either to optimize feature engineering or to optimize hyperparameters

Transfer Learning

- The Genetic Algorithm inherently uses supervised learning algorithm either to optimize feature engineering or to optimize hyperparameters

Supervised Machine Learning Metric (SML): Concept

- This special metric is a simple integration of several evaluation aspects of supervised machine learning models to calculate one score:
  - The metric covers following aspects:
    (1) Classification or Regression metric
      i. Binary Classification: AUC
      ii. Multi-Class Classification: Cohen‘s Cappa
      iii. Regression: Normalized Root-Mean-Square Error (RMSE / Standard Deviation)
    (2) Difference between train and test error
    (3) Training time in seconds
  - SML-Scoring is compareable within a Genetic Algorithm framework only !

Supervised Machine Learning Metric (SML): Formula

- SML-Score is calculated as follows:
  (1) Calculate test metric penalty score:

$$ML\ Test\ Metric\ Weight\ *\ (\frac{Start\ Value}{|1-ML\ Test\ Metric|}\ -Start\ Value)$$

  (2) Calculate difference between train metric and test metric:

$$ML\ Test\ Metric\ Weight\ *\ (\frac{Start\ Value}{1\ -\ |ML\ Train\ Metric-ML\ Test\ Metric|}\ -Start\ Value)$$

  (3) Calculate training time penatly score:

$$Training\ Time\ in\ secondes * (Training\ Time\ Weights\ * Start\ Value)$$

  - SML = Start Value – Test Metric Penalty Score – Train Test Metric Penalty Score – Training Time Penalty Score
- Scaling of the metric: see the happy_learnings.html / .ipynb

Resume

- Important and interesting insights of Happy ;) Learning are …

  - ✓ Genetic Algorithm is a powerful framework for reinforcement learning
    - ❖ Model Breeding | Feature Engineering | Feature Selection | Data Sampling
  - ✓ Genetic Algorithm can be used for transfer learning
  - ✓ Structured (tabular) data sets can be properly interpreted analytically
  - ✓ Feature engineering of structured (tabular) data sets can be fully automated (FeatureLearning)
  - ✓ Additive Shapley Explaination (FeatureTournament) is a powerful and stable framework for measuring feature importances of structured (tabular) data sets
  - ✓ Development of supervised machine learning models using structured (tabular) data can be fully automated (DataMiner)

Happy Learning ;)