Machine Learning Apps. Feature engineering

regressionw -

train_df, test_df

- LinearRegression
- linreg.fit(X_train, y_train)
- Predict
  - y_pred_linreg = linreg.predict(X_test)

Model Selection tools/Imports

- train_test_split(data, test_size, rs)

Regression

- mean_squared_error
- mean_absolute_error (y_test, y_pred)
- r2_score
- Weights_
- Intercept_

Feature Improvement + data cleaning

Scaling:

Linear Regression : Insensitive

KNN : Sensitive

Regularization LR : Sensitive

Trees : Insensitive

Usefulness of data normalization and other data trans.

The usual preprocessing in machine learning: Standardize each variable {mean 0, variance 1} StandardScaler

- MinmaxScaler: Scale to a given range
- Robust Scaler: Center data to median for outliers.
- MaxabsScaler: Max absolute value - Sparse data

Discretization:

- Uniform (non-linear): Seperate most frequent and Concentrate the less frequent. Reduce impact of outliers.
- Gaussian: Concentrate around mean.

* Expand several orders but most data is around first order → apply log

One hot Encoding

Aims to use categorical values in a model that only accepts continuous variables.

* has Colinearity problems

Data Imputation

Discard entire rows or columns

Imputation: mean, median, mode } Univariate

Linear Regression Imputation:

If we know a linear relationship then we can build a linear imputer Multivariate Imputation

Knn Imputation:

Discrete → most_frequent	train data	Discrete → Hamming
Continuous → mean		Continuous → Euclidean

Outlier detection Models

Gaussian Mixture Models for data modeling and Clustering

Feature Extraction:

$X = \begin{bmatrix} x_1 \\ \vdots \\ x_D \end{bmatrix}$ $X' = \begin{bmatrix} x_1 \\ \vdots \\ x_K \end{bmatrix}$ where $X' = U^T X^T$ where U is the transformation matrix

PCA

Obtain a reduced set of $K < D$ orthogonal variables → principal components.

principal components are linear combinations of the input vars. but sorted in order of decreasing variance.

Each column of U is a principal component

PCA recursive algorithm:

- Find first pc, U_1, that maximizes variance of the mapped data: $U_1^T X$ is of max variance.
- Deflate U_1 from data $X^2 = X - U_1 X U_1$
- find U_2 s.t. $U_2^T X^2$ is max var
- Repeat deflation.

2.1 Mathematical formulation

$$\mathbf{U} = \underset{\mathbf{U}}{\operatorname{argmax}}\ \operatorname{Tr}\left\{\mathbf{U}^T\mathbf{X}^T\mathbf{X}\mathbf{U}\right\} = \underset{\mathbf{U}}{\operatorname{argmax}}\ \operatorname{Tr}\left\{\mathbf{U}^T\mathbf{C}_{\mathbf{XX}}\mathbf{U}\right\}$$

$$\text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}$$

Solution:

Let's start considering the projection onto a one-dimensional space ($K = 1$). So, we want to solve the following optimization problem:

$$\mathbf{u}_1 = \underset{\mathbf{u}_i}{\operatorname{argmax}}\ \mathbf{u}_1^\top \mathbf{C}_{\mathbf{XX}}\mathbf{u}_1$$

$$\text{s.t. } \mathbf{u}_1^\top\mathbf{u}_1 = 1$$

If we apply Lagrange multipliers, we can include the constraint into functional by means of the lagrange multiplier λ_1 and we arrive to an unconstraint problem

$$\mathbf{u}_1 = \underset{\mathbf{u}_i}{\operatorname{argmax}}\ \mathbf{u}_1^\top \mathbf{C}_{\mathbf{XX}}\mathbf{u}_1 + \lambda_1\left(1 - \mathbf{u}_1^\top\mathbf{u}_1\right)$$

by making its derivate equal to zero, we obtain that the optimum solution has to satisfiy

$$\mathbf{C}_{\mathbf{XX}}\mathbf{u}_1 = \lambda_1\mathbf{u}_1$$

which indicates that $\mathbf{u}_1$ must be an eigenvector of the covariance matrix $\mathbf{C}_{\mathbf{XX}}$ and λ_1 is its associated eigenvalue.

Besides, if we left-multiply this expression by $\mathbf{u}_1^\top$ and make use of $\mathbf{u}_1^\top\mathbf{u}_1 = 1$, we obtain that

$$\mathbf{u}_1^\top\mathbf{C}_{\mathbf{XX}}\mathbf{u}_1 = \lambda_1$$

that is, the variance of the projected data by the eigenventor $\mathbf{u}_1$ is equal to its associated eigenvalue λ_1. So, we can get the maximum projected variance, with a single projection, if we set this first principal component $\mathbf{u}_1$ as the eigenvector with the largest eigenvalue λ_1. Later, we can define additional principal components in an incremental fashion by choosing each new direction as the one eigenvector with the next highest eigenvalue so that the projected variance is maximized.

So, we can obtain the first K projections of the PCA algorithm by solving the following eigenvalue problem

$$\mathbf{C}_{\mathbf{XX}}\mathbf{u} = \lambda\mathbf{u}$$

So, the projection matrix, $\mathbf{U}$, consists of the first eigenvectors of $\mathbf{C}_{\mathbf{xx}}$ (i.e., those associated with largest eigenvalues)

$$\mathbf{U} = \operatorname{eigs}(\mathbf{C}_{\mathbf{XX}})$$

Partial least squares

Find projections of input and output data with maximum covariance.

$U, V = \operatorname{argmax} \operatorname{Tr}\{U^T X^T Y V\} = \operatorname{argmax} \operatorname{Tr}\{U^T C_{xy} V\}$

Max # of projections = output classes - 1

Canonical Correlation Analysis

Find directions of max correlation between input and output data.

$u, v = \operatorname{argmax} \frac{(U^T C_{xy} V)^2}{U^T C_{xx} U \cdot C_{yy} V}$ and $U, V = \operatorname{argmax} \operatorname{Tr}\{U^T C_{xy} V\}$

Output classes - 1.

LDA Linear discriminant analysis

Classification method that projects each piece of data onto the real line by a linear operation $z = w^T x$

Feature Selection

Filtering methods (correlation, mutual info...)
wrappers (classification/regression performance over CV to find features)
Embedded (Integrated into model) e.g. L1 regressor

Filtering

- Independent from learning stage
  - Variance (unsupervised)
  - Correlation coefficient (regression)
  - Statistical tests (binary)
  - Mutual Information

Do we think correlation is a good reference criteria?

Mutual Info:

Mutual Information is capable of capturing any type of relationship between two random variables, unlike correlation.

$$MI(X,Y) = \iint p_{x,y}(x,y) \log \frac{p_{x,y}(x,y)}{p_x(x)\, p_y(y)} = [0, \infty] \text{ but } MI > 2 \text{ is rare.}$$

MI between two quantities is a measure of the degree to which knowledge of one quantity reduces the uncertainty about the other. * We don't know these distributions

→ Most use MI estimators based on histograms. But SKlearn's MI implementation uses K-NN based Estimation.

Summary

- Easy to use/interpret
- Computationaly Efficient
- able to detect any rel.
- May be important when interacting with another variable but not on it's own. (MI cannot detect)
- usefulness depends on model being used

Multivariate Criteria

- Selecting subsets rather than individual criteria.

Minimum Redundancy Maximal Relevance (MRMR)

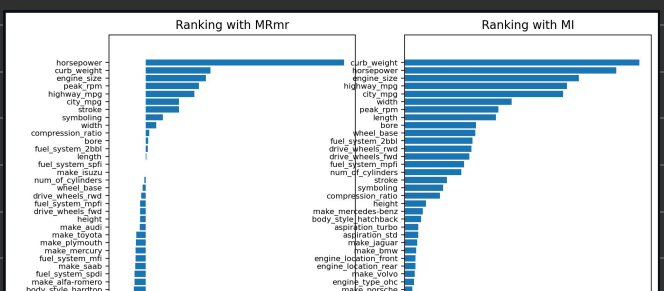
- Based on forward search, at each step we add feature.

$$\text{Relevance}(i) = R_{\text{REL}}(X_i, Y)$$

$$\text{Redundancy}(i) = \sum_{i' \in var_{\text{sel}}} R_{\text{RED}}(X_i, X_{i'})$$

$$\text{mrMR}(i) = \text{Relevance}(i) - \text{Redundancy}(i)$$

$$i^* = \underset{i}{\text{argmax}}\ \text{mrMR}(i)$$



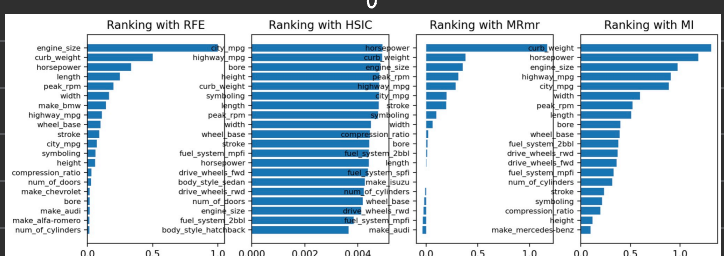
* mrMR subtracts redundancy (correlation)

Search Strategies: $2^D - 1$ subsets → greedy search.

Forward, Backward

Wrapper methods

Directly use performance of final classifier/regressor to evaluate features. → n-fold, CV process



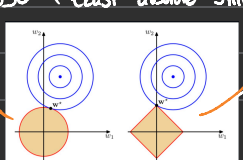
* RFE (recursive feature elim.)

Embedded methods

Carry out selection during train process { Random Forests → * RF sensitivity due to averaging between models.
L1 regularization

L1 regularization is used for overfitting but causes model coef. to drop to zero

`LASSO (least absolute shrinkage selection)



Let's consider a regularized linear regresion model, with L2 (Ridge Regression) and L1 (Lasso) regularizations, to analyze this property. To do this, let's reformulate their formulations as a least squares optimization problem plus a constraint, that is,

- **Ridge regression:**

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2$$

$$\text{s.t.} \sum_{j=1}^{D} w_j^2 \leq t$$

- **Lasso regression:**

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2$$

$$\text{s.t.} \sum_{j=1}^{D} |w_j| \leq t$$