INDIAN INSTITUTE OF TECHNOLOGY MADRAS
सिद्धिर्भवति कर्मजा

# IIT Madras

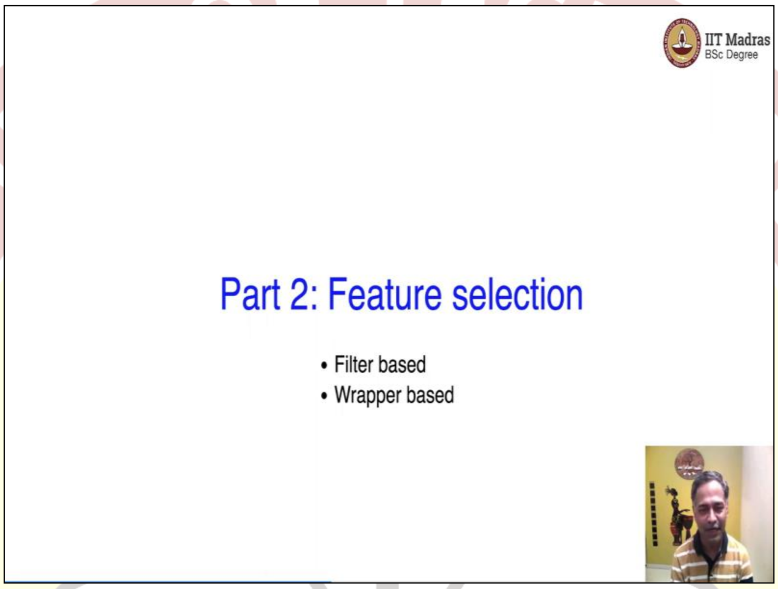
ONLINE DEGREE

# Machine Learning Practice
**Online Degree Programme**
**B. Sc in Programming and Data Science**
**Diploma Level**

**Dr. Ashish Tendulkar**
**Indian Institute of Technology – Madras**

## Part2: Feature Selection

**(Refer Slide Time: 00:12)**



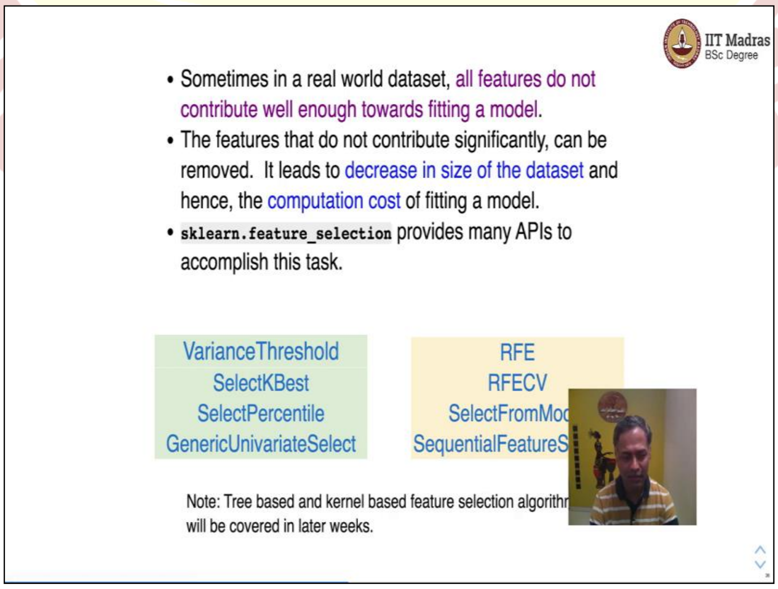
IIT Madras BSc Degree

### Part 2: Feature selection

- Filter based
- Wrapper based

Namaste welcome to the next video of the machine learning practice course. In this video, we will discuss feature selection APIs in Sklearn. There are two types of feature selection APIs one is filter based and the second is wrapper based.

**(Refer Slide Time: 00:29)**



IIT Madras BSc Degree

- Sometimes in a real world dataset, all features do not contribute well enough towards fitting a model.
- The features that do not contribute significantly, can be removed. It leads to decrease in size of the dataset and hence, the computation cost of fitting a model.
- `sklearn.feature_selection` provides many APIs to accomplish this task.

VarianceThreshold
SelectKBest
SelectPercentile
GenericUnivariateSelect

RFE
RFECV
SelectFromMod
SequentialFeatureS

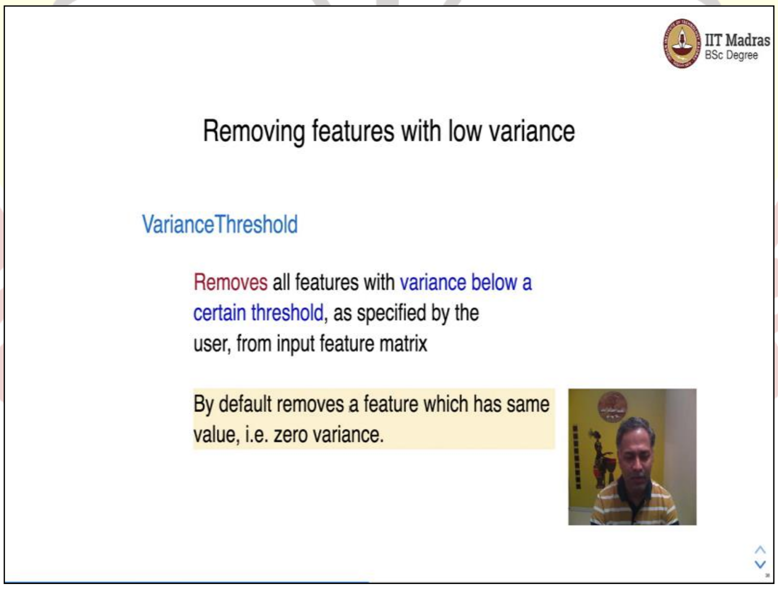
Note: Tree based and kernel based feature selection algorithr
will be covered in later weeks.

Sometimes in real-world datasets, all features do not contribute well enough towards fitting a model. The features that do not contribute significantly can be removed it leading to a decrease in the size of the dataset and hence the computational cost of fitting a model. Sklearn.feature_selection module provides many APIs to accomplish this task. These are some of the classes provided in the feature_selection module.

Variance Threshold, SelectKBest, SelectPercentile, get GenericUnivariateSelect then RFE which is recursive feature elimination. Recursive feature elimination with cross-validation, SelectFromModel and SequentialFeatureSelector. There are some more feature selectors based on trees and kernels which will be covered in later weeks when we study kernels and trees. The first four feature selection methods are basically the filter methods.

The VarianceThreshold, SelectKBest, SelectPercentile and GenericUnivariateSelect are filter based feature selection methods whereas the other four methods are on the right-hand side which is Recursive Feature Elimination Recursive, Feature Elimination with cross-validation, SelectFromModel, and SequentialFeatureSelector are wrapper based methods.

**(Refer Slide Time: 02:15)**



IIT Madras
BSc Degree

Removing features with low variance

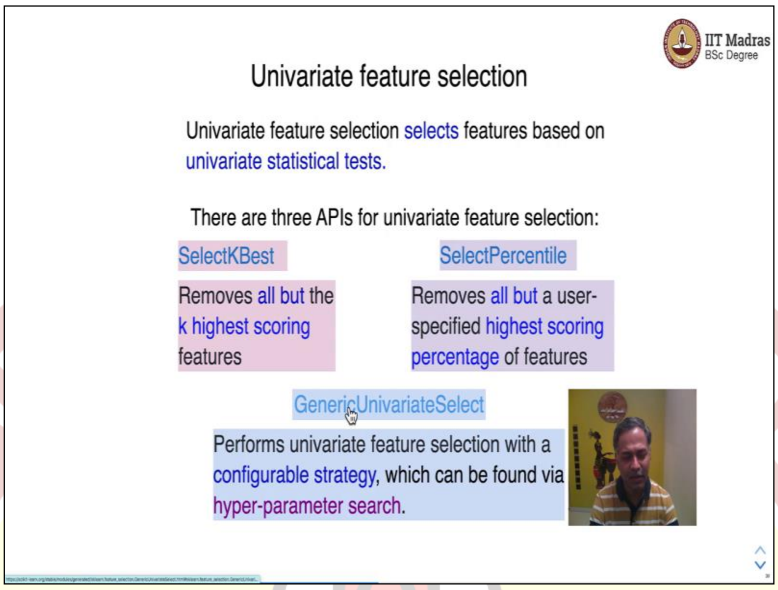
VarianceThreshold

Removes all features with variance below a certain threshold, as specified by the user, from input feature matrix

By default removes a feature which has same value, i.e. zero variance.

Let us look at filter based feature selection methods. So, the first one is VarianceThreshold which removes feature with low variance. What happens sometimes is in a feature in a future column or in a single feature all values are very close to each other or they are like a single value. So, such features have very low variance and therefore it is not so, useful in training. So, what variance threshold does it is it removes all features with variance below a certain threshold and this VarianceThreshold is provided by the user as an argument. So, by default, it removes a feature that has the same value that is zero variance.

**(Refer Slide Time: 03:12)**



## Univariate feature selection

Univariate feature selection selects features based on univariate statistical tests.

There are three APIs for univariate feature selection:

SelectKBest

Removes all but the k highest scoring features

SelectPercentile

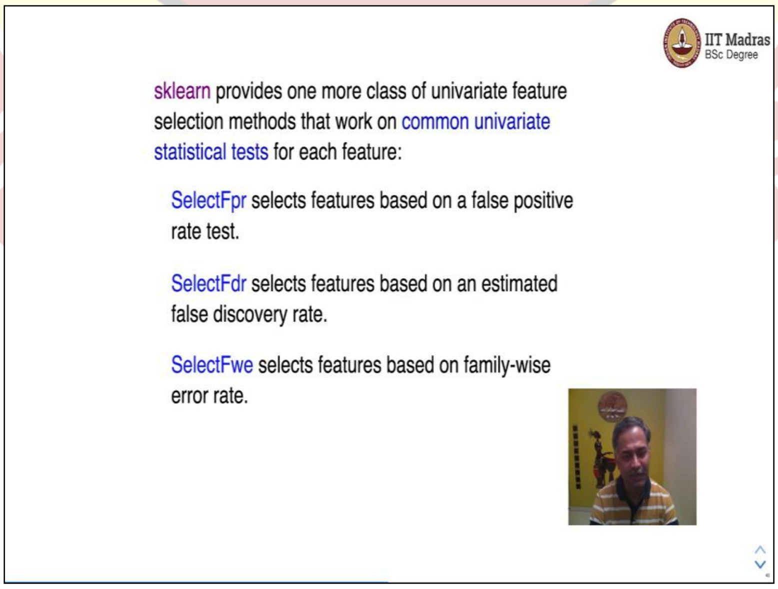
Removes all but a user-specified highest scoring percentage of features

GenericUnivariateSelect

Performs univariate feature selection with a configurable strategy, which can be found via hyper-parameter search.

The univariate feature selection selects features based on univariate statistical tests. There are three APIs for univariate feature selection SelectKBest, SelectPercentile and GenericUnivariateSelect. The SelectKBest removes all but the k highest scoring feature, SelectPercentile removes all but a user-specified highest scoring percentage of features and GenericUnivariateSelect performs univariate feature selection with a configurable strategy which can be found via hyperparameter search.

**(Refer Slide Time: 03:59)**



sklearn provides one more class of univariate feature selection methods that work on common univariate statistical tests for each feature:

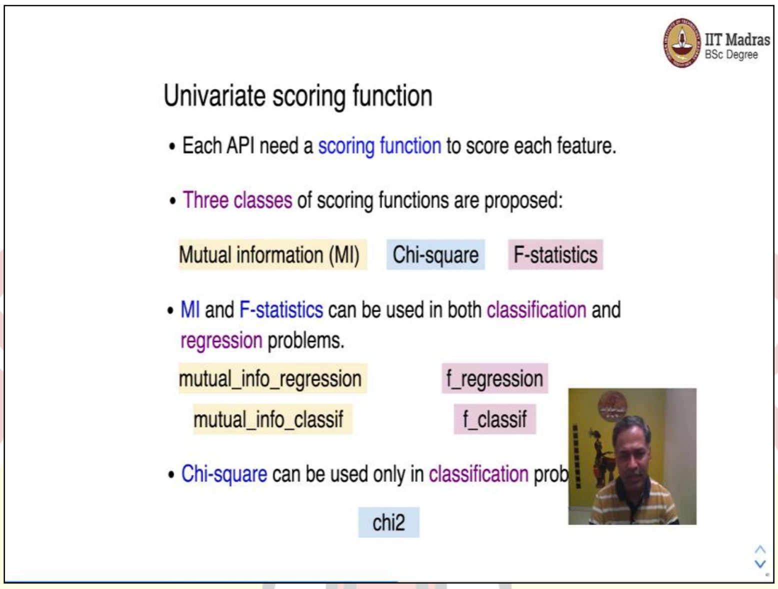
SelectFpr selects features based on a false positive rate test.

SelectFdr selects features based on an estimated false discovery rate.

SelectFwe selects features based on family-wise error rate.

Sklearn provides one more class of univariate feature selection methods that work on common univariate statistical tests for each feature. There is selectFpr that selects features based on a false positive rate test. Then there is selectFdr that selects features based on an estimated false discovery rate and then there is selectFwe that selects features based on family wise error rate.

**(Refer Slide Time: 04:30)**



## Univariate scoring function

- Each API need a scoring function to score each feature.
- Three classes of scoring functions are proposed:

Mutual information (MI) Chi-square F-statistics

- MI and F-statistics can be used in both classification and regression problems.

mutual_info_regression f_regression
mutual_info_classif f_classif

- Chi-square can be used only in classification prob

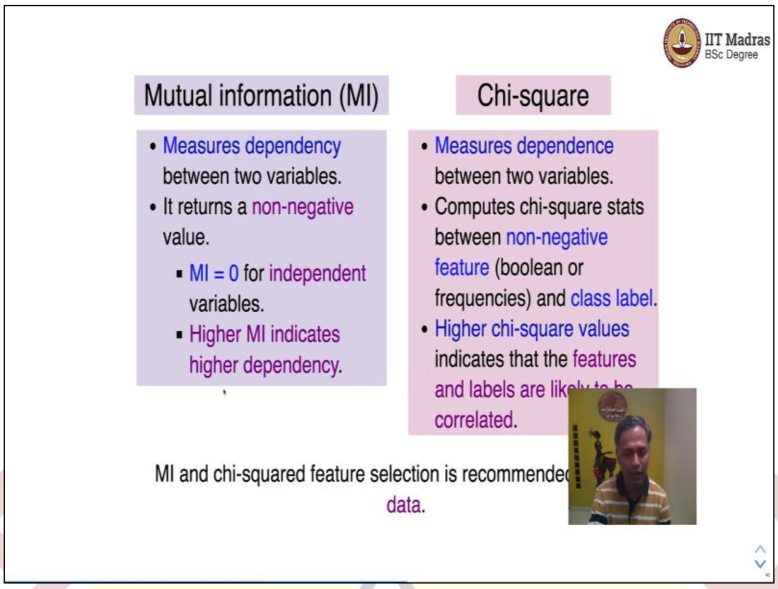
chi2

Let us look at univariate scoring functions. So, each univariate feature selection requires a scoring function to score each feature. There are three classes of scoring functions that are provided in the sklearn feature_selection module. The first one is mutual information second one is chi-square and the third one is F-statistics. Mutual information and F-statistics can be used in both classification and regression problems.

Mutual information is represented as mutual_info_regression for regression problems and mutual_info_classif for the classification problem. Name statistics are represented as F_regression for regression problems and F_classes for classification problems. So, there is a regression and classif suffix for regression and classification respectively.

Whereas the chi-square method can only be used in classification problems it goes by the string chi-square. So, there are these three main scoring functions that we will be using with univariate feature selectors.

**(Refer Slide Time: 05:57)**



IIT Madras
BSc Degree

## Mutual information (MI)

- Measures dependency between two variables.
- It returns a non-negative value.
  - MI = 0 for independent variables.
  - Higher MI indicates higher dependency.

## Chi-square

- Measures dependence between two variables.
- Computes chi-square stats between non-negative feature (boolean or frequencies) and class label.
- Higher chi-square values indicates that the features and labels are likely to be correlated.

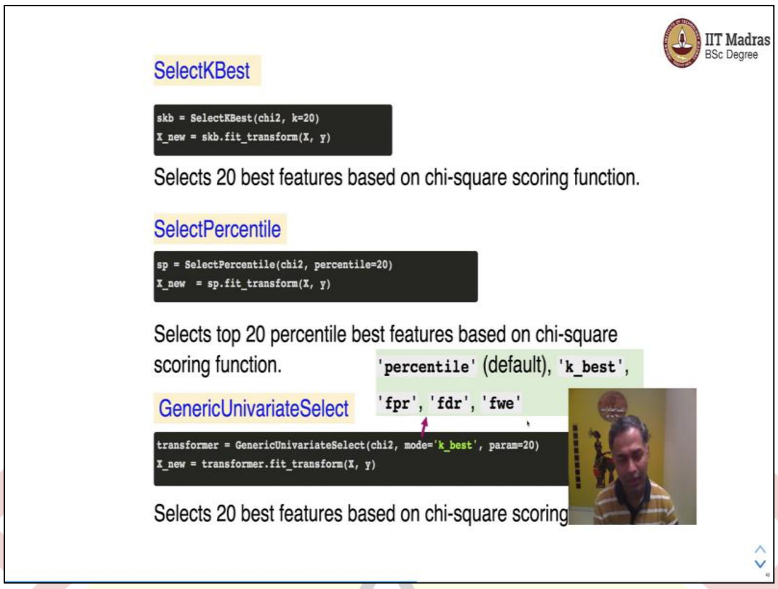
MI and chi-squared feature selection is recommended data.

Let us look at what is mutual information. So, mutual information is a concept from information theory it measures dependency between two variables. It returns a non-negative value that represents the dependency if mutual information is zero that means the two variables are independent they are independent of each other and if there is a high mutual information score that indicates that there is a higher dependency between two variables.

The Chi-square on the other hand measures dependence between two variables we typically compute Chi-square statistics between a non-negative feature and a class label. The non-negative features could be Boolean features or frequencies. The higher Chi-square value indicates that features and labels are likely to be correlated and such features that are correlated with labels are highly useful features for classification problems.

Hence we choose to include such kinds of features with higher Chi-square values. So, mutual information and Chi-squared feature selection are recommended for sparse data. So, we will focus on mainly mutual information and Chi-square for feature selection. We will not discuss if statistics further in this course.

**(Refer Slide Time: 07:41)**



SelectKBest

```
skb = SelectKBest(chi2, k=20)
X_new = skb.fit_transform(X, y)
```

Selects 20 best features based on chi-square scoring function.

SelectPercentile

```
sp = SelectPercentile(chi2, percentile=20)
X_new  = sp.fit_transform(X, y)
```

Selects top 20 percentile best features based on chi-square scoring function.

GenericUnivariateSelect

`'percentile'` (default), `'k_best'`, `'fpr'`, `'fdr'`, `'fwe'`

```
transformer = GenericUnivariateSelect(chi2, mode='k_best', param=20)
X_new = transformer.fit_transform(X, y)
```

Selects 20 best features based on chi-square scoring

Let us look at univariate feature selectors SelectKBest, SelectPercentile and GenericUnivariateSelect. So, the first one, let us look at the concrete example of SelectKBest. So, you can see that here we have instantiated SelectKBest object with Chi-square as the scoring function and we want k equal to 20 which is we want to select 20 best features based on the Chi-square scoring function.

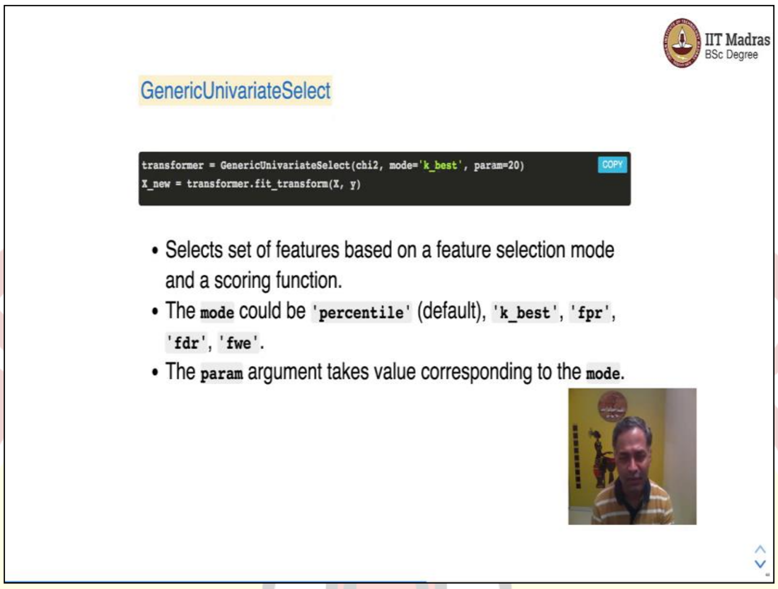
And then we apply the fit_transfer method on the feature matrix and the label vector and we obtain a transform feature matrix. So, this one selects the 20 best features based on the Chi-square scoring function. This is a small code snippet for select percentile we instantiate an object of select percentile class specify chi-square as a scoring function and specify that we want top 20 features.

And we apply fit_transform with feature matrix and label vector as arguments to obtain a transform feature matrix. So, this code snippet selects top 20 percentile-based features based on the Chi-square scoring function. We can specify the same thing as select k best with GenericUnivariateSelect. Here we instantiate an object of GenericUnivariateSelect with Chi-square as a scoring function the mode as KBest mode and the parameter is 20.

So, this is the parameter of the KBest method which will help us select 20 best features then we apply fit_transfer method on the on the feature matrix and the label vector to obtain a transform feature matrix. So, this code snippet selects 20 best feature based on Chi-square scoring function just like the select KBest and these are two equivalent ways of specifying the same things.

So, in GenericUnivariateSelect we can use other modes like percentile which is the default mode in GenericUnivariateSelect then there is k_best or you can use Fpr, Fdr and Fwe.

**(Refer Slide Time: 10:29)**



GenericUnivariateSelect

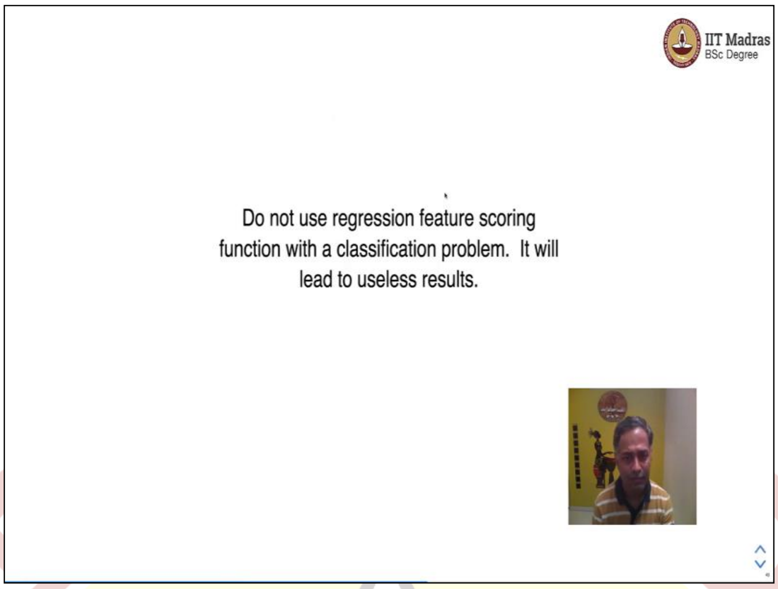
```
transformer = GenericUnivariateSelect(chi2, mode='k_best', param=20)
X_new = transformer.fit_transform(X, y)
```

- Selects set of features based on a feature selection mode and a scoring function.
- The `mode` could be `'percentile'` (default), `'k_best'`, `'fpr'`, `'fdr'`, `'fwe'`.
- The `param` argument takes value corresponding to the `mode`.

So, these are some of the modes that we discussed in the earlier slides. So, this GenericUnivariateSelect code snippet selects aset of features based on a feature selection mode and a scoring function that is this is the mode and this is a scoring function. The first argument is the scoring function second argument is the mode. The modes could be percentile which is by default or K_Best a Fpr Fdr and Fwe and the parameter arguments takes value corresponding to the mode.

So, whatever is the mode this parameter is passed to this particular model and then this parameter in this mode is used to calculate the best possible features.

**(Refer Slide Time: 11:20)**


Do not use regression feature scoring function with a classification problem. It will lead to useless results.

Finally, one word of caution for you do not use the regression feature scoring function with a classification problem. it will lead to useless results. So, that is it from the filter-based feature selection method. In the next video, we will discuss wrapper based feature selection methods.