

graphlab.decision_tree_classifier.DecisionTreeClassifier.pred

`DecisionTreeClassifier.predict(dataset, output_type='class', missing_value_action='auto')`

A flexible and advanced prediction API.

The target column is provided during `create()`. If the target column is in the *dataset* it will be ignored.

Parameters: **dataset** : SFrame

A dataset that has the same columns that were used during training. If the target column exists in `dataset` it will be ignored while making predictions.

output_type : {'probability', 'margin', 'class', 'probability_vector'}, optional.

Form of the predictions which are one of:

- **'probability': Prediction probability associated with the True**
class (not applicable for multi-class classification)
- **'margin':** Margin associated with the prediction (not applicable for multi-class classification)
- **'probability_vector':** Prediction probability associated with each class as a vector. The probability of the first class (sorted alphanumerically by name of the class in the training set) is in position 0 of the vector, the second in position 1 and so on.
- **'class': Class prediction. For multi-class classification, this**
returns the class with maximum probability.

missing_value_action : str, optional

Action to perform when missing values are encountered. Can be one of:

- **'auto':** By default the model will treat missing value as is.
- **'impute':** Proceed with evaluation by filling in the missing values with the mean of the training data. Missing values are also imputed if an entire column of data is missing during evaluation.
- **'error':** Do not proceed with evaluation and terminate with an error message.

Returns: **out** : SArray

Predicted target value for each example (i.e. row) in the dataset.

! See also

`create`, `evaluate`, `classify`

Examples

```
>>> m.predict(testdata)
>>> m.predict(testdata, output_type='probability')
>>> m.predict(testdata, output_type='margin')
```