svm_regressor.predict

```

Copy fnpredict(refself:SVMRegressor,X:Tensor)->Tensor;

```

Support Vector Machine regression prediction and one-class SVM anomaly detection.

Args

- self
- : SVMRegressor - A SVMRegressor object.
- X
- : Input 2D tensor.
-

Returns

- Tensor containing the Support Vector Machine regression prediction and one-class SVM anomaly detection of the input X.
-

Type Constraints

SVMRegressor andX must be fixed points

Examples

```

Copy useorion::numbers::FP16x16; useorion::operators::tensor::{Tensor,TensorTrait,FP16x16Tensor,U32Tensor}; useorion::operators::tensor::FP16x16TensorPartialEq;

useorion::operators::ml::svm::svm_regressor::{SVMRegressorTrait,POST_TRANSFORM,SVMRegressor}; useorion::operators::ml::svm::core::{KERNEL_TYPE};

fnexample_svm_regressor_linear()->Tensor { letcoefficients:Span=array![ FP16x16{ mag:65536, sign:false}, FP16x16{ mag:65536, sign:true}, FP16x16{ mag:54959, sign:false}, FP16x16{ mag:54959, sign:true}, FP16x16{ mag:29299, sign:false}, FP16x16{ mag:65536, sign:true}, FP16x16{ mag:36236, sign:false} ] .span(); letn_supports:usize=7; letone_class:usize=0; letrho:Span=array![FP16x16{ mag:35788, sign:false}].span(); letsupport_vectors:Span=array![ FP16x16{ mag:8421, sign:true}, FP16x16{ mag:5842, sign:false}, FP16x16{ mag:4510, sign:false}, FP16x16{ mag:5202, sign:true}, FP16x16{ mag:14783, sign:true}, FP16x16{ mag:17380, sign:true}, FP16x16{ mag:60595, sign:false}, FP16x16{ mag:1674, sign:true}, FP16x16{ mag:38669, sign:true}, FP16x16{ mag:63803, sign:false}, FP16x16{ mag:87720, sign:true}, FP16x16{ mag:22236, sign:false}, FP16x16{ mag:61816, sign:false}, FP16x16{ mag:34267, sign:true}, FP16x16{ mag:36418, sign:false}, FP16x16{ mag:27471, sign:false}, FP16x16{ mag:28421, sign:false}, FP16x16{ mag:69270, sign:true}, FP16x16{ mag:152819, sign:false}, FP16x16{ mag:4065, sign:false}, FP16x16{ mag:62274, sign:true} ] .span(); letpost_transform=POST_TRANSFORM::NONE; letkernel_params:Span=array![ FP16x16{ mag:27812, sign:false}, FP16x16{ mag:0, sign:false}, FP16x16{ mag:196608, sign:false} ] .span(); letkernel_type=KERNEL_TYPE::LINEAR;

letmutregressor:SVMRegressor=SVMRegressor{ coefficients, kernel_params, kernel_type, n_supports, one_class, post_transform, rho, support_vectors, };

letmutX:Tensor=TensorTrait::new( array![3,3].span(), array![ FP16x16{ mag:32768, sign:true}, FP16x16{ mag:26214, sign:true}, FP16x16{ mag:19660, sign:true}, FP16x16{ mag:13107, sign:true}, FP16x16{ mag:6553, sign:true}, FP16x16{ mag:0, sign:false}, FP16x16{ mag:6553, sign:false}, FP16x16{ mag:13107, sign:false}, FP16x16{ mag:19660, sign:false}, ] .span() );

returnSVMRegressorTrait::predict(refregressor,X); }

                    [[-0.468206], [0.227487], [0.92318]]

```

Previous SVM Regressor Next Sequence

Last updated2 months ago