Column

- name: String {readOnly}
- type: String {readOnly}
- + getColumnName(): String
- + isNumber(): boolean
- + toString(): String

<< Enumeration >>

QUERY_TYPE

MIN

MAX

<<Exception>>

DatabaseConnectionException

DbAccess

- DRIVER_CLASS_NAME: String {readOnly}
- DBMS: String {readOnly}
- SERVER: String {readOnly}
- PORT: int {readOnly}
- DATABASE: String {readOnly}
- USER_ID: String {readOnly}
- PASSWORD: String {readOnly}
- con: Connection
- + getConnection(): Connection
- + closeConnection()

<<throws>>

<<Exception>>

InsufficentColumnNumberException

TableData

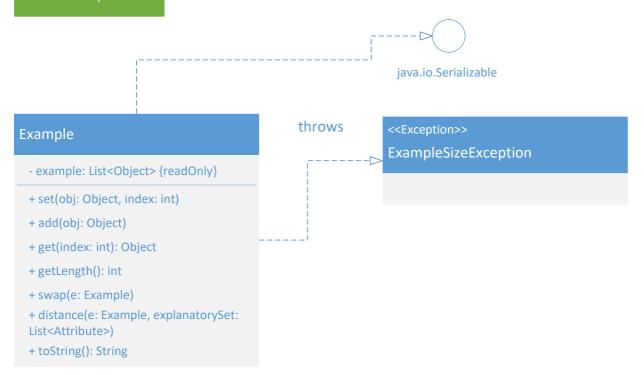
- db: DbAccess
- table: String
- tSchema: TableSchema
- transSet: List<Example>
- target: List
- init()
- + getExamples(): List<Example>
- + getTargetValues(): List
- + getAggregateColumnValue(column:
- Column, aggregate: QUERY_TYPE): Object

TableSchema

- tableSchema: List<Column>
- target: Column
- tableName: String
- + target(): Column
- + getNumberOfAttributes(): int
- + getTableName(): String
- + iterator(): Iterator<Column>

<<throws>>

example



mining

java.io.Serializable

KNN

- trainingSet: Data {readOnly}

+ predict(out: ObjectOutputStream, in: ObjectInputStream): double

+ salva(nomeFile: String)

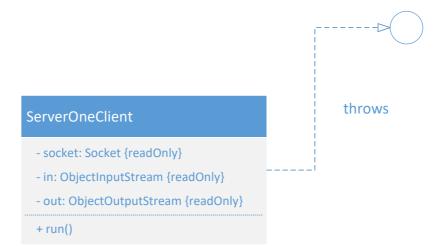
+ carica(nomeFile: String): KNN

+ toString(): String

data.TrainingDataException

MultiServe

- PORT: int {readOnly}
- run()
- + main(args String[])



client

Client

- socket: Socket {readOnly}
- out: ObjectOutputStream {readOnly}
- in: ObjectInputStream {readOnly}
- + serverOut(output: Object)
- + serverIn(): Object
- + close()

ClientApplication

- c: Client
- mainMessage: Label
- stage: Stage
- vBox: VBox
- ip: String[]
- + start(primaryStage: Stage)
- TableNameScene(message: String)
- trainingSetScene()
- example()
- buildExample(risposta: StringBuilder, flag: boolean)
- prediction(text: Label)
- confirm(vBox: VBox)
- main(args: String[])