

```

SetDirectory[NotebookDirectory[]];
raw = OpenRead["reviews.csv"];
reviews = Map[StringSplit[#, "|"] &, ReadList[raw, String]];
(*start pre-processing data*)
(*spell checking, tokenization, remove symbols*)
(*extract words from dataset*)
numOfReviews = Length[reviews];
indexOfReviews = 2;
$RecursionLimit = Infinity;
reviewsWords = List[];
While[indexOfReviews ≤ 10 000,
  review = reviews[[indexOfReviews]];
  reviewStr = review[[2]];
  newReviewStr = DeleteStopwords[reviewStr];
  reviewWords = TextWords[newReviewStr];
  reviewWords = WordStem[reviewWords];
  reviewWords = ToLowerCase[reviewWords];
  reviewWords = DeleteDuplicates[reviewWords];
  numOfWords = Length[reviewWords];
  indexOfWord = 1;
  indexesOfDeleteWord = List[];
  correctedReviewWords = List[];
  While[indexOfWord ≤ numOfWords,
    If[DictionaryWordQ[reviewWords[[indexOfWord]]] == True, correctedReviewWords =
      Append[correctedReviewWords, reviewWords[[indexOfWord]]];
    ];
    indexOfWord++;
  ];

  reviewsWords = Append[reviewsWords, correctedReviewWords];
  indexOfReviews++;
];
Export["pre-processingWords_10k.mx", reviewsWords];
(*feature extractor TF-IDF*)

(*Dimension Reduce*)

(*create training dataset*)
(*create testing dataset*)
(*train SVM*)
(*test*)

```

```

SetDirectory[NotebookDirectory[]];
raw = OpenRead["reviews.csv"];
reviews = Map[StringSplit[#, "|"] &, ReadList[raw, String]];
$RecursionLimit = $IterationLimit = Infinity;
reviewsWords = Import["pre-processingWords_10k.mx"]
fe = FeatureExtraction[reviewsWords, "TFIDF"]
features = fe[reviewsWords];
listOfFeatures = List@@ features;
testData = Take[listOfFeatures, {1, -1, 5}];
trainData = Drop[listOfFeatures, {1, -1, 5}];
indexOfReviews = 2;
classes = List[];
While[indexOfReviews ≤ 10 000,
  review = reviews[[indexOfReviews]];
  class = Take[review, 1];
  classes = Append[classes, class[[1]]];
  indexOfReviews++;
];
testClasses = Take[classes, {1, -1, 5}];
trainClasses = Drop[classes, {1, -1, 5}];
trainingSet = trainData → trainClasses;
testingSet = testData → testClasses;

```

```

{{great, love, pat, listen, year, good, mood, make, feel, better, bad, just,
  like, sugar, rain, life, vocal, lyric, kill, hidden, gem, desert, isl,
  book, big, beyond, matter, black, white, young, old, male, thing, sing},
  ... 9997 ... , {leg, distress, item, 6, month, old, love,
  ... 7 ... , place, fine, lose, unless, doubt, us, longer}}

```

large output

show less

show more

show all

set size limit...

FeatureExtractorFunction[



Input type: **NominalSequence**
 Output type: **NumericalVector** (length: 8532)

]

```

$RecursionLimit = $IterationLimit = Infinity;
svm = Classify[trainingSet, Method → "SupportVectorMachine"];
(*rf = Classify[trainingSet, Method → "RandomForest"];
nn = Classify[trainingSet, Method → "NeuralNetwork"];*)
svmCM = ClassifierMeasurements[svm, testingSet];
(*rfCM = ClassifierMeasurements[rf, testingSet];
nnCM = ClassifierMeasurements[nn, testingSet];*)
svmCM["Accuracy"]
svmCM["Precision"]
svmCM["Recall"]
svmCM["ROCCurve"]
(*rfCM["Accuracy"]
rfCM["Precision"]
rfCM["Recall"]
rfCM["ROCCurve"]
nnCM["Accuracy"]
nnCM["Precision"]
nnCM["Recall"]
nnCM["ROCCurve"]*)

```

0.7435

⟨ | negative → 0.710599, positive → 0.782514 | ⟩

⟨ | negative → 0.794845, positive → 0.695146 | ⟩

