NEC Advanced Analytics -

RAPID 機械学習
マッチング　for Linux V2.2
ユーザガイド

第 1.0 版
2018 年 5 月

日本電気株式会社

はしがき

本書は、「NEC Advanced Analytics - RAPID 機械学習 マッチング for Linux V2.2」（以下、本製品と記載）の利用者のための説明書です。本製品の利用方法について解説しています。本製品の利用前に、必ずお読みください。

また、本製品に関する説明書は、セットアップ媒体に格納されています。必要に応じて、参照してください。

＜商標について＞

- Windows 、Internet Explorer 、Hyper-V 、Excel 、PowerPoint は 、米 国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。
- Intel、Intel Core、Xeon は、米国 Intel 社の商標または登録商標です。
- Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。
- Red Hat は、米国 Red Hat, Inc.ならびにその子会社の登録商標です。
- その他、本書に記載されている会社名、製品名は、一般に各社の商標または登録商標です。

＜略語・用語について＞

- Windows の正式名称は、Microsoft Windows Operating System です。

＜輸出する際の注意事項＞

本製品（ソフトウェアを含む）は、外国為替及び外国貿易法で規定される規制貨物（または役務）に該当することがあります。 その場合、日本国外へ輸出する場合には日本国政府の輸出許可が必要です。 なお、輸出許可申請手続にあたり資料等が必要な場合には、お買い上げの販売店またはお近くの当社営業拠点にご相談ください。

更新履歴

版数	日付	改版内容
1.0 版	2018 年 5 月 21 日	初版

目次

第1章 はじめに

本章では、本書で使用する用語を定義します。

1.1 用語集

本書で使用する用語を表 1-1 に定義します。

表 1-1 本書で使用する用語の説明

用語	説明
機械学習	人間が自然に行っている学習能力と同様の機能を計算機で実現しようとする技術・手法です。一般に、学習データを分析して、有用な規則、ルール、知識表現、判断基準などを抽出して予測モデルを生成した後、予測データに予測モデルを適用して何らかの判断を行う一連の手続きを指します。
予測モデル	学習データを分析して、有用な規則、ルール、知識表現、判断基準などを抽出したものです。計算機が予測データに対して何らかの判断を行う際に使用します。
学習データ	計算機が予測モデルを作るために使用するデータ群です。マッチングでは属性データ(クエリ)、属性データ(ターゲット)、正解ラベルの 3 つ組からなるデータ群、フィルタリングでは属性データ(クエリ)、正解ラベルの 2 つ組からなるデータ群を指します。
検証データ	学習で作られる予測モデルの精度を検証するためのデータ群です。 学習データと同じく、マッチングでは属性データ(クエリ)、属性データ(ターゲット)、正解ラベルの 3 つ組からなるデータ群、フィルタリングでは属性データ(クエリ)、正解ラベルの 2 つ組からなるデータ群です。
予測データ	計算機が予測モデルを用いて何らかの判断をする際に対象とするデータです。本書では、マッチングでは求職者データ、求人企業データの 2 つ組からなるデータ群、フィルタリングでは求職者データを指します。予測データは正解ラベルを持ちません。
属性データ	分析の対象とする属性を持ったデータです。 例えば求職者データでは求職者に関する様々な属性値を集めたデータ、求人企業データでは求人企業に関する様々な属性値を集めたデータが属性データとなります。
求職者データ	本書では属性データ(クエリ)の例として、求職者データを用います。 求職者に関する様々な属性値を集めたデータです。性別、年齢、学歴などのデモグラフィックや、職務履歴書などが代表的です。
求人企業データ	本書では属性データ(ターゲット)の例として、求人企業データを用います。 求人企業に関する様々な属性値を集めたデータです。業種・業態、約款などの企業情報や求人票などが代表的です。
正解ラベル	計算機に学習させる判断基準を記述したデータです。教師ラベルとも言います。本書では、求職者データと求人企業データの関係性(合格・不合格など)を記述します。
分類問題	属性データをカテゴリに分類する問題です。 分類問題では、正解ラベルとして合格・不合格、P(マッチする)・N(マッチしない)といったカテゴリを正解とします。
回帰問題	属性データから数値を予測する問題です。 分類問題では、正解ラベルとして 0.1、-0.1、100、といった実数値を正解とします。
SSI	NEC が独自開発したマッチング専用の機械学習アルゴリズムです。Supervised Semantic Indexing の略称です。
SSE	NEC が独自開発したフィルタリング専用の機械学習アルゴリズムです。Supervised

	Sequence Embedding の略称です。
マッチング	求職者データ、求人企業データの 2 つ組を入力データとして、両者の適合度合いをマッチングスコアとして数値化する機械学習アルゴリズム全般を指します。
フィルタリング	求職者データの 1 つ組を入力データとして、その適合度合いをフィルタリングスコアとして数値化する機械学習アルゴリズム全般を指します。マッチングにおいて、求人企業データを固定した特殊ケースに相当します。
マッチングスコア	SSI などマッチング専用の機械学習アルゴリズムが算出する、2 つ組の入力データの適合度合いを表す予測値です。
フィルタリングスコア	SSE などフィルタリング専用の機械学習アルゴリズムが算出する、1 つ組の入力データの適合度合いを表す予測値です。
管理者ユーザ	Linux において、管理者ユーザ権限を持つユーザアカウントを指します。本書では、本製品をインストールする際に使用します。
標準ユーザ	Linux において、標準ユーザ権限を持つユーザアカウントを指します。本書では、本製品を使用する際に使用します。

第2章 本製品の概要

本章では、本製品の概要、システム構成、モジュール概要について説明します。

2.1 RAPID 機械学習 マッチング for Linux V2.2 とは

本ソフトウェアは、NEC 独自のディープラーニング・アルゴリズムによるマッチング/フィルタリング・アプリケーションです。本ソフトウェアは、NEC 製の商用プロプライエタリ製品です。本製品は、2 つの対象の間のマッチング・テンプレートと、1 つの対象に対するフィルタリング・テンプレートを提供します。

本書ではマッチング・テンプレートの 2 つの対象を求職者データ、求人企業データ、フィルタリング・テンプレートの 1 つの対象を求職者データとして説明します。

2.1.1 求職者と求人企業のマッチング

マッチングに関する NEC 独自の機械学習アルゴリズムである Supervised Semantic Indexing（以下、SSI と記載）を活用した、求職者と求人企業のマッチング機能を提供します。本機能の概要を図 2-1 に記載します。

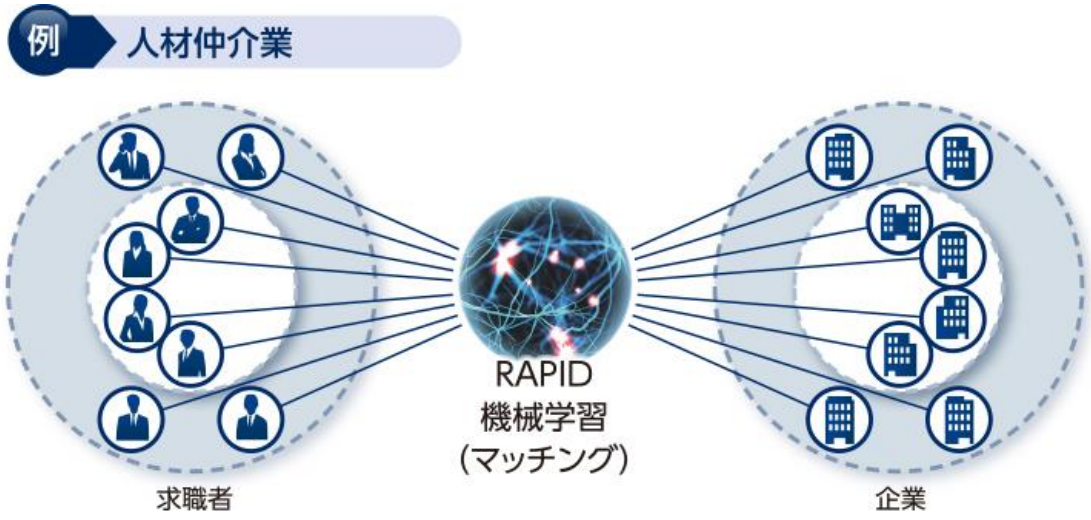

図 2-1 求職者と求人企業のマッチング

2.1.2 求職者のフィルタリング

フィルタリングに関する NEC 独自の機械学習アルゴリズムである Supervised Sequence Embedding(以下、SSE と記載)を活用した、求職者のフィルタリング機能を提供します。本機能の概要を図 2-2 に記載します。

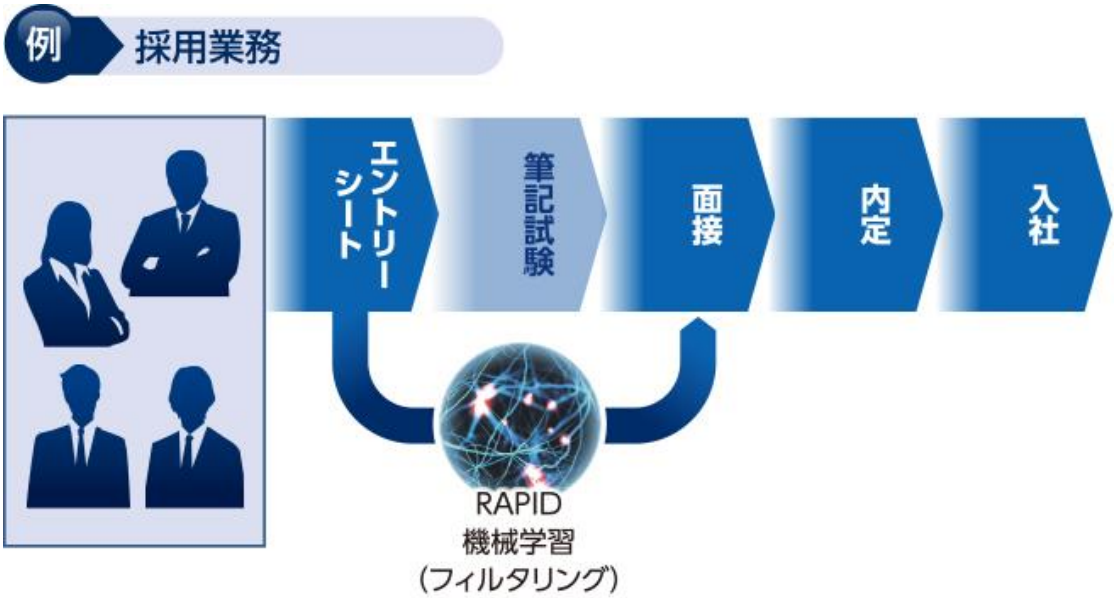

図 2-2 求職者のフィルタリング

2.2 システム構成

本製品は、2.1 節に記載した NEC 独自の機械学習アルゴリズム(マッチング／フィルタリング)を活用するために必要となるコマンド群(データ加工、学習、バッチ予測、各種ヘルパー)を利用者に提供します。本製品のシステム構成を図 2-3 に記載します。

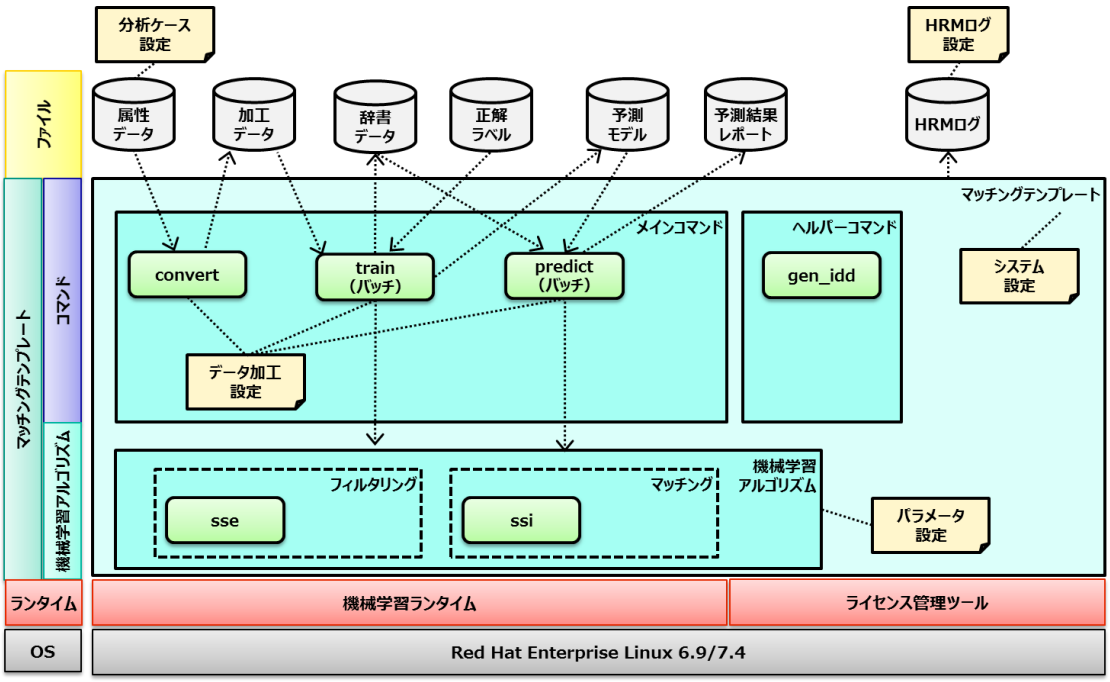

図 2-3 本製品のシステム構成

2.3 モジュール概要

図 2-3 の各モジュールの説明を表 2-1 に記載します。各モジュールの使用方法については、第 3 章 を参照してください。

表 2-1 本製品を構成するモジュールの説明

用語	説明
属性データ	本製品の入力データです。求職者データと求人企業データから構成され、求職者と求人企業に関する属性値を格納します。データ形式は 4.2.1 節を参照してください。
加工データ	データ加工設定に従い、convert コマンドで属性データを加工したデータです。
辞書データ	train コマンドで加工データから生成した特徴を辞書化したデータです。
正解ラベル	求職者データの求職者 ID と求人企業データの求人企業 ID に対して、正解／不正解ラベルを付与したデータです。データ形式は 4.2.2 節を参照してください。
予測モデル	train コマンドで学習データ（特徴ベクトル、正解ラベル）から生成したデータです。予測データ（特徴ベクトル）から予測値（マッチングスコア／フィルタリングスコア）を算出する際に使用します。
予測結果レポート	predict コマンドで予測データから生成したデータです。予測データに対する予測値を格納します。データ形式は 4.3.1 節を参照してください。
実行ログ	本製品のログファイルです。データ形式は 6.1 節を参照してください。
分析ケース設定	属性データ、データ加工設定、正解ラベルからなる分析ケースを指示する設定ファイルです。データ形式は 5.1.2 節を参照してください。
データ加工設定	属性データのデータ加工方法を指示する設定ファイルです。データ形式は 5.1.3 節を参照してください。
システム設定	本製品のシステム設定を格納する設定ファイルです。データ形式は 5.1.1 節を参照してください。

パラメータ設定	本製品の機械学習アルゴリズムが使用するパラメータを指示する設定ファイルです。データ形式は 5.1.4 節を参照してください。
実行ログ設定	本製品の実行ログに関する設定を格納する設定ファイルです。データ形式は 5.1.5 節を参照してください。
convert コマンド	属性データから加工データを生成するコマンドです。データ加工設定を参照します。コマンド形式は 4.1.1 節を参照してください。
train コマンド	学習データ（特徴ベクトル、正解ラベル）から予測モデルを生成するコマンドです。パラメータ設定を参照します。コマンド形式は 4.1.2 節、4.1.3 節を参照してください。
predict コマンド	予測データ（特徴ベクトル）から予測結果レポートを生成しファイル出力するバッチ予測コマンドです。パラメータ設定を参照します。コマンド形式は 4.1.4 節、4.1.5 節を参照してください。
gen_idd コマンド	属性データからデータ加工設定の雛形を生成するヘルパーコマンドです。コマンド形式は 4.1.6 節を参照してください。
sse アルゴリズム	NEC が独自開発したフィルタリング専用の機械学習アルゴリズムです。Supervised Sequence Embedding の略称です。
ssi アルゴリズム	NEC が独自開発したマッチング専用の機械学習アルゴリズムです。Supervised Semantic Indexing の略称です。
機械学習ランタイム	本製品の rpd スクリプトや機械学習アルゴリズムの実行基盤となるランタイムです。

第3章 使用方法

本章では、本製品の使用方法について説明します。

なお、本章では、セットアップガイドに記載した動作環境のうち、「Red Hat Enterprise Linux 7.4」を想定してセットアップ方法を説明しますが、「Red Hat Enterprise Linux 6.9」と手順が異なる場合は、その旨を明記します。

3.1 データ分析の流れ

本製品を活用したデータ分析の流れを図 3-1 に記載します。データ分析は、①データ加工（convert）、②ベクトル化（vectorize）、③学習（train）、④予測（predict）、の 4 つの分析作業から構成されます。これらの分析作業の概要を表 3-1 に記載します。

図 3-1 データ分析の流れ

表 3-1 分析作業の概要

分析作業	概要説明	対応するコマンド
データ加工 （convert）	属性データから加工データを生成します。	convert コマンド
ベクトル化 （vectorize）	加工データから特徴ベクトルを生成します。 本製品では学習コマンド、予測コマンド内で実施されます。	train コマンド predict コマンド
学習 （train）	特徴ベクトルと正解ラベルの関係性を学習し、予測モデルを生成します。	train コマンド
予測 （predict）	予測モデルを用いて、未知の特徴ベクトルに対する予測値を算出します。	predict コマンド

3.2 予測精度の定量評価

図 3-1 のデータ分析結果として得られる予測精度は、定量評価により、その良し悪しを判断する必要があります。予測精度を定量評価するための代表的な方法を表 3-2 に記載します。定量評価の結果、本製品が期待する予測精度を達成していないと判断した場合は、予測精度を改善するための調整作業を行う必要があります。

表 3-2 予測精度の代表的な評価手法

予測精度評価手法	概要説明
学習誤差評価	学習データを学習して予測モデルを生成した後、学習データに対して予測モデルを適用し、学習データに対する予測精度を算出します。得られた予測精度と正解ラベルとの差異の多寡を評価します。 この手法では、予測モデルが学習データをどの程度説明できるのか（どの程度学習できたのか）を定量評価することができます。セルフ・バリデーションともいいます。
汎化誤差評価	入力データを k 分割し、(k-1)分割を学習データ、残りの 1 分割を予測データとします。学習データを学習して予測モデルを生成した後、予測データに対して予測モデルを適用し、予測データに対する予測精度を算出します。得られた予測精度と正解ラベルとの差異の多寡を評価します。k 分割したデータを 1 分割分ずつ入れ替えながらこの手順を k 回繰り返します。この評価のイメージ図を図 3-2 に記載します。 この手法では、予測モデルが未知データをどの程度説明できるのか（どの程度汎化できたのか）を定量評価することができます。クロス・バリデーションまたは交差検定ともいます。

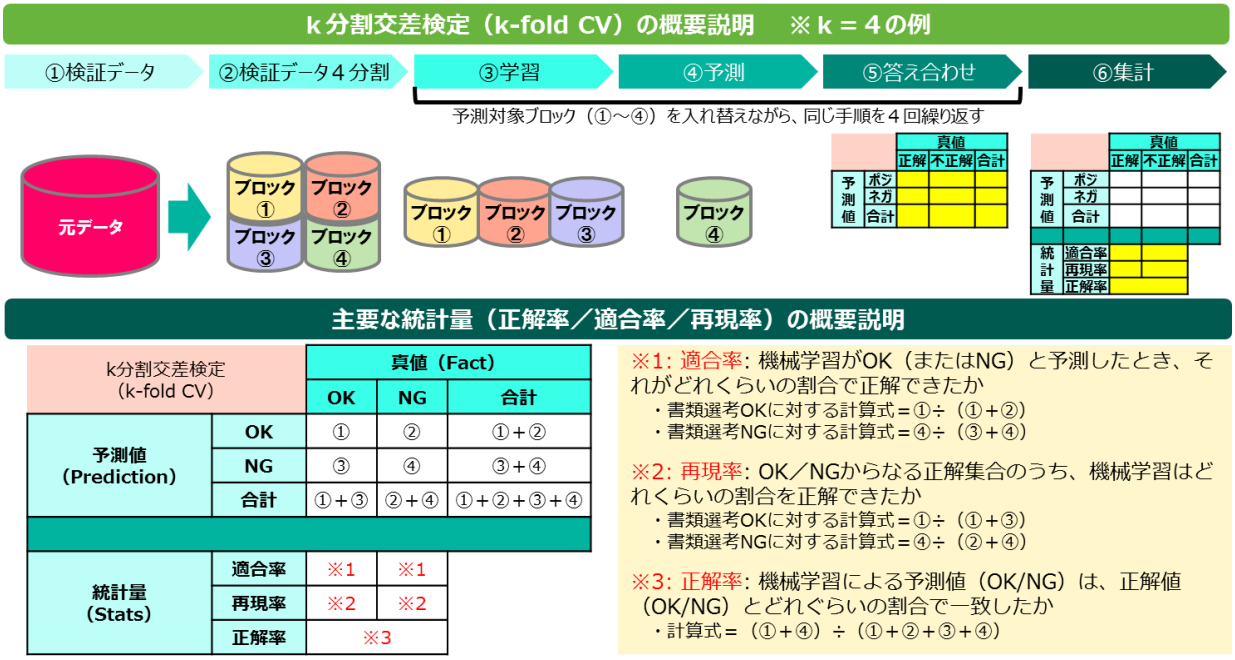

図 3-2 汎化誤差評価のイメージ図

3.3 本製品が提供する機械学習アルゴリズム

本製品は、2 つの機能（マッチング、フィルタリング）、2 つの機械学習アルゴリズム（ssi／sse）を提供します。機能と機械学習アルゴリズムの関係を表 3-3 に記載します。

表 3-3 提供機能と機械学習アルゴリズムの関係

機能	機械学習アルゴリズム	概要説明
マッチング	ssi	マッチングに関する NEC 独自の機械学習アルゴリズムである SSI を活用したマッチング機能です。求職者[q]、求人企業[t]、この求職者と求人企業のマッチ度[l; 1:マッチする、0:マッチしない]のデータ組（q, t, l）を入力データとする特徴があります。
フィルタリング	sse	フィルタリングに関する NEC 独自の機械学習アルゴリズムである SSE を活用したフィルタリング機能です。求職者[q]、この求職者のマッチ度[l; 1:マッチする、0:マッチしない]のデータ組（q, l）を入力データとする特徴があります。

以降、本製品が提供する機械学習アルゴリズム（ssi／sse）毎に、表 3-1 の分析作業、表 3-2 の予測精度評価手法をそれぞれ行う手順を説明します。

なお、本章では、本製品のサンプルデータを用いて説明を進めますが、その他の実データを利用する場合には、必要に応じて手順を読み替えてください。**※注 1、※注 2**

⚠注 1: 機械学習アルゴリズムによる手順の差異

マッチングとフィルタリングでは、データ分析の流れは同じですが、入力ファイル、出力ファイル、設定ファイルの数や構成に差異があります。このため、本章では、本製品が提供する機械学習アルゴリズム（ssi／sse）毎に、本製品の使用手順をそれぞれ説明します。

⚠注 2: 分析作業の実行ユーザ

本章の動作確認方法は、実行ユーザとして標準ユーザを想定しています。
管理者ユーザでログインしている場合は、いったんログアウトし、標準ユーザでログインし直してください。

3.4 マッチング機能（ssi）

マッチングに関する NEC 独自の機械学習アルゴリズムである SSI を活用したマッチング機能です。求職者[q]、求人企業[t]、この求職者と求人企業のマッチ度[l; 1:マッチする、0:マッチしない]のデータ組（q, t, l）を入力データとする特徴があります。

この特徴は、入力データに対して間隔尺度（interval scale）が定義できる場合に有効です。例えば、求職者 A は求人企業 B とマッチングしない（0）、求職者 A は求人企業 C とマッチングする（1）というように、求職者[q]と求人企業[t]のデータ組（q,t）に対して、マッチングする（1）、マッチングしない（0）を定義できる場合に有効です。このケースでは、求職者 A を[q]、求人企業 B を[t-]、求人企業 C を[t+]として、データ組（q,t-）に対してマッチングスコア 0、データ組（q,t+）に対してマッチングスコア 1 を算出するよう、機械学習アルゴリズムにより予測モデルが調整されます。

3.4.1 データ加工

属性データから加工データを生成する作業です。本製品が提供する convert コマンドを利用して作業します。

1. 本製品をセットアップしたマシンに標準ユーザでログインします。

2. データ加工する属性データを準備します。
   属性データは、「**＜インストールパス＞**/template/matching/project_sample/ssi」にあらかじめ用意してありますので、任意のディレクトリにコピーしてください。以降では、このディレクトリを**＜データパス＞**と記載します。属性データは、既定では「/opt/nec/pyrapid/template/matching/project_sample/ssi/」に格納されています。

3. 「**＜データパス＞**/data_conf_ssi.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、**[root_path]**設定項目に**＜データパス＞**を設定します。**※注 3**

```
“root_path”："＜データパス＞",
```

⚠注 3: root_path の設定値に関する制限事項

root_path は**＜データパス＞**を絶対パスで指定し、その両端をダブルクォーテーション（””）で囲ってください。

4. 「**＜データパス＞**/data_conf_ssi.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、**[train_data_q]**、**[train_data_t]**、**[train_label]**、**[validate_data_q]**、**[validate_data_t]**、**[validate_label]**、**[predict_data_q]**、

[predict_data_t]、[predict_label]設定項目にファイル名を設定します。**※注 4**

```
“train_data_q” : "＜ファイル名＞",
“train_data_t” : "＜ファイル名＞",
“train_label” : "＜ファイル名＞",

“validate_data_q” : "＜ファイル名＞",
“validate_data_t” : "＜ファイル名＞",
“validate_label” : "＜ファイル名＞",

“predict_data_q” : "＜ファイル名＞",
“predict_data_t” : "＜ファイル名＞",
“predict_label” : "＜ファイル名＞",
```

⚠注 4: [validate_data_q] 、[validate_data_t] 、[validate_label]設定項目について

[validate_data_q]、[validate_data_t]、[validate_label]設定項目に値を設定した場合のみ、学習コマンドのモデル作成時に検証データでの誤差の計算を行います。
[validate_data_q]、[validate_data_t]、[validate_label]設定項目を設定しない場合は、学習コマンドのモデル作成時に検証データでの誤差の計算を行わずに、予測モデルの作成を行います。

5. 「**＜データパス＞**/train_data_q_idd.json」にあるデータ加工設定ファイル（idd_q.json）を、viなどのテキストエディタで開き、分析ケース設定ファイル（data.json）の**[train_data_q]**設定項目、**[validate_data_q]**設定項目と**[predict_data_q]**設定項目で指定した属性データの属性情報を設定します。

6. 「**＜データパス＞**/train_data_t_idd.json」にあるデータ加工設定ファイル（idd_t.json）を、viなどのテキストエディタで開き、分析ケース設定ファイル（data.json）の**[train_data_t]**設定項目、**[validate_data_t]**設定項目と**[predict_data_t]**設定項目で指定した属性データの属性情報を設定します。

7. 属性データを加工します。

コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/convert.rpd␣cls␣＜データパス＞/data_conf_ssi.json⏎
```

コマンド実行に成功すると、**＜データパス＞**直下に「shaped_data」ディレクトリが自動的に生成され、その中に次の 8 つのファイルが格納されます。

- shaped_data_train_q.csv
- shaped_data_train_t.csv
- shaped_data_validate_q.csv
- shaped_data_validate_t.csv
- shaped_data_predict_q.csv
- shaped_data_predict_t.csv
- train_data_q_idd_conv.json
- train_data_t_idd_conv.json

また、「**＜データパス＞**」ディレクトリに前処理済み分析ケース設定ファイル data_conf_ssi_new.json が作成されます。**※注 5**

⚠注 5: data_conf_ssi_new.json について

data_conf_ssi_new.json は convert コマンドで新しく作成されたデータ加工設定ファイルを参照しているため、学習コマンド、予測コマンドでは data_conf_ssi_new.json を設定ファイルとして使用してください。

[convert コマンド実行前]
data_conf_ssi.json
“idd_q” : “train_data_q_idd.json”
“idd_t” : “train_data_t_idd.json”

[convert コマンド実行後]
data_conf_ssi_new.json
“idd_q” : “shaped_data/train_data_q_idd_conv.json”
“idd_t” : “shaped_data/train_data_t_idd_conv.json”

以上で、データ加工は完了です。

3.4.2 学習

特徴ベクトルと正解ラベルを学習し、予測モデルを生成する作業です。本製品が提供する train

コマンドを利用して作業します。本作業は、3.4.1 節のデータ加工を正常に完了していることが前提となります。

1. 本製品をセットアップしたマシンに標準ユーザでログインします。

2. 特徴ベクトルと正解ラベルの関係性を学習します。
   コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/train_ssi.rpd␣cls␣＜データパス＞/data_conf_ssi_new.json⏎
```

   コマンド実行に成功すると、「train」ディレクトリに次の 2 つのファイルが格納されます。
   - model_cls.bin
   - train_log.log

   また、テキスト列を含むデータの場合、テキスト列ごとに辞書ファイルが 2 つの形式で出力されます。
   - word_dic_**.pkl
   - word_dic_**.json

以上で、学習は完了です。

3.4.3 バッチ予測

学習していない未知の特徴ベクトルに予測モデルを適用し、マッチングスコア（予測値）を生成し、ファイル出力する作業です。本製品が提供する predict コマンドを利用して作業します。本作業は、3.4.2 節の学習を正常に完了していることが前提となります。

1. 本製品をセットアップしたマシンに標準ユーザでログインします。

2. 学習していない未知の特徴ベクトルをバッチ予測します。
   コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/predict_ssi.rpd␣cls␣＜データパス＞/data_conf_ssi_new.json⏎
```

   コマンド実行に成功すると、「predict」ディレクトリに次の 1 つのファイルが格納されます。
   - result_cls.csv

8. 未知の特徴ベクトルに対する予測結果を確認します。
「result_cls.csv」を vi などのテキストエディタで開き、未知の特徴ベクトルに対するマッチングスコア（予測値）を確認します。「result_cls.csv」のデータ形式に関する説明は、4.3.1 節を参照してください。

以上で、バッチ予測は完了です。

3.4.4 学習誤差評価（セルフ・バリデーション）

3.4.2 節で作成した予測モデルの予測精度を定量評価する作業です。予測モデルを作成する際に使用した学習データをそのまま予測データとして使用して予測値（マッチングスコア）を算出し、得られた予測結果を正解ラベルと突合することにより、作成した予測モデルが学習データをどの程度学習できているのかを定量的に測定することができます。本製品が提供する convert コマンド、train コマンド、predict コマンドを組合せて作業します。

1. 本製品をセットアップしたマシンに標準ユーザでログインします。

2. 学習誤差評価に使用する学習データを準備します。
サンプルデータは、「**＜インストールパス＞**/template/matching/project_sample/ssi」にあらかじめ用意してありますので、任意のディレクトリにコピーしてください。以降では、このディレクトリを**＜データパス＞**と記載します。サンプルデータは、既定では「/opt/nec/pyrapid/template/matching/project_sample/ssi/」に格納されています。

3. 「**＜データパス＞**/data_conf_ssi.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、**[root_path]**設定項目に**＜データパス＞**を設定します。**※注 6**

```
“root_path” : "＜データパス＞",
```

⚠注 6: root_path の設定値に関する制限事項

root_path は**＜データパス＞**を絶対パスで指定し、その両端をダブルクォーテーション（””）で囲ってください。

4. 「**＜データパス＞**/data_conf_ssi.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、予測データに学習データと同じファイルを設定します。

```
“train_data_q” : "train_data_q.csv",

“train_data_t” : "train_data_t.csv",

“train_label” : "train_label.csv",
```

```
“predict_data_q” : "train_data_q.csv",
“predict_data_t” : "train_data_t.csv",
“predict_label” : "train_label.csv",
```

5. 「**＜データパス＞**/train_data_q_idd.json」にあるデータ加工設定ファイル（idd_q.json）を、vi などのテキストエディタで開き、分析ケース設定ファイル（data.json）の**[train_data_q]**設定項目と**[predict_data_q]**設定項目で指定した属性データの属性情報を設定します。

6. 「**＜データパス＞**/train_data_t_idd.json」にあるデータ加工設定ファイル（idd_t.json）を、vi などのテキストエディタで開き、分析ケース設定ファイル（data.json）の**[train_data_t]**設定項目と**[predict_data_t]**設定項目で指定した属性データの属性情報を設定します。

7. 学習データを加工します。
   コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/convert.rpd␣cls␣＜データパス＞/data_conf_ssi.json⏎
```

コマンド実行に成功すると、**＜データパス＞**直下に「shaped_data」ディレクトリが自動的に生成され、その中に次の 8 つのファイルが格納されます。

- shaped_data_train_q.csv
- shaped_data_train_t.csv
- shaped_data_validate_q.csv
- shaped_data_validate_t.csv
- shaped_data_predict_q.csv
- shaped_data_predict_t.csv
- train_data_q_idd_conv.json
- train_data_t_idd_conv.json

また、「**＜データパス＞**」ディレクトリに前処理済み分析ケース設定ファイル data_conf_ssi_new.json が作成されます。**※注 7**

⚠注 7: data_conf_ssi_new.json について

data_conf_ssi_new.json は convert コマンドで新しく作成されたデータ加工設定ファイルを参照しているため、学習コマンド、予測コマンドでは data_conf_ssi_new.json を設定ファイルとして使用してください。

[convert コマンド実行前]

data_conf_ssi.json

“idd_q” : “train_data_q_idd.json”

“idd_t” : “train_data_t_idd.json”

[convert コマンド実行後]

data_conf_ssi_new.json

“idd_q” : “shaped_data/train_data_q_idd_conv.json”

“idd_t” : “shaped_data/train_data_t_idd_conv.json”

8. 学習データを学習します。

コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/train_ssi.rpd␣cls␣＜データパス＞/data_conf_ssi_new.json⏎
```

コマンド実行に成功すると、「train」ディレクトリに次の 2 つのファイルが格納されます。

- model_cls.bin
- train_log.log

また、テキスト列ごとに辞書ファイルが 2 つの形式で出力されます。

- word_dic_**.pkl
- word_dic_**.json

9. 学習データを予測します。

コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/predict_ssi.rpd␣cls␣＜データパス＞/data_conf_ssi_new.json⏎
```

コマンド実行に成功すると、「predict」ディレクトリに次の 1 つのファイルが格納されます。

- result_cls.csv

10. 学習データの予測結果を確認します。

「result_cls.csv」を vi などのテキストエディタで開き、サンプルデータに対する予測結果を確認します。「result_csv.csv」のデータ形式に関する説明は、4.3.1 節を参照してください。

3.4.5 汎化誤差評価（クロス・バリデーション）

3.4.2 節で作成した予測モデルの予測精度を定量評価する作業です。学習データを k 分割し、そのうち（k-1）分割を学習データ、残り 1 分割を予測データとして 3.4.1 ～3.4.3 節の手順を行い、予測データに対する予測値（マッチングスコア）を算出します。以降、k 分割のうち、予測データに使用する 1 分割を入れ替えながら、3.4.1 ～3.4.3 節の手順を k 回繰り返します。得られた k 回分の予測値（マッチングスコア）を正解ラベルと突合することにより、作成した予測モデルが未知の予測データをどの程度予測できているのかを定量的に測定することができます。本製品が提供する convert コマンド、train コマンド、predict コマンドを組合せて作業します。

以降では、学習データを 4 分割（k=4）する場合を例に手順を説明します。

1. 本製品をセットアップしたマシンに標準ユーザでログインします。

2. 汎化誤差評価に使用する学習データを準備します。
   サンプルは、「**＜インストールパス＞**/template/matching/project_sample/ssi/cv」にあらかじめ用意してありますので、任意のディレクトリにコピーしてください。以降では、このディレクトリを**＜データパス＞**と記載します。サンプルデータは、既定では「/opt/nec/pyrapid/template/matching/project_sample/ssi/cv」に格納されています。

   サンプルでは、**＜データパス＞**のデータを 4 分割し、3/4 を学習データ、1/4 を予測データとしています。

   ＜データパス＞/split_data
   ├── 1
   │　　├── train_data_q.csv
   │　　├── train_data_t.csv
   │　　├── train_label.csv
   │　　├── predict_data_q.csv
   │　　├── predict_data_t.csv
   │　　└── predict_label.csv
   ├── 2
   │　　├── train_data_q.csv
   │　　├── train_data_t.csv
   │　　├── train_label.csv
   │　　├── predict_data_q.csv
   │　　├── predict_data_t.csv
   │　　└── predict_label.csv
   ├── 3
   │　　├── train_data_q.csv
   │　　├── train_data_t.csv
   │　　├── train_label.csv

```
        ├── predict_data_q.csv
        ├── predict_data_t.csv
        └── predict_label.csv
    └── 4
        ├── train_data_q.csv
        ├── train_data_t.csv
        ├── train_label.csv
        ├── predict_data_q.csv
        ├── predict_data_t.csv
        └── predict_label.csv
```

以降、上記「**<データパス>**/split_data/1」ディレクトリを使用する場合を例に手順を説明しますが、同様の手順を「2」ディレクトリ、「3」ディレクトリ、「4」ディレクトリのそれぞれに対して行ってください。

3. 「**＜データパス＞**/train_data_q_idd.json」にあるデータ加工設定ファイル（idd_q.json）を、vi などのテキストエディタで開き、分析ケース設定ファイル（data.json）の**[train_data_q]**設定項目と**[predict_data_q]**設定項目で指定した属性データの属性情報を設定します。

4. 「**＜データパス＞**/train_data_t_idd.json」にあるデータ加工設定ファイル（idd_t.json）を、vi などのテキストエディタで開き、分析ケース設定ファイル（data.json）の**[train_data_t]**設定項目と**[predict_data_t]**設定項目で指定した属性データの属性情報を設定します。

5. 「**＜データパス＞**/data_conf_ssi.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、**[root_path]**設定項目に「**<データパス>**/split_data/1」ディレクトリの絶対パスを再設定します。**※注 8**

```
“root_path” : "＜データパス＞/split_data/1/",
```

⚠注 8: root_path の設定値に関する制限事項

root_path は**＜データパス＞**を絶対パスで指定し、その両端をダブルクォーテーション（””）で囲ってください。

6. 「**＜データパス＞**/data_conf_ssi.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、**[train_data_q]**、**[train_data_t]**、**[train_label]**、**[predict_data_q]**、**[predict_data_t]**、**[predict_label]**の各設定項目を再設定します。

```
“train_data_q” : "train_data_q.csv",
```

```
“train_data_t” : "train_data_t.csv",

“train_label” : "train_label.csv",



“predict_data_q” : "predict_data_q.csv",

“predict_data_t” : "predict_data_t.csv",

“predict_label” : "predict_label.csv",
```

7. 「**＜データパス＞**/train_data_q_idd.json」と「**＜データパス＞**/train_data_t_idd.json」を「**<データパス>**/split_data/1」ディレクトリにコピーします。

8. 学習データを加工します。
   コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/convert.rpd␣cls␣＜データパス＞/data_conf_ssi.json⏎
```

   コマンド実行に成功すると、「**<データパス>**/split_data/1」ディレクトリ直下に「shaped_data」ディレクトリが自動的に生成され、その中に次の 8 つのファイルが格納されます。

   - shaped_data_train_q.csv
   - shaped_data_train_t.csv
   - shaped_data_validate_q.csv
   - shaped_data_validate_t.csv
   - shaped_data_predict_q.csv
   - shaped_data_predict_t.csv
   - train_data_q_idd_conv.json
   - train_data_t_idd_conv.json

   また、「**＜データパス＞**」ディレクトリに前処理済み分析ケース設定ファイル data_conf_ssi_new.json が作成されます。

9. 学習データを学習します。
   コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/train_ssi.rpd␣cls␣＜データパス＞/data_conf_ssi_new.json⏎
```

コマンド実行に成功すると、「**<データパス>**/split_data/1/train」ディレクトリに次の 2 つのファイルが格納されます。

- model_cls.bin
- train_eval.log

10. 予測データを予測します。
コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/predict_ssi.rpd␣cls␣＜データパス＞/data_conf_ssi_new.json⏎
```

コマンド実行に成功すると、「**<データパス>**/split_data/1/predict」ディレクトリに次の 1 つのファイルが格納されます。

- result_cls.csv

11. 学習データの予測結果を確認します。
「result_cls.csv」を vi などのテキストエディタで開き、サンプルデータに対する予測結果を確認します。「result_cls.csv」のデータ形式に関する説明は、4.3.1 節を参照してください。

3.5 フィルタリング機能（sse）

フィルタリングに関する NEC 独自の機械学習アルゴリズムである SSE を活用したフィルタリング機能です。求職者[q]、この求職者のマッチ度[l; 1:マッチする、0:マッチしない]のデータ組（q, l）を入力データとする特徴があります。

この特徴は、入力データに対して間隔尺度（interval scale）が定義できる場合に有効です。例えば、求職者 A は自社とマッチングしない（0）、求職者 B は自社とマッチングする（1）というように、求職者[q]に対して、マッチングする（1）、マッチングしない（0）を定義できる場合に有効です。このケースでは、求職者 A を[q-]、求職者 B を[q+]として、データ（q-）に対してマッチングスコア 0、データ（q+）に対してマッチングスコア 1 を算出するよう、機械学習アルゴリズムにより予測モデルが調整されます。

3.5.1 データ加工

属性データから加工データを生成する作業です。本製品が提供する convert コマンドを利用して作業します。

1. 本製品をセットアップしたマシンに標準ユーザでログインします。

2. データ加工する属性データを準備します。
   属性データは、「**＜インストールパス＞**/template/matching/project_sample/sse」にあらかじめ用意してありますので、任意のディレクトリにコピーしてください。以降では、このディレクトリを**＜データパス＞**と記載します。サンプルデータは、既定では「/opt/nec/pyrapid/template/matching/project_sample/sse/」に格納されています。

3. 「**＜データパス＞**/data_conf_sse.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、**[root_path]**設定項目に**＜データパス＞**を設定します。**※注 9**

```
“root_path”：”＜データパス＞",
```

⚠注 9: root_path の設定値に関する制限事項

root_path は**＜データパス＞**を絶対パスで指定し、その両端をダブルクォーテーション（””）で囲ってください。

4. 「**＜データパス＞**/data_conf_ssi.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、**[train_data_q]**、**[train_label]**、**[validate_data_q]**、**[validate_label]**、**[predict_data_q]**、**[predict_label]設定項目**にファイル名を設定しま

す。※注 10

```
“train_data_q” : "＜ファイル名＞",
“train_label” : "＜ファイル名＞",

“validate_data_q” : "＜ファイル名＞",
“validate_label” : "＜ファイル名＞",

“predict_data_q” : "＜ファイル名＞",
“predict_label” : "＜ファイル名＞",
```

⚠注 10: [validate_data_q] 、[validate_label]設定項目について

[validate_data_q]、**[validate_label]**設定項目に値を設定した場合のみ、学習コマンドのモデル作成時に検証データでの誤差の計算を行います。
[validate_data_q]、**[validate_label]**設定項目を設定しない場合は、学習コマンドのモデル作成時に検証データでの誤差の計算を行わずに、予測モデルの作成を行います。

5. 「**＜データパス＞**/train_data_idd.json」にあるデータ加工設定ファイル（idd.json）を、vi などのテキストエディタで開き、分析ケース設定ファイル（data.json）の**[train_data_q]**設定項目、**[validate_data_q]**設定項目、**[predict_data_q]**設定項目で指定した属性データの属性情報を設定します。

6. 属性データを加工します。
コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/convert.rpd␣cls␣＜データパス＞/data_conf_sse.json⏎
```

コマンド実行に成功すると、**＜データパス＞**直下に「shaped_data」ディレクトリが自動的に生成され、その中に次の 4 つのファイルが格納されます。

- shaped_data_train_q.csv
- shaped_data_validate_q.csv
- shaped_data_predict_q.csv
- train_data_idd_conv.json

また、「＜データパス＞」ディレクトリに前処理済み分析ケース設定ファイル data_conf_sse_new.json が作成されます。※注 11

⚠注 11: data_conf_sse_new.json について

data_conf_sse_new.json は convert コマンドで新しく作成されたデータ加工設定ファイルを参照しているため、学習コマンド、予測コマンドでは data_conf_sse_new.json を設定ファイルとして使用してください。

[convert コマンド実行前]
data_conf_sse.json
“idd_q” : “train_data_q_idd.json”

[convert コマンド実行後]
data_conf_sse_new.json
“idd_q” : “shaped_data/train_data_q_idd_conv.json”

以上で、データ加工は完了です。

3.5.2 学習

特徴ベクトルと正解ラベルを学習し、予測モデルを生成する作業です。本製品が提供する train コマンドを利用して作業します。本作業は、3.5.1 節のデータ加工を正常に完了していることが前提となります。

1. 本製品をセットアップしたマシンに標準ユーザでログインします。

2. 特徴ベクトルと正解ラベルの関係性を学習します。
コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/train_sse.rpd␣cls␣＜データパス＞/data_conf_sse_new.json⏎
```

コマンド実行に成功すると、「train」ディレクトリに次の 2 つのファイルが格納されます。

- model_cls.bin
- train_log.log

以上で、学習は完了です。

3.5.3 バッチ予測

学習していない未知の特徴ベクトルに予測モデルを適用し、マッチングスコア（予測値）を生成し、ファイル出力する作業です。本製品が提供する predict コマンドを利用して作業します。本作業は、3.5.2 節の学習を正常に完了していることが前提となります。

1. 本製品をセットアップしたマシンに標準ユーザでログインします。

2. 学習していない未知の特徴ベクトルをバッチ予測します。
   コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/predict_sse.rpd␣cls␣＜データパス＞/data_conf_sse_new.json⏎
```

   コマンド実行に成功すると、「predict」ディレクトリに次の 1 つのファイルが格納されます。
   - result_cls.csv

3. 未知の特徴ベクトルに対する予測結果を確認します。
「result_cls.csv」を vi などのテキストエディタで開き、未知の特徴ベクトルに対するマッチングスコア（予測値）を確認します。「result_cls.csv」のデータ形式に関する説明は、4.3.1 節を参照してください。

以上で、バッチ予測は完了です。

3.5.4 学習誤差評価（セルフ・バリデーション）

3.5.2 節で作成した予測モデルの予測精度を定量評価する作業です。予測モデルを作成する際に使用した学習データをそのまま予測データとして使用して予測値（フィルタリングスコア）を算出し、得られた予測結果を正解ラベルと突合することにより、作成した予測モデルが学習データをどの程度学習できているのかを定量的に測定することができます。本製品が提供する convert コマンド、train コマンド、predict コマンドを組合せて作業します。

1. 本製品をセットアップしたマシンに標準ユーザでログインします。

2. 学習誤差評価に使用する学習データを準備します。
サンプルデータは、「**＜インストールパス＞**/template/matching/project_sample/sse/」にあらかじめ用意してありますので、任意のディレクトリにコピーしてください。以降では、このディレクトリを**＜データパス＞**と記載します。サンプルデータは、既定では「/opt/nec/pyrapid/template/matching/project_sample/sse/」に格納されています。

3. 「**＜データパス＞**/data_conf_sse.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、**[root_path]**設定項目に**＜データパス＞**を設定します。**※注12**

```
“root_path” : "＜データパス＞",
```

⚠注 12: root_path の設定値に関する制限事項

root_path は**＜データパス＞**を絶対パスで指定し、その両端をダブルクォーテーション（””）で囲ってください。

4. 「**＜データパス＞**/data_conf_sse.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、予測データに学習データと同じファイルを設定します。

```
“train_data_q” : "train_data.csv",

“train_label” : "train_label.csv",



“predict_data_q” : "train_data.csv",

“predict_label” : "train_label.csv",
```

5. 「**＜データパス＞**/train_data_idd.json」にあるデータ加工設定ファイル（idd.json）を、vi などのテキストエディタで開き、分析ケース設定ファイル（data.json）の**[train_data_q]**設定項目、**[validate_data_q]**設定項目、**[predict_data_q]**設定項目で指定した属性データの属性情報を設定します。

6. 学習データを加工します。
コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/convert.rpd␣cls␣＜データパス＞/data_conf_sse.json⏎
```

コマンド実行に成功すると、**＜データパス＞**直下に「shaped_data」ディレクトリが自動的に生成され、その中に次の 4 つのファイルが格納されます。

- shaped_data_train.csv
- shaped_data_validate.csv
- shaped_data_predict.csv
- train_data_idd_conv.json

また、「**＜データパス＞**」ディレクトリに前処理済み分析ケース設定ファイル data_conf_sse_new.json が作成されます。

7. 学習データを学習します。
コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/train_sse.rpd␣cls␣＜データパス＞/data_conf_sse_new.json⏎
```

コマンド実行に成功すると、「train」ディレクトリに次の 2 つのファイルが格納されます。

- model_cls.bin
- train_log.log

8. 学習データを予測します。
コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/predict_sse.rp␣cls␣＜データパス＞/data_conf_sse_new.json⏎
```

コマンド実行に成功すると、「predict」ディレクトリに次の 1 つのファイルが格納されます。

- result_cls.csv

9. 学習データの予測結果を確認します。
「result_cls.csv」を vi などのテキストエディタで開き、サンプルデータに対する予測結果を確認します。「result_cls.csv」のデータ形式に関する説明は、4.3.1 節を参照してください。

3.5.5 汎化誤差評価（クロス・バリデーション）

3.5.2 節で作成した予測モデルの予測精度を定量評価する作業です。学習データを k 分割し、

そのうち(k-1)分割を学習データ、残り 1 分割を予測データとして 3.5.1 ～3.5.3 節の手順を行い、予測データに対する予測値(マッチングスコア)を算出します。以降、k 分割のうち、予測データに使用する 1 分割を入れ替えながら、3.5.1 ～3.5.3 節の手順を k 回繰り返します。得られた k 回分の予測値(マッチングスコア)を正解ラベルと突合することにより、作成した予測モデルが未知の予測データをどの程度予測できているのかを定量的に測定することができます。本製品が提供する convert コマンド、train コマンド、predict コマンドを組合せて作業します。

以降では、学習データを 4 分割(k=4)する場合を例に手順を説明します。

1. 本製品をセットアップしたマシンに標準ユーザでログインします。

2. 汎化誤差評価に使用する学習データを準備します。
   サンプルデータは、「**＜インストールパス＞**/template/matching/project_sample/sse/cv」にあらかじめ用意してありますので、任意のディレクトリにコピーしてください。以降では、このディレクトリを**＜データパス＞**と記載します。サンプルデータは、既定では「/opt/nec/pyrapid/template/matching/project_sample/sse/cv」に格納されています。

   サンプルでは、**＜データパス＞**のデータを 4 分割し、3/4 を学習データ、1/4 を予測データとしています。

   ＜データパス＞/split_data
   ├── 1
   │　　├── train_data.csv
   │　　├── train_label.csv
   │　　├── predict_data.csv
   │　　└── predict_label.csv
   ├── 2
   │　　├── train_data.csv
   │　　├── train_label.csv
   │　　├── predict_data.csv
   │　　└── predict_label.csv
   ├── 3
   │　　├── train_data.csv
   │　　├── train_label.csv
   │　　├── predict_data.csv
   │　　└── predict_label.csv
   └── 4
   　　├── train_data.csv
   　　├── train_label.csv
   　　├── predict_data.csv
   　　└── predict_label.csv

以降、上記「**<データパス>**/split_data/1」ディレクトリを使用する場合を例に手順を説明しますが、同様の手順を「2」ディレクトリ、「3」ディレクトリ、「4」ディレクトリのそれぞれに対して行ってください。

3. 「**＜データパス＞**/train_data_idd.json」にあるデータ加工設定ファイル（idd.json）を、vi などのテキストエディタで開き、分析ケース設定ファイル（data.json）の**[train_data_q]**設定項目、**[validate_data_q]**設定項目、**[predict_data_q]**設定項目で指定した属性データの属性情報を設定します。

4. 「**＜データパス＞**/data_conf_sse.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、**[root_path]**設定項目に「**<データパス>**/split_data/1」ディレクトリの絶対パスを再設定します。**※注 13**

```
“root_path” : "＜データパス＞/split_data/1/",
```

⚠注 13: root_path の設定値に関する制限事項

root_path は**＜データパス＞**を絶対パスで指定し、その両端をダブルクォーテーション（”“）で囲ってください。

5. 「**＜データパス＞**/data_conf_sse.json」にある分析ケース設定ファイル（data.json）を、vi などのテキストエディタで開き、**[train_data_q]**、**[train_label]**、**[predict_data_q]**、**[predict_label]**の各設定項目を再設定します。

```
“train_data_q” : "train_data.csv",
“train_label” : "train_label.csv",


“predict_data_q” : "predict_data.csv",
“predict_label” : "predict_label.csv",
```

6. 「**＜データパス＞**/train_data_idd.json」を「**<データパス>**/split_data/1」ディレクトリにコピーします。

7. 学習データを加工します。
コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/convert.rpd␣cls␣＜データパス＞/data_conf_sse.json⏎
```

コマンド実行に成功すると、「**<データパス>**/split_data/1」ディレクトリ直下に「shaped_data」ディレクトリが自動的に生成され、その中に次の 4 つのファイルが格納されます。

- shaped_data_train.csv
- shaped_data_validate.csv
- shaped_data_predict.csv
- train_data_idd_conv.json

また、「**＜データパス＞**」ディレクトリに前処理済み分析ケース設定ファイル data_conf_sse_new.json が作成されます。

8. 学習データを学習します。
   コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/train_sse.rpd␣cls␣＜データパス＞/data_conf_sse_new.json⏎
```

コマンド実行に成功すると、「**<データパス>**/split_data/1/train」ディレクトリに次の 2 つのファイルが格納されます。

- model_cls.bin
- train_log.log

9. 予測データを予測します。
   コンソールで、次のコマンドを実行します。

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/predict_sse.rpd␣cls␣＜データパス＞/data_conf_sse_new.json⏎
```

コマンド実行に成功すると、「**<データパス>**/split_data/1/predict」ディレクトリに次の 1 つのファイルが格納されます。

- result_cls.csv

10. 学習データの予測結果を確認します。

「result_cls.csv」を vi などのテキストエディタで開き、サンプルデータに対する予測結果を確認します。「result_cls.csv」のデータ形式に関する説明は、4.3.1 節を参照してください。

3.6 予測コマンドをスキップする手順

これまでは、データ加工、学習、予測の一連の分析作業を続けて実行する手順を説明しましたが、実運用では、予測モデルを更新するため、予測コマンドをスキップし、データ加工、学習までの一連の分析作業を続けて行いたい場合があります。本節では、この場合の実行手順を説明します。

マッチング機能（ssi）

1. 分析ケース設定ファイルの**[predict_data_q]**、**[predict_data_t]**、**[predict_label]**の各設定項目の先頭に#を付け、“#predict_data_q”、 “#predict_data_t”、 “#predict_label”と記載します。

```
“#predict_data_q” : "predict_data_q.csv",

“#predict_data_t” : "predict_data_t.csv",

“#predict_label” : “predict_label”,
```

2. データ加工の手順（3.4.1 節）を実行します。
3. 学習の手順（3.4.2 節）を実行します。

フィルタリング機能（sse）

1. 分析ケース設定ファイルの**[predict_data_q]**、**[predict_label]**の各設定項目の先頭に#を付け、“#predict_data_q”、 “#predict_label”と記載します。

```
“#predict_data_q” : "predict_data_q.csv",

“#predict_label” : "predict_label",
```

2. データ加工の手順（3.5.1 節）を実行します。
3. 学習の手順（3.5.2 節）を実行します。

3.7 学習コマンドをスキップする手順

これまでは、データ加工、学習、予測の一連の分析作業を続けて実行する手順を説明しましたが、実運用では、以前に作成した予測モデルを利用するため、学習コマンドをスキップし、データ加工、予測までの一連の分析作業を続けて行いたい場合があります。本節では、この場合の実行手順を説明します。

マッチング機能（ssi）

1. 分析ケース設定ファイルの**[train_data_q]**、**[train_data_t]**、**[train_label]**の各設定項目の先頭に#を付け、“#train_data_q”、 “#train_data_t”、 “#train_label”と記載します。

```
“#train_data_q” : "train_data_q.csv",

“#train_data_t” : "train_data_t.csv",

“#train_label” : “train_label.csv”,
```

2. データ加工の手順（3.4.1 節）を実行します。
3. バッチ予測の手順（3.4.3 節）を実行します。

フィルタリング機能（sse）

1. 分析ケース設定ファイルの**[train_data_q]**、**[train_label]** の各設定項目の先頭に#を付け、“#train_data_q”、“#train_label”と記載します。

```
“#train_data_q” : “train_data_q.csv”,

“#train_label” : “train_label.csv”,
```

2. データ加工の手順（3.5.1 節）を実行します。
3. バッチ予測の手順（3.5.3 節）を実行します。

第4章 コマンド

本章では、本製品が提供するコマンド群について説明します。

4.1 コマンド一覧

本製品が提供するコマンド群を表 4-1 に記載します。

表 4-1 本製品が提供するコマンド群

コマンド	ファイル名	説明
データ加工コマンド	convert.rpd	属性データから加工データを生成するコマンド
学習コマンド	train_ssi.rpd	特徴ベクトルと正解ラベルの関係性を学習して予測モデルを生成するコマンド(マッチング)
	train_sse.rpd	特徴ベクトルと正解ラベルの関係性を学習して予測モデルを生成するコマンド(フィルタリング)
バッチ予測コマンド	predict_ssi.rpd	特徴ベクトルに予測モデルを適用して予測結果レポートを生成しファイル出力するバッチコマンド(マッチング)
	predict_sse.rpd	特徴ベクトルに予測モデルを適用して予測結果レポートを生成しファイル出力するバッチコマンド(フィルタリング)
IDD 雛形生成コマンド	gen_idd.rpd	属性データからデータ加工設定ファイル（IDD）の雛形を生成するコマンド

4.1.1 データ加工コマンド（convert.rpd）

属性データから加工データを生成します。

書式

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/convert.rpd␣{MODE}␣{DATA.JSON}
```

引数

引数名	説明
MODE	分析する問題を cls または reg から選択します。 cls：分類問題 reg：回帰問題
DATA.JSON	分析ケース設定ファイル（data.json）の絶対パスを指定します。分析ケース設定ファイルの詳細は、5.1.2 節を参照してください。分析ケース設定ファイルには、任意のファイル名を設定できます。

オプション

なし

実行結果

実行結果	説明
正常終了	加工データが生成されます。 【マッチング機能（ssi）の場合】 引数の**[DATA.JSON]**で指定した**＜データパス＞**直下に「shaped_data」ディレクトリが自動的に生成され、その中に次の 8 つのファイルが格納されます。 1. shaped_data_train_q.csv 2. shaped_data_train_t.csv 3. shaped_data_validate_q.csv 4. shaped_data_validate_t.csv 5. shaped_data_predict_q.csv 6. shaped_data_predict_t.csv 7. train_data_q_idd_conv.json 8. train_data_t_idd_conv.json また、**＜データパス＞**ディレクトリに前処理済み分析ケース設定ファイル**[DATA_new.JSON]**が作成されます。※注 14 【フィルタリング機能（sse）の場合】 引数の**[DATA.JSON]**で指定した**＜データパス＞**直下に「shaped_data」ディレクトリが自動的に生成され、その中に次の 4 つのファイルが格納されます。 1. shaped_data_train.csv 2. shaped_data_validate.csv 3. shaped_data_predict.csv

	4. train_data_q_idd_conv.csv また、**＜データパス＞**ディレクトリに前処理済み分析ケース設定ファイル**[DATA_new.JSON]**が作成されます。※注 14
異常終了	実行ログにエラーメッセージが出力されます。エラーメッセージの詳細は、第 7 章 を参照してください。

⚠注 14: [DATA_new.JSON]について

[DATA_new.JSON]は convert コマンドで新しく作成されたデータ加工設定ファイルを参照しているため、学習コマンド、予測コマンドでは[DATA_new.JSON]を設定ファイルとして使用してください。

・マッチングの場合

[convert コマンド実行前]

[DATA.JSON]

“idd_q” : “train_data_q_idd.json”

“idd_t” : “train_data_t_idd.json”

[convert コマンド実行後]

[DATA_new.JSON]

“idd_q” : “shaped_data/train_data_q_idd_conv.json”

“idd_t” : “shaped_data/train_data_t_idd_conv.json”

・フィルタリングの場合

[convert コマンド実行前]

[DATA.JSON]

“idd_q” : “train_data_q_idd.json”

[convert コマンド実行後]

[DATA_new.JSON]

“idd_q” : “shaped_data/train_data_q_idd_conv.json”

4.1.2 学習コマンド（train_ssi.rpd）

特徴ベクトルと正解ラベルの関係性を学習して予測モデルを生成します。

書式

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/train_ssi.rpd␣{MODE}␣{DATA.JSON}␣
[HPARAM.JSON]
```

引数

引数名	説明
MODE	分析する問題を cls または reg から選択します。 cls：分類問題 reg：回帰問題
DATA.JSON	分析ケース設定ファイル（data.json）の絶対パスを指定します。分析ケース設定ファイルの詳細は、5.1.2 節を参照してください。分析ケース設定ファイルには、任意のファイル名を設定できます。
HPARAM.JSON	パラメータ設定ファイル（hparam.json）の絶対パスを指定します。パラメータ設定ファイルの詳細は、5.1.4 節を参照してください。パラメータ設定ファイルには、任意のファイル名を設定できます。**※注 15**

⚠注 15: HPARAM.JSON の指定について

HPARAM.JSON は省略可能です。この場合、「<インストールパス＞/template/matching/etc」配下の機械学習アルゴリズム毎に用意されたディレクトリ内にあるパラメータ設定ファイル（hparam.json）が既定で使用されます。パラメータ設定ファイルの詳細は、5.1.4 節を参照してください。

オプション

なし

実行結果

実行結果	説明
正常終了	予測モデルが生成されます。 引数の**[DATA.JSON]**で指定した**＜データパス＞**直下の「train」ディレクトリに次の 2 つのファイルが格納されます。 1. 【分類問題の場合】 model_cls.bin 【回帰問題の場合】 model_reg.bin 2. train_log.log
異常終了	実行ログにエラーメッセージが出力されます。エラーメッセージの詳細は、第 7 章 を

	参照してください。

4.1.3 学習コマンド（train_sse.rpd）

特徴ベクトルと正解ラベルの関係性を学習して予測モデルを生成します。

書式

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/train_sse.rpd␣{MODE}␣{DATA.JSON}␣
[HPARAM.JSON]
```

引数

引数名	説明
MODE	分析する問題を cls または reg から選択します。 cls：分類問題 reg：回帰問題
DATA.JSON	分析ケース設定ファイル（data.json）の絶対パスを指定します。分析ケース設定ファイルの詳細は、5.1.2 節を参照してください。分析ケース設定ファイルには、任意のファイル名を設定できます。
HPARAM.JSON	パラメータ設定ファイル（hparam.json）の絶対パスを指定します。パラメータ設定ファイルの詳細は、5.1.4 節を参照してください。パラメータ設定ファイルには、任意のファイル名を設定できます。**※注 16**

⚠注 16: HPARAM.JSON の指定について

HPARAM.JSON は省略可能です。この場合、「<インストールパス＞/template/matching/etc」配下の機械学習アルゴリズム毎に用意されたディレクトリ内にあるパラメータ設定ファイル（hparam.json）が既定で使用されます。パラメータ設定ファイルの詳細は、5.1.4 節を参照してください。

オプション

なし

実行結果

実行結果	説明
正常終了	予測モデルが生成されます。 引数の**[DATA.JSON]**で指定した**＜データパス＞**直下の「train」ディレクトリに次の 2 つのファイルが格納されます。 1. 【分類問題の場合】 model_cls.bin 【回帰問題の場合】 model_reg.bin 2. train_log.log
異常終了	実行ログにエラーメッセージが出力されます。エラーメッセージの詳細は、第 7 章 を

	参照してください。

4.1.4 バッチ予測コマンド（predict_ssi.rpd）

特徴ベクトルに予測モデルを適用して予測結果レポートを生成しファイル出力します。

書式

```
$ pyrapid␣<インストールパス>/template/matching/bin/predict_ssi.rpd␣{MODE}␣{DATA.JSON}␣
[HPARAM.JSON]
```

引数

引数名	説明
MODE	分析する問題を cls または reg から選択します。 cls：分類問題 reg：回帰問題
DATA.JSON	分析ケース設定ファイル（data.json）の絶対パスを指定します。分析ケース設定ファイルの詳細は、5.1.2 節を参照してください。分析ケース設定ファイルには、任意のファイル名を設定できます。
HPARAM.JSON	**学習コマンド実行時と同じ**パラメータ設定ファイル（hparam.json）の絶対パスを指定します。パラメータ設定ファイルの詳細は、5.1.4 節を参照してください。パラメータ設定ファイルには、任意のファイル名を設定できます。**※注 17**

⚠注 17: HPARAM.JSON の指定について

HPARAM.JSON は省略可能です。この場合、「<インストールパス>/template/matching/etc」配下の機械学習アルゴリズム毎に用意されたディレクトリ内にあるパラメータ設定ファイル（hparam.json）が既定で使用されます。パラメータ設定ファイルの詳細は、5.1.4 節を参照してください。

オプション

なし

実行結果

実行結果	説明
正常終了	引数の**[DATA.JSON]**で指定した**<データパス>**直下の「predict」ディレクトリに予測結果レポート（result_cls.csv または result_reg.csv）が格納されます。予測結果レポートの詳細は、4.3.1 節を参照してください。
異常終了	実行ログにエラーメッセージが出力されます。エラーメッセージの詳細は、第 7 章 を参照してください。

4.1.5 バッチ予測コマンド（predict_sse.rpd）

特徴ベクトルに予測モデルを適用して予測結果レポートを生成しファイル出力します。

書式

```
$ pyrapid␣<インストールパス>/template/matching/bin/predict_sse.rpd␣{MODE}␣{DATA.JSON}␣
[HPARAM.JSON]
```

引数

引数名	説明
MODE	分析する問題を cls または reg から選択します。 cls：分類問題 reg：回帰問題
DATA.JSON	分析ケース設定ファイル（data.json）の絶対パスを指定します。分析ケース設定ファイルの詳細は、5.1.2 節を参照してください。分析ケース設定ファイルには、任意のファイル名を設定できます。
HPARAM.JSON	**学習コマンド実行時と同じ**パラメータ設定ファイル（hparam.json）の絶対パスを指定します。パラメータ設定ファイルの詳細は、5.1.4 節を参照してください。パラメータ設定ファイルには、任意のファイル名を設定できます。**※注 18**

⚠注 18: HPARAM.JSON の指定について

HPARAM.JSON は省略可能です。この場合、「<インストールパス＞/template/matching/etc」配下の機械学習アルゴリズム毎に用意されたディレクトリ内にあるパラメータ設定ファイル（hparam.json）が既定で使用されます。パラメータ設定ファイルの詳細は、5.1.4 節を参照してください。

オプション

なし

実行結果

実行結果	説明
正常終了	引数の**[DATA.JSON]**で指定した**＜データパス＞**直下の「predict」ディレクトリに予測結果レポート（result_cls.csv または result_reg.csv）が格納されます。予測結果レポートの詳細は、4.3.1 節を参照してください。
異常終了	実行ログにエラーメッセージが出力されます。エラーメッセージの詳細は、第 7 章 を参照してください。

4.1.6 IDD 雛形生成コマンド（gen_idd.rpd）

属性データからデータ加工設定ファイル（IDD）の雛形を生成します。

書式

```
$ pyrapid␣＜インストールパス＞/template/matching/bin/gen_idd.rpd␣{DATA_PATH}
```

引数

引数名	説明
DATA_PATH	属性データ（data.csv）の絶対パスを指定します。

オプション

なし

実行結果

実行結果	説明
正常終了	指定した属性データのデータ加工設定ファイル（IDD）の雛形が生成されます。 引数の**[DATA_PATH]**に指定したファイルと同じディレクトリに、idd.json が格納されます。
異常終了	実行ログにエラーメッセージが出力されます。エラーメッセージの詳細は、第 7 章 を参照してください。

4.2 入力ファイル

本製品の入力ファイルは、属性データと正解ラベルの 2 ファイルです。以降で、それぞれのファイル仕様を説明します。

4.2.1 属性データ（data.csv）

属性データは、分析対象に関する情報を格納したファイルです。マッチング機能（ssi）では、求職者に関する属性データ（data_q.csv）と、求人企業に関する属性データ（data_t.csv）の 2 つの入力ファイルが必要です。一方、フィルタリング機能（sse）では、求職者に関する属性データ（data_q.csv）の 1 つの入力ファイルが必要です。

属性データのファイル仕様を表 4-2 に記載します。また、属性データのサンプルを表 4-3 に記載します。

表 4-2 属性データのファイル仕様

項目	仕様
ファイルパス	分析ケース設定ファイル（data.json）で指定します。分析ケース設定ファイルの詳細は、5.1.2 節を参照してください。
データ形式	CSV のフラットファイルです。既定の区切り文字は半角カンマです。 ※区切り文字は、分析ケース設定ファイル（data.json）で変更できます。
文字エンコード	UTF-8（BOM なし）です。
改行コード	LF です。
1 行目	2 行目以降のデータの属性名が書かれたヘッダ行です。
2 行目以降	1 行目の各属性に対応する属性値が書かれたデータ行です。
データ行の ID	各データ行には、その行を特定するためのユニークな ID（プライマリ・キー）を設定してください。
使用可能文字	属性名は半角英数のみ使用できます。半角空白は使用できません。
属性名の重複	属性名は重複できません。ヘッダ行でユニークな名称を設定してください。
複数の属性値	同一属性に複数の属性値を設定する場合は、半角空白を区切り文字として、それらの属性値を並べてください。区切り文字は変更できません。
属性値の範囲	すべての属性には、1 つ以上の属性値を設定してください。属性値がない場合は、空文字を設定してください。
属性値の制約	**属性名・属性値の区切り文字（既定では半角カンマ）と改行コードは、属性名・属性値に含めないよう、事前に別の文字に置換しておく必要があります。 ※区切り文字は、分析ケース設定ファイル（data.json）で変更できます。**

表 4-3 属性データのサンプル

【求職者の属性データ例】

qid,tab1.col1,tab1.col2,tab2.col1,tab2.col2［LF］

q1,22,4.5,6 7,私の名前は日電太郎です。［LF］

q2,24,,10 11 12,私の名前は日電次郎です。［LF］

【求人企業の属性データ例】

tid,tab3.col1,tab3.col2,tab4.col1,tab4.col2［LF］

t1,22,弊社は日本電気です。,3 4 5,4.5［LF］

t2,24,弊社は NEC です。,,3.2［LF］

4.2.2 正解ラベル（label.csv）

正解ラベルは、求職者の属性データと求人企業の属性データの関係性（正解／不正解）に関する情報を格納するファイルです。マッチング機能（ssi）では、求職者と求人企業に対する関係性（正解／不正解）を設定します。一方、フィルタリング機能（sse）では、求職者に対する関係性（正解／不正解）を設定します。

正解ラベルのファイル仕様を表 4-4 に記載します。また、正解ラベルのサンプルを表 4-5 に記載します。

表 4-4 正解ラベルのファイル仕様

項目	仕様
ファイルパス	分析ケース設定ファイル（data.json）で指定します。分析ケース設定ファイルの詳細は、5.1.2 節を参照してください。
データ形式	CSV のフラットファイルです。既定の区切り文字は半角カンマです。 ※区切り文字は、分析ケース設定ファイル（data.json）で変更できます。
文字エンコード	UTF-8（BOM なし）です。
改行コード	LF です。
1 行目	2 行目以降のデータの項目名が書かれたヘッダ行です。
2 行目以降	1 行目の各項目に対応する項目値が書かれたデータ行です。 【マッチング機能（ssi）の場合】 各データ行は、次の 3 つの項目値を持ちます。 1. 求職者の属性データの ID（qid） 2. 求人企業の属性データの ID（tid） 3. 正解ラベル 【分類問題の場合】 求職者（qid）と求人企業（tid）の正解ラベル（既定ではマッチする［P］／ マッチしない［N］） ※正解ラベル文字は、分析ケース設定ファイル（data.json）で変更できます。 【回帰問題の場合】 実数値

	【フィルタリング機能（sse）の場合】 各データ行は、次の 2 つの項目値を持ちます。 1. 求職者の属性データの ID（qid） 2. 正解ラベル 【分類問題の場合】 求職者（qid）の正解ラベル（既定では正解［P］／ 不正解［N］） ※正解ラベル文字は、分析ケース設定ファイル（data.json）で変更できます。 【回帰問題の場合】 実数値
使用可能文字	項目名は半角英数のみ使用できます。半角空白は使用できません。
属性名の重複	項目名は重複できません。ヘッダ行でユニークな名称を設定してください。
属性値の制約	**属性名・属性値の区切り文字（既定では半角カンマ）と改行コードは、属性名・属性値に含めないよう、事前に別の文字に置換しておく必要があります。 ※区切り文字は、分析ケース設定ファイル（data.json）で変更できます。**

表 4-5 正解ラベルのサンプル

【マッチング機能（ssi）の正解ラベル例（分類問題）】

qid,tid,label［LF］

q1,t1,P［LF］

q1,t2,N［LF］

q2,t1,N［LF］

q2,t2,P［LF］

【マッチング機能（ssi）の正解ラベル例（回帰問題）】

qid,tid,label［LF］

q1,t1,0.1［LF］

q1,t2,0.5［LF］

q2,t1,0.6［LF］

q2,t2,0.9［LF］

【フィルタリング機能（sse）の正解ラベル例（分類問題）】

qid,label［LF］

q1,P［LF］

q2,N［LF］

【フィルタリング機能（sse）の正解ラベル例（回帰問題）】

```
qid,label［LF］
q1,50［LF］
q2,100［LF］
q3,300［LF］
q4,150［LF］
```

4.3 出力ファイル

本製品の出力ファイルは、予測結果レポートの 1 ファイルです。以降で、このファイル仕様を説明します。

4.3.1 予測結果レポート（result.csv）

予測結果レポートは、求職者の属性データと求人企業の属性データの関係性（正解／不正解）に関する予測結果を格納するファイルです。マッチング機能（ssi）では、求職者と求人企業に対する関係性（正解／不正解）の予測結果（マッチングスコア）を格納します。一方、フィルタリング機能（sse）では、求職者に対する関係性（正解／不正解）に対する予測結果（フィルタリングスコア）を格納します。

予測結果レポートのファイル仕様を表 4-6 に記載します。また、予測結果レポートのサンプルを表 4-7 に記載します。

表 4-6 予測結果レポートのファイル仕様

項目	仕様
ファイルパス	**＜データパス＞**/predict/result.csv
データ形式	CSV のフラットファイルです。既定の区切り文字は半角カンマです。 ※区切り文字は、システム設定ファイル（system.json）で変更できます。
文字エンコード	UTF-8（BOM なし）です。
改行コード	LF です。
1 行目	2 行目以降のデータの項目名が書かれたヘッダ行です。
2 行目以降	1 行目の各項目に対応する項目値が書かれたデータ行です。 【分類問題の場合】 【マッチング機能（ssi）】 各データ行は、次の 5 つの項目値を持ちます。 1. **[key_q]**求職者 ID 2. **[key_t]**求人企業 ID 3. **[truth]**求職者 ID と求人企業 ID の関係性に関する正解ラベル（正解／不正解） 4. **[predict]**求職者 ID と求人企業 ID の関係性に関する予測結果（正解／不正解） 5. **[score]**求職者 ID と求人企業 ID の関係性に関するマッチングスコア（0.0～1.0） 【フィルタリング機能（sse）】 各データ行は、次の 6 つの項目値を持ちます。 1. **[key]**求職者 ID 2. **[truth]**求職者 ID の正解ラベル 3. **[predict]**求職者 ID の予測結果のラベル 4. **[score_label1]**求職者 ID の正解ラベル（label1）に対する確信度（0.0～1.0） 5. **[score_label2]**求職者 ID の正解ラベル（label2）に対する確信度（0.0～1.0）

	6. **[score_labelN]**求職者 ID の正解ラベル（labelN）に対する確信度（0.0～1.0） ※指定した正解ラベル数の socre が出力されます。 【回帰問題の場合】 【マッチング機能（ssi）】 各データ行は、次の 4 つの項目値を持ちます。 1. **[key_q]**求職者 ID 2. **[key_t]**求人企業 ID 3. **[truth]**求職者 ID と求人企業 ID の関係性に関する正解値 4. **[predict]**求職者 ID と求人企業 ID の関係性に関する予測値 【フィルタリング機能（sse）】 各データ行は、次の 3 つの項目値を持ちます。 1. **[key]**求職者 ID 2. **[truth]**求職者 ID の正解値 3. **[predict]**求職者 ID の予測値

表 4-7 予測結果レポートのサンプル

【マッチング機能（ssi）の予測結果レポート例（分類問題）】

q_id,t_id,truth,predict,score［LF］

q1,t1,P,P,0.97417241334915［LF］

q1,t2,N,N,0.44298422336578［LF］

q2,t1,N,N,0.25773549079895［LF］

q2,t2,P,P,0.21370947360992［LF］

【フィルタリング機能（sse）の予測結果レポート例（分類問題）】

key,truth,predict,score_p,score_n［LF］

q1,P,P,0.99878114461899,0.0012102944310755［LF］

q2,N,N,0.45876097679138,0.54074198007584［LF］

【マッチング機能（ssi）の予測結果レポート例（回帰問題）】

q_id,t_id,truth,predict［LF］

q1,t1,0.96,0.92［LF］

q1,t2,0.03,0.13［LF］

q2,t1,0.24,0.43［LF］

q2,t2,0.87,0.44［LF］

【フィルタリング機能（sse）の予測結果レポート例（回帰問題）】

key,truth,predict［LF］

q1,0.53,0.48［LF］

q2,0.12,0.23［LF］

第5章 設定ファイル

本章では、本システムの設定ファイル群について説明します。

5.1 設定ファイル一覧

本製品が提供する設定ファイル群を表 5-1 に記載します。

表 5-1 本製品の設定ファイル一覧

設定ファイル	ファイル名	説明
システム設定ファイル	system.json	本製品のシステム設定を定義する設定ファイルです。
分析ケース設定ファイル	data.json	分析に使用する属性データ(data.csv)、正解ラベル(label.csv)、データ加工設定ファイル(idd.json)に関する情報を定義する設定ファイルです。
データ加工設定ファイル	idd.json	属性データのデータ加工方法を定義する設定ファイルです。
パラメータ設定ファイル	hparam.json	マッチング機能(ssi)、フィルタリング機能(sse)で使用するNEC 独自の機械学習アルゴリズムのパラメータを定義する設定ファイルです。
実行ログ設定ファイル	logger_config.conf	実行ログに関する情報を定義する設定ファイルです。

5.1.1 システム設定ファイル(system.json)

本製品のシステム設定を定義する設定ファイルです。システム設定ファイルのファイル仕様を表 5-2 に記載します。

表 5-2 システム設定ファイルのファイル仕様

項目	仕様
ファイルパス	**<インストールパス>/**template/matching/etc/system.json
データ形式	json 形式です。「設定項目:設定値」の形式で設定します。
文字エンコード	UTF-8(BOM なし)です。
改行コード	LF です。

表 5-3 システム設定ファイル設定項目

JSON_KEY		説明	データ型	デフォルト値
path				
	shaped_path	データ加工コマンドの出力先ディレクトリの**<データパス>**からの相対パスです。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	"shaped_data/"

	tshaped_data_q	データ加工コマンドにより加工された学習用求職者属性データファイルのファイル名です。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	"shaped_data_train_q.csv"
	tshaped_data_t	データ加工コマンドにより加工された学習用求人企業属性データファイルのファイル名です。マッチング機能(ssi)のみで使用されます。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	"shaped_data_train_t.csv"
	vshaped_data_q	データ加工コマンドにより加工された検証用求職者属性データファイルのファイル名です。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	"shaped_data_validate_q.csv"
	vshaped_data_t	データ加工コマンドにより加工された検証用求人企業属性データファイルのファイル名です。マッチング機能(ssi)のみで使用されます。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	"shaped_data_validate_t.csv"
	pshaped_data_q	データ加工コマンドにより加工された予測用求職者属性データファイルのファイル名です。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	"shaped_data_predict_q.csv"
	pshaped_data_t	データ加工コマンドにより加工された予測用求人企業属性データファイルのファイル名です。マッチング機能(ssi)のみで使用されます。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	"shaped_data_predict_t.csv"
	train_path	学習コマンドの出力先ディレクトリの**<データパス>**からの相対パスです。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“train/”
	model_path	学習コマンドが生成する予測モデルのファイル名です。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“model”
	model_ext	学習コマンドが生成する予測モデルの拡張子名です。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“.bin”
	word_dic_pref	属性データファイルをベクトル化する際に使用した辞書のファイル名です。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“word_dic_”

	word_dic_ext	属性データファイルをベクトル化する際に使用した辞書のファイル名の拡張子名です。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“.pkl”
	predict_path	予測コマンドの出力先ディレクトリの**<データパス>**からの相対パスです。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“predict/”
	result_csv	予測コマンドが出力する予測結果レポートのファイル名です。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“result_”
	result_csv_ext	予測コマンドが出力する予測結果レポートのファイル名の拡張子です。半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“.csv”
	idd_suffix	データ加工設定ファイル名の接尾辞です。	文字列	“_idd.json”
idd				
	T	データ加工設定ファイルの有効フラグを定義する文字です。この設定は変更できません。	文字列	“T”
	F	データ加工設定ファイルの有効フラグを定義する文字です。この設定は変更できません。	文字列	“F”
	item			
	idxname			
	use_flag	データ加工設定ファイルの use_flag の項目名を定義する文字です。この設定は変更できません。	文字列	“use_flag”
	key_flag	データ加工設定ファイルの key_flag の項目名を定義する文字です。この設定は変更できません。	文字列	“key_flag”
	not_null_flag	データ加工設定ファイルの not_null_flag の項目名を定義する文字です。この設定は変更できません。	文字列	“not_null_flag”
	data_type	データ加工設定ファイルの data_type の項目名を定義する文字です。この設定は変更できません。	文字列	“data_type”
	processes	データ加工設定ファイルの processes の項目名を定義する文字です。この設定は変更できません。	文字列	“processes”
	param	データ加工設定ファイルの param の項目名を定義する文字です。この設定は変更できません。	文字列	“param”
	data_type			
	numeric	データ加工設定ファイルの「data_type」項	文字列	“numeric”

				目に指定する数値型の文字列定義です。この設定は変更できません。		
			category	データ加工設定ファイルの「data_type」項目に指定するカテゴリ型の文字列定義です。この設定は変更できません。	文字列	“category”
			text	データ加工設定ファイルの「data_type」項目に指定する分かち書き対象文字列型の文字列定義です。この設定は変更できません。	文字列	“text”
			morphed	データ加工設定ファイルの「data_type」項目に指定する分かち書き済み文字列型の文字列定義です。この設定は変更できません。	文字列	“morphed”
		processes				
			round	データ加工設定ファイルの「processes」項目に指定する前処理の文字列定義です。この設定は変更できません。	文字列	“round”
			binalize_column	データ加工設定ファイルの「processes」項目に指定する前処理の文字列定義です。この設定は変更できません。	文字列	“binalize_column”
			normalize_num	データ加工設定ファイルの「processes」項目に指定する前処理の文字列定義です。この設定は変更できません。	文字列	“normalize_num”
			standardize_num	データ加工設定ファイルの「processes」項目に指定する前処理の文字列定義です。この設定は変更できません。	文字列	“standardize_num”
			logarithm_num	データ加工設定ファイルの「processes」項目に指定する前処理の文字列定義です。この設定は変更できません。	文字列	“logarithm_num”
		param				
			dummy	データ加工設定ファイルの「param」項目に指定するダミー変数の文字列定義です。この設定は変更できません。	文字列	“dummy”
			mean	データ加工設定ファイルの「param」項目に指定する統計値の文字列定義です。この設定は変更できません。	文字列	“mean”
			max	データ加工設定ファイルの「param」項目に指定する統計値の文字列定義です。この設定は変更できません。	文字列	“max”
			min	データ加工設定ファイルの「param」項目に指定する統計値の文字列定義です。この設定は変更できません。	文字列	“min”
			std	データ加工設定ファイルの「param」項目に指定する統計値の文字列定義です。この設定は変更できません。	文字列	“std”
			var	データ加工設定ファイルの「param」項目に指定する統計値の文字列定義です。この	文字列	“var”

		設定は変更できません。		
	process_delim	データ加工設定ファイルの「processes」項目に複数設定する場合に使用するデリミタ文字です。半角空白文字が既定文字です。この設定は変更できません。	文字列	“,”
	morph_delim	データ加工設定ファイルの「data_type」項目に「morphed」を指定する場合に使用する分かち書き済み文字列のデリミタ文字です。この設定は変更できません。	文字列	“ “
join		マッチング機能（ssi_rl／ssi_mse）で使用する求職者 ID と求人企業 ID の結合文字です。半角英数文字を指定できます。	文字列	“_”
result				
	delim	予測結果レポートのデリミタ文字です。この設定は変更できません。	文字列	“,”
algorithm				
	sse	NEC 独自の機械学習アルゴリズムである SSE（sse）の文字列定義です。この設定は変更できません。	文字列	“sse”
	ssi	NEC 独自の機械学習アルゴリズムである SSI（ssi）の文字列定義です。この設定は変更できません。	文字列	“ssi”
mode				
	CLASSIFICATION	分類問題の文字列定義です。この設定は変更できません。	文字列	“cls”
	REGRESSION	回帰問題の文字列定義です。この設定は変更できません。	文字列	“reg”
padding		辞書ファイルで未知語のインデックスとして割り当てる文字列の定義です。 この設定は変更できません。	文字列	"[[PADDING]]"
padding_len		テキストをベクトル化する際に、不足する長さを padding するための文字列の定義です。 この設定は変更できません。	文字列	"[[PADDING_LEN]]"
predict				
	score_digit	予測結果レポートに出力する確信度の有効桁数です。「3 桁」が既定の有効桁数です。	文字列	“%.3f”
	no_label	オンライン予測プロセスの予測状態不明の文字列定義です。この設定は変更できません。	文字列	“na”
data_conf				
	root_path	分析ケース設定ファイルの**[root_path]**設定項目の文字列定義です。この設定は変更できません。	文字列	“root_path”
	algorithm	分析ケース設定ファイルの**[algorithm]**設定項目の文字列定義です。この設定は変	文字列	“algorithm”

		更できません。		
	train_data_q	分析ケース設定ファイルの**[train_data_q]**設定項目の文字列定義です。この設定は変更できません。	文字列	“train_data_q”
	train_data_t	分析ケース設定ファイルの**[train_data_t]**設定項目の文字列定義です。この設定は変更できません。	文字列	“train_data_t”
	train_label	分析ケース設定ファイルの**[train_label]**設定項目の文字列定義です。この設定は変更できません。	文字列	“train_label”
	validate_data_q	分析ケース設定ファイルの**[validate_data_q]**設定項目の文字列定義です。この設定は変更できません。	文字列	“validate_data_q”
	validate_data_t	分析ケース設定ファイルの**[validate_data_t]**設定項目の文字列定義です。この設定は変更できません。	文字列	“validate_data_t”
	validate_label	分析ケース設定ファイルの**[validate_label]**設定項目の文字列定義です。この設定は変更できません。	文字列	“validate_label”
	predict_data_q	分析ケース設定ファイルの**[predict_data_q]**設定項目の文字列定義です。この設定は変更できません。	文字列	“predict_data_q”
	predict_data_t	分析ケース設定ファイルの**[predict_data_t]**設定項目の文字列定義です。この設定は変更できません。	文字列	“predict_data_t”
	predict_label	分析ケース設定ファイルの**[predict_label]**設定項目の文字列定義です。この設定は変更できません。	文字列	“predict_label”
	data_delim	分析ケース設定ファイルの**[data_delim]**設定項目の文字列定義です。この設定は変更できません。	文字列	“data_delim”
	label_delim	分析ケース設定ファイルの**[label_delim]**設定項目の文字列定義です。この設定は変更できません。	文字列	“label_delim”
	idd_q	分析ケース設定ファイルの**[idd_q]**設定項目の文字列定義です。この設定は変更できません。	文字列	“idd_q”
	idd_t	分析ケース設定ファイルの**[idd_t]**設定項目の文字列定義です。この設定は変更できません。	文字列	“idd_t”
	label	分析ケース設定ファイルの**[label]**設定項目の文字列定義です。この設定は変更できません。	文字列	“label”
	model_path	分析ケース設定ファイルの**[model_path]**設定項目の文字列定義です。この設定は変更できません。	文字列	“model_path”
	word_dic_pref	分析ケース設定ファイルの	文字列	"word_dic_pref"

		[word_dic_pref]設定項目の文字列定義です。この設定は変更できません。		
	word_dic_ext	分析ケース設定ファイルの**[word_dic_ext]**設定項目の文字列定義です。この設定は変更できません。	文字列	"word_dic_ext"
	result_csv	分析ケース設定ファイルの**[result_csv]**設定項目の文字列定義です。この設定は変更できません。	文字列	"result_csv"
	result_csv_ext	分析ケース設定ファイルの**[result_csv_ext]**設定項目の文字列定義です。この設定は変更できません。	文字列	“result_csv_ext”
	log_config	分析ケース設定ファイルの**[log_config]**設定項目の文字列定義です。この設定は変更できません。	文字列	“log_config”

5.1.2 分析ケース設定ファイル（data.json）

分析に使用する属性データ（data.csv）、正解ラベル（label.csv）、データ加工設定ファイル（idd.json）に関する情報を定義する設定ファイルです。ファイル仕様は、マッチング機能（ssi）とフィルタリング機能（sse）とで異なります。

マッチング機能（ssi）

マッチング機能（ssi）に対応する分析ケース設定ファイルのファイル仕様を表 5-4、設定項目を表 5-5 に記載します。

表 5-4 分析ケース設定ファイルのファイル仕様 - マッチング機能（ssi）

項目	仕様
ファイルパス	**＜データパス＞**/data.json
データ形式	json 形式です。「設定項目:設定値」の形式で設定します。
文字エンコード	UTF-8（BOM なし）です。
改行コード	LF です。

表 5-5 分析ケース設定ファイル設定項目 - マッチング機能（ssi）

JSON KEY	説明	データ型	デフォルト値
root_path	分析に使用する属性データ（data.csv）、正解ラベル（label.csv）、データ加工設定ファイル（idd.json）の格納ディレクトリの絶対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」（半角引用符）で囲んでください。	文字列	“/opt/nec/pyrapid/template/matching/project_sample/ssi/"
algorithm	分析に使用する機械学習アルゴリズムを設定します。「ssi」を指定できます。	文字列	“ssi”

train_data_q	学習用求職者属性データファイルの[**root_path**]からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“train_data_q.csv”
train_data_t	学習用求人企業属性データファイルの[**root_path**]からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“train_data_t.csv”
train_label	学習用正解ラベルファイルの[**root_path**]からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“train_label.csv”
validate_data_q	検証用求職者属性データファイルの[**root_path**]からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“validate_data_q.csv”
validate_data_t	検証用求人企業属性データファイルの[**root_path**]からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“validate_data_t.csv”
validate_label	検証用正解ラベルファイルの[**root_path**]からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“validate_label.csv”
predict_data_q	予測用求職者属性データファイルの[**root_path**]からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“predict_data_q.csv”
predict_data_t	予測用求人企業属性データファイルの[**root_path**]からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“predict_data_t.csv”
predict_label	予測用正解ラベルファイルの[**root_path**]からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“predict_label.csv”
data_delim	属性データファイルのデリミタ文字です。「,」が既定文字です。半角英数字、半角空白文字、半角記号文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“,”
label_delim	正解ラベルファイルのデリミタ文字です。「,」が既定文字です。半角英数字、半角空白文字、半角記号文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“,”
idd_q	求職者属性データファイルに対するデータ加工設定ファイルの[**root_path**]からの相対パスです。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“idd_q.json”

idd_t	求人企業属性データファイルに対するデータ加工設定ファイルの**[root_path]**からの相対パスです。全角文字、半角英数字、半角空白文字を指定できます。 文字列は「"」(半角引用符)で囲んでください。	文字列	“idd_t.json”
label	分類問題に使用する正解ラベルのラベル文字定義です。マッチングでは、マッチする、マッチしないの 2 つのラベルを定義してください。 半角英数字を指定できます。文字列は「"」(半角引用符)で囲んでください。	配列	[“P”, ”N”]
log_config	実行ログ設定ファイルを読み込む場合、実行ログ設定ファイルの絶対パスを指定します。「""」が既定文字です。 文字列は「"」(半角引用符)で囲んでください。 ログ設定ファイルの詳細は 5.1.5 節を参照してください。	文字列	“”

フィルタリング機能(sse)

フィルタリング機能(sse)に対応する分析ケース設定ファイルのファイル仕様を表 5-6、設定項目を表 5-7 に記載します。

表 5-6 分析ケース設定ファイルのファイル仕様 - フィルタリング機能(sse)

項目	仕様
ファイルパス	**<データパス>**/data.json
データ形式	json 形式です。「設定項目:設定値」の形式で設定します。
文字エンコード	UTF-8(BOM なし)です。
改行コード	LF です。

表 5-7 分析ケース設定ファイル設定項目 – フィルタリング機能(sse)

JSON KEY	説明	データ型	デフォルト値
root_path	分析に使用する属性データ(data.csv)、正解ラベル(label.csv)、データ加工設定ファイル(idd.json)の格納ディレクトリの絶対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“/opt/nec/pyrapid/template/matching/project_sample/sse/"
algorithm	分析に使用する機械学習アルゴリズムを設定します。「sse」を設定できます。	文字列	“sse”
train_data_q	学習用求職者属性データファイルの**[root_path]**からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“train_data_q.csv”
train_label	学習用正解ラベルファイルの**[root_path]**からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は	文字列	“train_label.csv”

	「"」(半角引用符)で囲んでください。		
validate_data_q	検証用求職者属性データファイルの**[root_path]**からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“validate_data_q.csv”
validate_label	検証用正解ラベルファイルの**[root_path]**からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“validate_label.csv”
predict_data_q	予測用求職者属性データファイルの**[root_path]**からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“predict_data_q.csv”
predict_label	予測用正解ラベルファイルの**[root_path]**からの相対パスを設定します。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“predict_label.csv”
data_delim	属性データファイルのデリミタ文字です。「,」が既定文字です。半角英数字、半角空白文字、半角記号文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“,”
label_delim	正解ラベルファイルのデリミタ文字です。「,」が既定文字です。半角英数字、半角空白文字、半角記号文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“,”
idd_q	求職者属性データファイルに対するデータ加工設定ファイルの**[root_path]**からの相対パスです。全角文字、半角英数字、半角空白文字を指定できます。文字列は「"」(半角引用符)で囲んでください。	文字列	“idd_q.json”
label	分類問題に使用する正解ラベルのラベル文字定義です。フィルタリングでは分析対象に合わせて 2 つ以上のラベルを定義してください。半角英数字を指定できます。文字列は「"」(半角引用符)で囲んでください。	配列	[“P”, “N”]
log_config	実行ログ設定ファイルを読み込む場合、実行ログ設定ファイルの絶対パスを指定します。「""」が既定文字です。 文字列は「"」(半角引用符)で囲んでください。 ログ設定ファイルの詳細は 5.1.5 節を参照してください。	文字列	“”

5.1.3 データ加工設定ファイル(idd.json)

属性データのデータ加工方法を定義する設定ファイルです。マッチング機能(ssi)では、求職者に関する属性データ(data_q.csv)と、求人企業に関する属性データ(data_t.csv)の 2 つの入力

ファイルに対応するデータ加工設定ファイル（idd_q.json／idd_t.json）がそれぞれ必要です。一方、フィルタリング機能（sse）では、求職者に関する属性データ（data_q.csv）の 1 つの入力ファイルに対応するデータ加工設定ファイル（idd.json）が必要です。

データ加工設定ファイルのファイル仕様を表 5-8、設定項目を表 5-9 に記載します。また、データ加工設定ファイルのサンプルを表 5-10 に記載します。

表 5-8 データ加工設定ファイル仕様

項目	仕様
ファイルパス	分析ケース設定ファイル（data.json）で指定します。分析ケース設定ファイルの詳細は、5.1.2 節を参照してください。
データ形式	CSV のフラットファイルです。既定の区切り文字は半角カンマです。 ※区切り文字は、分析ケース設定（data.json）で変更できます。
文字エンコード	UTF-8（BOM なし）です。
改行コード	LF です。

表 5-9 データ加工設定ファイル設定項目

JSON KEY		説明	範囲
[COLUMN_NAME]		属性データの属性名を設定します。	
	use_flag	属性データの該当属性をデータ加工対象とするかどうかを設定します。 **[T]**データ加工対象とします。 **[F]**データ加工対象から除外します。	“T”/ ”F”
	key_flag	属性データの該当属性をキー文字列（属性データの各行をユニークに特定するための ID）として使用するかどうかを設定します。 **[T]**キー文字列とします。 **[F]**キー文字列から除外します。	“T”/ “F”
	not_null_flag	属性データの該当属性が空文字の場合に、属性データの該当行を実行ログに WARNING 出力する機能を有効にするかどうかを設定します。 **[T]**有効にします。 **[F]**無効にします。	“T”/ “F”
	data_type	属性データの該当属性のデータ種別を設定します。 **[numeric]**数値型です。 **[category]**カテゴリ型です。 **[text]**分かち書き対象文字列型です。 **[morphed]**分かち書き済み文字列型です。	“numeric”/ “category”/ “text”/ “morphed”
	processes	属性データの該当属性に対するデータ加工処理方法を設定します。**※19** データ加工処理方法は複数設定できます。 **[round]** 数値の丸めを行います。値に丸めの単位を 0.0 より大きい実数で設定します。 **例）“round”**：0.5	“round”:実数値/ “binalize_column”/ “normalize_num”/ “stanadrdize_num”/ “logarithm_num”

		[binalize_column] 　　カテゴリ値のダミー変数化を行います **[normalize_num]** 　　数値の正規化を行います **[standardize_num]** 　　数値の標準化を行います **[logarithm_num]** 　　数値の対数化を行います	

⚠注 19: [processes]で処理方法を指定しない場合

[processes]設定項目を指定しない場合は、"processes": [] と設定してください。

OK 例）

```
{
    "q_id": {
        "data_type": "numeric",
        "key_flag": "T",
        "use_flag": "T",
        "not_null_flag":"F",
        "processes": [
        ]
    },
    "q_attr1": {
        "data_type": "text",
        "key_flag": "F",
        "use_flag": "T",
        "not_null_flag":"F",
        "processes": [
        ]
    }
}
```

NG 例）

```
{
    "q_id": {
        "data_type": "numeric",
        "key_flag": "T",
        "use_flag": "T",
        "not_null_flag":"F"
    },
    "q_attr1": {
        "data_type": "text",
```

```
            "key_flag": "F",
            "use_flag": "T",
            "not_null_flag":"F"
        }
}
```

表 5-10 データ加工設定ファイルのサンプル

```
{
    "q_id": {
        "data_type": "numeric",
        "key_flag": "T",
        "use_flag": "T",
        "not_null_flag":"F",
        "processes": [
        ]
    },
    "q_attr1": {
        "data_type": "text",
        "key_flag": "F",
        "use_flag": "T",
        "not_null_flag":"F",
        "processes": [
        ]
    },
    "q_attr2": {
        "data_type": "text",
        "key_flag": "F",
        "use_flag": "T",
        "not_null_flag":"F",
        "processes": [
        ]
    }
}
```

5.1.4 パラメータ設定ファイル（hparam.json）

マッチング機能（ssi）、フィルタリング機能（sse）で使用する NEC 独自の機械学習アルゴリズムのパラメータを定義する設定ファイルです。ファイル仕様は機械学習アルゴリズム（ssi／sse）毎に異なります。

マッチング機能（ssi）

マッチング機能（ssi）に対応するパラメータ設定ファイルのファイル仕様を表 5-11、設定項目を表 5-12 に記載します。

表 5-11 パラメータ設定ファイル仕様 - マッチング機能（ssi）

項目	仕様
ファイルパス	＜任意のパス＞/hparam.json
データ形式	json 形式です。「設定項目:設定値」の形式で設定します。
文字エンコード	UTF-8（BOM なし）です。
改行コード	LF です。

表 5-12 パラメータ設定ファイル設定項目 - マッチング機能（ssi）

JSON KEY	説明	データ型	範囲	デフォルト値
algorithm	使用するアルゴリズム名です。“ssi”を指定してください。	文字列	ssi	ssi
epoch	学習コマンド実行時のエポック数です。1 以上の整数値を指定できます。	整数値	1~	10
lrate	学習コマンド実行時の学習率の開始値です。0 より大きい実数値を指定できます。	実数値	0~	0.1
batch_size	ミニバッチのバッチサイズを指定します。	整数値	1~	10
embedding_dim_q	ネットワークのパラメータです。属性値（クエリ）のテキスト列を embedding_dim_q 次元まで縮約します。	整数値	1~	10
emmbedding_dim_num_q	ネットワークのパラメータです。属性値（クエリ）の数値列を emmbedding_dim_num_q 次元まで縮約します。	整数値	1~	20
emmbedding_dim_num_q_mid	ネットワークのパラメータです。属性値（クエリ）の数値列を emmbedding_dim_num_q 次元から、emmbedding_dim_num_q_mid 次元まで縮約します。	整数値	1~	10
out_dim_q	ネットワークのパラメータです。属性値（クエリ）の出力の次元数です。	整数値	1~	10
embedding_dim_t	ネットワークのパラメータです。属性値（ターゲット）のテキスト	整数値	1~	10

	列を embedding_dim_t 次元まで縮約します。			
emmbedding_dim_num_t	ネットワークのパラメータです。属性値（ターゲット）の数値列をemmbedding_dim_num_t 次元まで縮約します。	整数値	1~	20
emmbedding_dim_num_t_mid	ネットワークのパラメータです。ネットワークのパラメータです。属性値（ターゲット）の数値列をemmbedding_dim_num_t 次元から、emmbedding_dim_num_t_mid 次元まで縮約します。	整数値	1~	10
out_dim_t	ネットワークのパラメータです。属性値（ターゲット）の出力の次元数です。	整数値	1~	10
matching_dim	ネットワークのパラメータです。マッチングを行う次元数を指定します。	整数値	1~	10
ngram_q	属性データ（クエリ）のテキスト列に使用する N-gram を設定します。	整数値	1~	2
ngram_t	属性データ（ターゲット）のテキスト列に使用する N-gram を設定します。	整数値	1~	2
eps	ネットワークのパラメータです。0 以上の微小量。	浮動小数点数		1e-6
dropout_rate	ドロップアウト率を設定します。	実数値	0.0~1.0	0.5
shuffle	ミニバッチでデータをシャッフルするか設定します。 true:データをシャッフルする false:ラベルの順序でミニバッチを作成する	BOOL	true/false	false
num_workers	学習、予測に使用するプロセス数を指定します。1 以上の整数値を指定できます。	整数値	1~	1
cutoff	SSI（ssi）が算出する確信度（0.0～1.0）から正解（P）／不正解（N）に 2 値化する際に使用するカットオフ値です。「0.5」が既定値です。[0.0,1.0]の間の実数値を指定できます。	実数値	0.0~1.0	0.5

フィルタリング機能（sse）

フィルタリング機能（sse）に対応するパラメータ設定ファイルのファイル仕様を表 5-13、設定項目を表 5-14 に記載します。

表 5-13 パラメータ設定ファイル仕様 - フィルタリング機能（sse）

項目	仕様
ファイルパス	＜任意のパス＞/hparam.json
データ形式	json 形式です。「設定項目:設定値」の形式で設定します。
文字エンコード	UTF-8（BOM なし）です。
改行コード	LF です。

表 5-14 パラメータ設定ファイル設定項目 - フィルタリング機能（sse）

JSON KEY	説明	データ型	範囲	デフォルト値
algorithm	使用するアルゴリズム名です。“sse”を指定してください。	文字列	sse	sse
epoch	学習コマンド実行時のエポック数です。「10」が既定値です。1 以上の整数値を指定できます。	整数値	1~	10
lrate	学習コマンド実行時の学習率です。0 より大きい実数値を指定できます。	実数値	0~	0.1
batch_size	ミニバッチのバッチサイズを指定します。	整数値	1~	10
embedding_dim_q	ネットワークのパラメータです。属性値（クエリ）のテキスト列を embedding_dim_q 次元まで縮約します。	整数値	1~	10
emmbedding_dim_num_q	ネットワークのパラメータです。属性値（クエリ）の数値列を emmbedding_dim_num_q 次元まで縮約します。	整数値	1~	20
emmbedding_dim_num_q_mid	ネットワークのパラメータです。属性値（クエリ）の数値列を emmbedding_dim_num_q 次元から、emmbedding_dim_num_q_mid 次元まで縮約します。	整数値	1~	10
out_dim_q	ネットワークのパラメータです。属性値（クエリ）の出力の次元数です。	整数値	1~	10
ngram_q	属性データ（クエリ）のテキスト列に使用する N-gram を設定	整数値	1~	2

	します。			
dropout_rate	ドロップアウト率を設定します。	実数値	0~1	0.5
shuffle	ミニバッチでデータをシャッフルするか設定します。 true:データをシャッフルする false:ラベルの順序でミニバッチを作成する	BOOL	true/false	false
num_workers	学習、予測に使用するプロセス数を指定します。1 以上の整数値を指定できます。	整数値	1~	1
set_weight	ラベル毎に重みの設定を行うか指定します。 true:重みの設定を行う。重みの設定方法は、weight で指定。 false:重みの設定を行わない。	BOOL	true/false	false
weight	set_weight を true にした場合のラベル毎の重みを設定します。 “auto”: ラベルの件数に応じた重みを自動で設定します。 “default”: すべてのラベルの重みを 1 にします。 [wight1, weight2]: 各ラベルに任意の重みを設定する場合、配列で指定します **例）**[0.5, 0.8]	文字列/配列	“auto”/“default”/配列で指定する場合 0.0~	“default”

5.1.5 実行ログ設定ファイル（logger_config.conf）

本製品の実行ログに関する設定を定義する設定ファイルです。実行ログ設定ファイルのファイル仕様を表 5-15、設定ファイルの設定項目を表 5-16、デフォルトの設定ファイルを表 5-17 に記載します。

表 5-15 実行ログ設定ファイル仕様

項目	仕様
ファイルパス	/opt/nec/pyrapid/template/matching/etc/logger_config.conf

データ形式	INI 形式です。「設定項目=設定値」の形式で設定します。
文字エンコード	UTF-8（BOM なし）です。
改行コード	LF です。

表 5-16　実行ログ設定ファイル項目

項目		説明
[loggers]	keys	ログ設定名です。 「root」を指定してください。
[handlers]	keys	ログ出力先設定名です。 「fileHandler」を指定してください。
[formatters]	keys	ログフォーマット設定名です。 「simpleFormatter」を指定してください。
[logger_root]	level	ログの出力レベルです。既定では「INFO」です。 「DEBUG」、「INFO」、「WARN」、「ERROR」、 「FATAL」のいずれかを指定できます。
	handlers	使用するログ出力設定です。 「fileHandler」を指定してください。
[handler_fileHandler]	class	ログ出力形式です。 「FileHandler」を指定してください。
	formatter	使用するログフォーマット設定です。 「simpleFormatter」を指定してください。
	args	出力するログファイルのパスの設定です。 (ログファイルの絶対パス, 'a' , 'utf-8')
[formatter_simpleFormatter]	format	出力するログのフォーマットです。

表 5-17　実行ログ設定ファイルサンプル

```
[loggers] ［LF］
keys=root ［LF］

[handlers] ［LF］
keys=fileHandler ［LF］

[formatters] ［LF］
```

```
keys=simpleFormatter ［LF]


[logger_root] ［LF]
level=ERROR ［LF]
handlers=fileHandler ［LF]


[handler_fileHandler] ［LF]
class=FileHandler ［LF]
formatter=simpleFormatter ［LF]
args=('/var/log/nec/pyrapid/etc/test_logger.log','a', 'utf-8') ［LF]


[formatter_simpleFormatter] ［LF]
format = %(levelname)s %(process)d %(filename)s %(funcName)s %(asctime)s %(message)s ［LF]
```

第6章 ログファイル

本章では、本製品が出力するログファイル群について説明します。

6.1 ログファイル一覧

本製品が出力するログファイル一覧を表 6-1 に記載します。

表 6-1 本製品が出力するログファイル一覧

ログファイル	ファイル名	説明
実行ログ	HRM.log	マッチングテンプレート（図 2-3）が実行時に出力するアプリケーションログ
学習誤差評価ログ	train_eval.log	マッチングテンプレート（図 2-3）が学習時に出力する学習誤差に関するアプリケーションログ

6.1.1 実行ログ（pyrapid_logger.log）

マッチングテンプレート（図 2-3）が実行時に出力するアプリケーションログです。

実行ログのファイル仕様を表 6-2 に記載します。また、実行ログの出力サンプルを表 6-3 に記載します。

表 6-2 実行ログのファイル仕様

項目		仕様
ファイルパス		実行ログ設定ファイル（logger_config.conf）で指定します。実行ログ設定ファイルの詳細は、 5.1.5 節を参照してください。 /var/log/nec/pyrapid/etc/pyrapid_logger.log がデフォルトのパスになります。
データ形式		CSV のフラットファイルです。既定の区切り文字は半角空白文字です。
文字エンコード		UTF-8（BOM なし）です。
改行コード		LF です。
出力項目		
項 1	ログレベル	次のいずれかのログレベルを出力します。 **1.** **[FATAL]**本製品の実行を継続できない想定外のエラーが発生した場合に出力します **2.** **[ERROR]**本製品の実行を継続できない想定内のエラーが発生した場合に出力します **3.** **[WARN]**本製品の実行は継続できますが、利用者に通知すべき想定内のエラーが発生した場合に出力します **4.** **[INFO]**本製品の実行時、実行経過に関する簡易情報を出力します **5.** **[DEBUG]**本製品の実行時、実行経過に関する詳細情報を出力します
項 2	プロセス ID	本製品の実行プロセスのプロセス ID です。
項 3	コマンド名	実行したコマンド名を出力します。
項 4	関数名	実行中の関数名を出力します
項 5	ログ発生時刻	ログ発生時刻を「yyyy-mm-dd hh:mm:ss.sss」形式で出力します。

項 6	ログメッセージ	ログのメッセージ本文です。ログレベルが[**ERROR**]の場合、エラーコードをあわせて出力します。エラーコードの詳細は、第 7 章 を参照してください。

表 6-3 実行ログの出力サンプル

```
INFO 23987 convert.py main 2018-04-03 11:38:18,838 [START] convert['/opt/nec/pyrapid/template/matching/bin/convert.rpd', 'cls', 'data_conf.json']［LF]
```

6.1.2 学習誤差評価ログ（train_log.log）

マッチングテンプレート（図 2-3）が学習時に出力する学習誤差に関するアプリケーションログです。

学習時に出力する学習誤差評価ログのファイル仕様を表 6-4 学習誤差評価ログのファイル仕様表 6-4 に、各項目の説明を表 6-5 記載します。また、出力サンプルを表 6-6 に記載します。

表 6-4 学習誤差評価ログのファイル仕様

項目	仕様
ファイルパス	**＜データパス＞**/train/train_log.log
データ形式	CSV のフラットファイルです。既定の区切り文字は半角カンマです。
文字エンコード	UTF-8（BOM なし）です。
改行コード	LF です。

表 6-5 学習誤差評価ログの項目

項目	説明
epoch	epoch 数です。学習データが含むすべてのサンプルデータを 1 回学習した状態を 1 epoch と定義します。
count	学習した学習データの総サンプル数です。
avg_loss	学習データに対する 1epoch での誤差の平均
avg_loss_validate	検証データに対する 1epoch での誤差の平均

表 6-6 学習誤差評価ログの出力サンプル

```
========== START @ 03/07/18 11:08:58 ==========

epoch, count, avg_loss, avg_loss_validate

1,7500,0.50753,0.67185 [LF]

2,15000,0.59393,0.74330 [LF]

3,22500,0.61440,0.78329 [LF]
```

第7章 エラーメッセージ

本章では、本製品のコマンド群が出力するエラーメッセージ一覧を説明します。

7.1 データ加工コマンド

データ加工コマンドに関するエラーコード、原因、処置の一覧を表 7-1 に記載します。

表 7-1 データ加工コマンドのエラーコード一覧

エラーコード	原因	処置
HRM0502101	デフォルトの ログ設定ファイルの読み込みに失敗しました。	デフォルトの実行ログ設定ファイルが「<インストールパス>/template/matching/etc/logger_config.conf」に存在することを確認してください。
HRM0502102	デフォルトの ログ出力先ディレクトリがありません。	デフォルトの実行ログ設定ファイルの項目「[handler_fileHandler] args」に指定したパスが正しいか確認してください。
HRM0502103	システム設定ファイルの読み込みに失敗しました。	システム設定ファイルが「<インストールパス>/template/matching/etc/system.conf」に存在することを確認してください。
HRM0502104	ログ設定ファイルで指定されたログ出力ディレクトがありません。	実行ログ設定ファイルの項目「[handler_fileHandler] args」に指定したパスが正しいか確認してください。
HRM0502105	ログ設定ファイルの読み込みに失敗しました。	「5.1.5 実行ログ設定ファイル（logger_config.conf）」を参照し、実行ログ設定ファイルの設定を見直してください。
HRM0502106	データ加工設定ファイルの読み込みに失敗しました。	データ加工設定ファイルで指定した値が間違っています。ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。
HRM0502109	属性データファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。
HRM0502110	指定先の学習用ファイルが存在しません。	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認してください。
HRM0502111	データ加工設定ファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。
HRM0502112	前処理後のデータ加工設定ファイルがありません。	ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。
HRM0502113	検証データファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。
HRM0502114	検証データファイルが存在しません。	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認してください。
HRM0502115	予測データファイルの読み込に失敗しました。	ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。

HRM0502116	予測データファイルが存在しません	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認してください。
HRM0502117	分析ケース設定ファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認してください。
HRM0502901	使用できない型が使われています。	PP・サポートサービス からお問い合わせください。
HRM0502999	内部エラーが発生しました。	ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。

7.2 学習コマンド（train_ssi.rpd）

学習コマンドに関するエラーコード、原因、処置の一覧を表 7-2 に記載します。

表 7-2 学習コマンドのエラーコード一覧

エラーコード	原因	処置
HRM0106101	デフォルトの ログ設定ファイルの読み込みに失敗しました。	デフォルトの実行ログ設定ファイルが「<インストールパス >/template/matching/etc/logger_config.conf」に存在することを確認してください。
HRM0106102	デフォルトの ログ出力先ディレクトリがありません。	デフォルトの実行ログ設定ファイルの項目「[handler_fileHandler] args」に指定したパスが正しいか確認してください。
HRM0106103	システム設定ファイルの読み込みに失敗しました。	システム設定ファイルが「<インストールパス >/template/matching/etc/system.conf」に存在することを確認してください。
HRM0106104	ログ設定ファイルで指定されたログ出力ディレクトがありません。	実行ログ設定ファイルの項目「[handler_fileHandler] args」に指定したパスが正しいか確認してください。
HRM0106105	ログ設定ファイルの読み込みに失敗しました。	「5.1.5 実行ログ設定ファイル（logger_config.conf）」を参照し、実行ログ設定ファイルの設定を見直してください。
HRM0106106	ラベルの定義に含まれないラベルです。	ログファイルから当該のエラーコードを検索し、確認したラベルをもとに学習データファイルを見直してください。
HRM0106107	学習ファイルの[text]を指定した列に[ngram_q]で指定した値以下の単語数の行が存在します。	当該ファイルの当該列のレコードを取り除くか、[ngram_q]に指定する値を見直してください。
HRM0106108	学習ファイルの[text]を指定した列に[ngram_t]で指定した値以下の単語数の行が存在します。	当該ファイルの当該列のレコードを取り除くか、[ngram_t]に指定する値を見直してください。
HRM0106117	分析ケース設定ファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認して

		ください。
HRM0106118	パラメータ設定ファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認してください。
HRM0106999	内部エラーが発生しました。	ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。
HRM0100101	データ加工設定ファイル設定項目の[key_flag]について[T]が 2 つ以上設定されています。	データ加工設定ファイルを見直し、設定項目の[key_flag]について[T]を一つだけ設定してください。
HRM0100102	データ加工設定ファイル設定項目の[key_flag]について[T]が一つも設定されていません。	データ加工設定ファイルを見直し、設定項目の[key_flag]について[T]を一つだけ設定してください。
HRM0100103	システム設定ファイルの読み込みに失敗しました。	システム設定ファイルが「<インストールパス>/template/matching/etc/system.conf」に存在することを確認してください。
HRM0100104	学習データに含まれないラベルです。	ログファイルから当該のエラーコードを検索し、確認したラベルをもとに学習データファイルを見直してください。
HRM0100105	検証ファイルの[text]を指定した列に[ngram_q]で指定した値以下の単語数の行が存在します。	当該ファイルの当該列のレコードを取り除くか、[ngram_q]に指定する値を見直してください。
HRM0100106	検証ファイルの[text]を指定した列に[ngram_t]で指定した値以下の単語数の行が存在します。	当該ファイルの当該列のレコードを取り除くか、[ngram_t]に指定する値を見直してください。

7.3 学習コマンド（train_sse.rpd）

学習コマンドに関するエラーコード、原因、処置の一覧を表 7-2 に記載します。

表 7-3 学習コマンドのエラーコード一覧

エラーコード	原因	処置
HRM0107101	デフォルトの ログ設定ファイルの読み込みに失敗しました。	デフォルトの実行ログ設定ファイルが「<インストールパス>/template/matching/etc/logger_config.conf」に存在することを確認してください。
HRM0107102	デフォルトの ログ出力先ディレクトリがありません。	デフォルトの実行ログ設定ファイルの項目「[handler_fileHandler] args」に指定したパスが正しいか確認してください。
HRM0107103	システム設定ファイルの読み込みに失敗しました。	システム設定ファイルが「<インストールパス>/template/matching/etc/system.conf」に存在することを確認してください。
HRM0107104	ログ設定ファイルで指定されたログ出力ディレクトがありません。	実行ログ設定ファイルの項目「[handler_fileHandler] args」に指定したパスが正しいか確認してください。
HRM0107105	ログ設定ファイルの読み込みに失敗しまし	「5.1.5 実行ログ設定ファイル

	た。	（logger_config.conf）」を参照し、実行ログ設定ファイルの設定を見直してください。
HRM0107106	ラベルの定義に含まれないラベルです。	ログファイルから当該のエラーコードを検索し、確認したラベルをもとに学習データファイルを見直してください。
HRM0106107	学習ファイルの[text]を指定した列に[ngram_q]で指定した値以下の単語数の行が存在します。	当該ファイルの当該列のレコードを取り除くか、[ngram_q]に指定する値を見直してください。
HRM0107117	分析ケース設定ファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認してください。
HRM0107118	パラメータ設定ファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認してください。
HRM0107999	内部エラーが発生しました。	ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。
HRM0100101	データ加工設定ファイル設定項目の[key_flag]について[T]が 2 つ以上設定されています。	データ加工設定ファイルを見直し、設定項目の[key_flag]について[T]を一つだけ設定してください。
HRM0100102	データ加工設定ファイル設定項目の[key_flag]について[T]が一つも設定されていません。	データ加工設定ファイルを見直し、設定項目の[key_flag]について[T]を一つだけ設定してください。
HRM0100103	システム設定ファイルの読み込みに失敗しました。	システム設定ファイルが「<インストールパス>/template/matching/etc/system.conf」に存在することを確認してください。
HRM0100104	学習データに含まれないラベルです。	ログファイルから当該のエラーコードを検索し、確認したラベルをもとに学習データファイルを見直してください。
HRM0100105	検証ファイルの[text]を指定した列に[ngram_q]で指定した値以下の単語数の行が存在します。	当該ファイルの当該列のレコードを取り除くか、[ngram_q]に指定する値を見直してください。

7.4 バッチ予測コマンド（predict_ssi.rpd）

バッチ予測コマンドに関するエラーコード、原因、処置の一覧を表 7-4 に記載します。

表 7-4 バッチ予測コマンドのエラーコード一覧

エラーコード	原因	処置
HRM0208101	デフォルトの ログ設定ファイルの読み込みに失敗しました。	デフォルトの実行ログ設定ファイルが「<インストールパス>/template/matching/etc/logger_config.conf」に存在することを確認してください。
HRM0208102	デフォルトの ログ出力先ディレクトリがありません。	デフォルトの実行ログ設定ファイルの項目「[handler_fileHandler] args」に指定したパスが正しいか確認してください。

HRM0208103	システム設定ファイルの読み込みに失敗しました。	システム設定ファイルが「<インストールパス>/template/matching/etc/system.conf」に存在することを確認してください。
HRM0208104	ログ設定ファイルで指定されたログ出力ディレクトがありません。	実行ログ設定ファイルの項目「[handler_fileHandler] args」に指定したパスが正しいか確認してください。
HRM0208105	ログ設定ファイルの読み込みに失敗しました。	「5.1.5 実行ログ設定ファイル（logger_config.conf）」を参照し、実行ログ設定ファイルの設定を見直してください。
HRM0208106	予測ファイルの[text]を指定した列に[ngram_q]で指定した値以下の単語数の行が存在します。	当該ファイルの当該列のレコードを取り除くか、[ngram_q]に指定する値を見直してください。
HRM0208107	予測ファイルの[text]を指定した列に[ngram_t]で指定した値以下の単語数の行が存在します。	当該ファイルの当該列のレコードを取り除くか、[ngram_t]に指定する値を見直してください。
HRM0208117	分析ケース設定ファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認してください。
HRM0208118	パラメータ設定ファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認してください。
HRM0208999	内部エラーが発生しました。	ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。
HRM0100101	データ加工設定ファイル設定項目の[key_flag]について[T]が 2 つ以上設定されています。	データ加工設定ファイルを見直し、設定項目の[key_flag]について[T]を一つだけ設定してください。
HRM0100102	データ加工設定ファイル設定項目の[key_flag]について[T]が一つも設定されていません。	データ加工設定ファイルを見直し、設定項目の[key_flag]について[T]を一つだけ設定してください。
HRM0100103	システム設定ファイルの読み込みに失敗しました。	システム設定ファイルが「<インストールパス>/template/matching/etc/system.conf」に存在することを確認してください。

7.5 バッチ予測コマンド（predict_sse.rpd）

バッチ予測コマンドに関するエラーコード、原因、処置の一覧を表 7-4 に記載します。

表 7-5 バッチ予測コマンドのエラーコード一覧

エラーコード	原因	処置
HRM0209101	デフォルトの ログ設定ファイルの読み込みに失敗しました。	デフォルトの実行ログ設定ファイルが「<インストールパス>/template/matching/etc/logger_config.conf」に存在することを確認してください。
HRM0209102	デフォルトの ログ出力先ディレクトリがあり	デフォルトの実行ログ設定ファイルの項目

	ません。	「[handler_fileHandler] args」に指定したパスが正しいか確認してください。
HRM0209103	システム設定ファイルの読み込みに失敗しました。	システム設定ファイルが「<インストールパス>/template/matching/etc/system.conf」に存在することを確認してください。
HRM0209104	ログ設定ファイルで指定されたログ出力ディレクトがありません。	実行ログ設定ファイルの項目「[handler_fileHandler] args」に指定したパスが正しいか確認してください。
HRM0209105	ログ設定ファイルの読み込みに失敗しました。	「5.1.5 実行ログ設定ファイル（logger_config.conf）」を参照し、実行ログ設定ファイルの設定を見直してください。
HRM0209106	予測ファイルの[text]を指定した列に[ngram_q]で指定した値以下の単語数の行が存在します。	当該ファイルの当該列のレコードを取り除くか、[ngram_q]に指定する値を見直してください。
HRM0209117	分析ケース設定ファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認してください。
HRM0209118	パラメータ設定ファイルの読み込みに失敗しました。	ログファイルから当該のエラーコードを検索し、指定したファイルパスが正しいか確認してください。
HRM0209999	内部エラーが発生しました。	ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。
HRM0100101	データ加工設定ファイル設定項目の[key_flag]について[T]が 2 つ以上設定されています。	データ加工設定ファイルを見直し、設定項目の[key_flag]について[T]を一つだけ設定してください。
HRM0100102	データ加工設定ファイル設定項目の[key_flag]について[T]が一つも設定されていません。	データ加工設定ファイルを見直し、設定項目の[key_flag]について[T]を一つだけ設定してください。
HRM0100103	システム設定ファイルの読み込みに失敗しました。	システム設定ファイルが「<インストールパス>/template/matching/etc/system.conf」に存在することを確認してください。

7.6 IDD 雛形生成コマンド

IDD 雛形生成コマンドに関するエラーコード、原因、処置の一覧を表 7-6 に記載します。

表 7-6 IDD 雛形生成コマンドのエラーコード一覧

エラーコード	原因	処置
HRM0704101	デフォルトの ログ設定ファイルの読み込みに失敗しました。	デフォルトの実行ログ設定ファイルが「<インストールパス>/template/matching/etc/logger_config.conf」に存在することを確認してください。
HRM0704102	デフォルトの ログ出力先ディレクトリがありません。	デフォルトの実行ログ設定ファイルの項目「[handler_fileHandler] args」に指定したパスが正しいか確認してください。

HRM0704103	システム設定ファイルの読み込みに失敗しました。	システム設定ファイルが「<インストールパス>/template/matching/etc/system.conf」に存在することを確認してください。
HRM0704108	属性データのファイルパスが存在しません。	引数で指定した属性データのファイルパスを見直してください。
HRM0704109	属性データの読み込みに失敗しました。	
HRM0704107	データ加工設定ファイルの書き込みに失敗しました。	「<データパス>」ディレクトリ配下に標準ユーザの書き込み権限が付与されていることを確認してください。
HRM0704999	内部エラーが発生しました。	ログファイルから当該のエラーコードを検索し、エラー詳細を確認してください。

第8章 注意・制限事項

本章では、本製品を利用する際の注意事項、制限事項について説明します。

8.1 注意事項

本節では、本製品を利用する際の注意事項を説明します。

8.1.1 本製品の使用時に使用するユーザ権限

- 本製品は、標準ユーザで使用してください。管理者ユーザでは使用しないでください。

8.1.2 分析ケース設定ファイルの[root_path]について

- 分析ケース設定ファイル（data.json）の**[root_path]**設定項目には、**＜データパス＞**を絶対パスで指定し、その両端をダブルクォーテーション（””）で囲ってください。

8.2 制限事項

8.2.1 コンテナ型の環境での実行について

- 本製品は、コンテナ型の環境で使用せず、物理マシン、または仮想マシンで使用してください。

第9章 トラブルシューティング

本章では、本製品を利用する際のトラブルシューティングについて説明します。

9.1 ログファイルが出力されない

- 本製品を実行する標準ユーザが、ログファイルの出力先ディレクトリへの書き込み権限を持っていることを確認してください。
- 本製品を実行する標準ユーザが、ログファイルへの書き込み権限を持っていることを確認してください。
- 動作確認などで本製品を管理者ユーザで実行すると、管理者ユーザの書き込み権限でログファイルが生成されてしまうことがあります。この場合は、いったんすべてのログファイルを削除してください。

NEC Advanced Analytics - RAPID 機械学習
マッチング for Linux V2.2
ユーザガイド

2018 年 5 月