Team SWEg API 1.0

[Base URL: /api]
http://swegapi.com

API endpoints for a microservice that predicts potential applicant success using machine learning.

Team SWEg (Eric Fan, Jihyo Chung, Kyungmin Kim, Leo Jung, Victor Waddell) - Website

models API endpoints to manage machine learning models

GET /models Gets a list of all the models

Parameters Try it out

No parameters

Responses Response content type: application/json

Code	Description
200	Success

Example Value | Model

```
{
  "model_id": "string",
  "train_acc": 1,
  "valid_acc": 1,
  "model_class": "logistic",
  "score_func": "f_regression",
  "num_features": 1,
  "k": 1
}
```

POST /models Creates and trains a model with given model class and hyperparameters

Parameters Try it out

Name	Description
payload * required object (body)	

Example Value | Model

```
{
  "model_class": "logistic",
  "score_func": "f_regression",
  "num_features": 1,
  "k": 1
}
```

Parameter content type: application/json

Responses Response content type: application/json

Code	Description
201	Success

Example Value | Model

```
{
  "model_id": "string",
  "train_acc": 1,
  "valid_acc": 1
}
```

400	Invalid input

GET /models/{model_id} Gets a model with a given ID

Parameters Try it out

Name	Description
model_id * required string (path)	The model ID model_id

Responses Response content type: application/json

Code	Description
200	Success

Example Value | Model

```
{
  "model_id": "string",
  "train_acc": 1,
  "valid_acc": 1,
  "model_class": "logistic",
  "score_func": "f_regression",
  "num_features": 1,
  "k": 1
}
```

400	Invalid input
404	Model does not exist

DELETE /models/{model_id} Deletes a model with a given ID

Parameters Try it out

Name	Description
model_id * required string (path)	The model ID model_id

Responses Response content type: application/json

Code	Description
204	Success
400	Invalid input
404	Model does not exist

POST /models/{model_id}/predict Predicts the success of an applicant using a given model

Parameters Try it out

Name	Description
payload * required object (body)	

Example Value | Model

```
{
  "father_edu": 1,
  "mother_job": "teacher",
  "father_job": "teacher",
  "reason": "home",
  "guardian": "mother",
  "travel_time": 1,
  "study_time": 1,
  "failures": 1,
  "school_support": "yes",
  "family_support": "yes",
  "paid": "yes",
  "activities": "yes",
  "nursery": "yes",
  "higher": "yes",
  "internet": "yes",
  "romantic": "yes",
  "family_rel": 1,
  "free_time": 1,
  "going_out": 1,
  "workday_alcohol": 1,
  "weekend_alcohol": 1,
  "health": 1,
  "absences": 93
}
```

Parameter content type: application/json

model_id * required string (path)	The model ID model_id

Responses Response content type: application/json

Code	Description
200	Success

Example Value | Model

```
{
  "model_id": "string",
  "success": true
}
```

400	Invalid input
404	Model does not exist

Models

TrainMetadata {

model_class*	string title: Model class example: logistic The name of the model class Enum: ["0": logistic, linear]
score_func*	string title: Score function example: f_regression The score function to use to select features Enum: ["0": f_regression, mutual_info_regression, f_classif, mutual_info_classif, chi2]
num_features*	integer title: Number of features minimum: 1 maximum: 41 The number of features to select for training
k*	integer title: K-fold cross validation minimum: 2 Value used in K-fold cross validation

}

ModelMetadata {

model_id*	string title: Model ID The ID of the created model
train_acc*	number title: Training accuracy minimum: 0 maximum: 1 Model accuracy tested on training set
valid_acc	number title: Validation accuracy minimum: 0 maximum: 1 Model accuracy tested on validation set
model_class*	string title: Model class example: logistic The name of the model class Enum: ["0": logistic, linear]
score_func*	string title: Score function example: f_regression The score function to use to select features Enum: ["0": f_regression, mutual_info_regression, f_classif, mutual_info_classif, chi2]
num_features*	integer title: Number of features minimum: 1 maximum: 41 The number of features to select for training
k*	integer title: K-fold cross validation minimum: 2 Value used in K-fold cross validation

}

TrainResult {

model_id*	string title: Model ID The ID of the created model
train_acc*	number title: Training accuracy minimum: 0 maximum: 1 Model accuracy tested on training set
valid_acc	number title: Validation accuracy minimum: 0 maximum: 1 Model accuracy tested on validation set

}

Applicant {

school*	string title: School example: GP Student's school (GP - Gabriel Pereira, MS - Mousinho da Silveira) Enum: ["0": GP, MS]
sex*	string title: Sex example: F Student's sex (F - female, M - male) Enum: ["0": F, M]
age*	integer title: Age minimum: 15 maximum: 22 Student's age
address*	string title: Address example: U Student's home address type (U - urban, R - rural) Enum: ["0": U, R]
family_size*	string title: Family size example: LE3 Family size (LE3 - <=3, GT3 - >3) Enum: ["0": LE3, GT3]
p_status*	string title: P status example: T Parent's cohabitation status (T - living together, A - apart) Enum: ["0": T, A]
mother_edu*	integer title: Mother's education minimum: 0 maximum: 4 Mother's education level (0 - none, 1 - 4th grade, 2 - 5-9th grade, 3 - secondary, 4 - higher)
father_edu*	integer title: Father's education minimum: 0 maximum: 4 Father's education level (0 - none, 1 - 4th grade, 2 - 5-9th grade, 3 - secondary, 4 - higher)
mother_job*	string title: Mother's job example: teacher Mother's job (teacher, health care related, civil services (e.g. administrative or police), at_home or other) Enum: ["0": teacher, health, services, at_home, other]
father_job*	string title: Father's job example: teacher Father's job (teacher, health care related, civil services (e.g. administrative or police), at_home or other) Enum: ["0": teacher, health, services, at_home, other]
reason*	string title: Reason example: home Reason to choose this school (close to home, school reputation, course preference or other) Enum: ["0": home, reputation, course, other]
guardian*	string title: Guardian example: mother Student's guardian (mother, father, or other) Enum: ["0": mother, father, other]
travel_time*	integer title: Travel time minimum: 1 maximum: 4 Home to school travel time (1 - <15 min, 2 - 15-30 min, 3 - 30-60 min, 4 - >60 min)
study_time*	integer title: Study time minimum: 1 maximum: 4 Weekly study time (1 - <2 hrs, 2 - 2-5 hrs, 3 - 5-10 hrs, 4 - >10 hrs)
failures*	integer title: Failures minimum: 0 maximum: 4 Number of past class failures
school_support*	string title: School support example: yes Extra educational support (yes or no) Enum: ["0": yes, no]
family_support*	string title: Family support example: yes Family educational support (yes or no) Enum: ["0": yes, no]

```
    paid*                   string
                            title: Paid
                            example: yes

                            Extra paid classes within the course subject (yes or no)

                            Enum:

                             [ yes, no ]
    activities*             string
                            title: Activities
                            example: yes

                            Participated in extra-curricular activities (yes or no)

                            Enum:

                             [ yes, no ]
    nursery*                string
                            title: Nursery
                            example: yes

                            Attended nursery school (yes or no)

                            Enum:

                             [ yes, no ]
    higher*                 string
                            title: Higher
                            example: yes

                            Wants to take higher education (yes or no)

                            Enum:

                             [ yes, no ]
    internet*               string
                            title: Internet
                            example: yes

                            Internet access at home (yes or no)

                            Enum:

                             [ yes, no ]
    romantic*               string
                            title: Romantic
                            example: yes

                            With a romantic relationship (yes or no)

                            Enum:

                             [ yes, no ]
    family_rel*             integer
                            title: Family relationship
                            minimum: 1
                            maximum: 5

                            Quality of family relationship

    free_time*              integer
                            title: Free time
                            minimum: 1
                            maximum: 5

                            Free time after school

    going_out*              integer
                            title: Going out amount
                            minimum: 1
                            maximum: 5

                            Going out with friends

    workday_alcohol*        integer
                            title: Alcohol (workday)
                            minimum: 1
                            maximum: 5

                            Workday alcohol consumption

    weekend_alcohol*        integer
                            title: Alcohol (weekend)
                            minimum: 1
                            maximum: 5

                            Weekend alcohol consumption

    health*                 integer
                            title: Health
                            minimum: 1
                            maximum: 5

                            Current health status

    absences*               integer
                            title: Absences
                            minimum: 0
                            maximum: 93

                            Number of school absences

}
```

PredictionResult {

```
    model_id*               string
                            title: Model ID

                            The ID of the model used to make prediction

    success*                boolean
                            title: Predicted success

                            The success of the given applicant predicted by the model

}
```