# Модуль 2. Работа с ClearML.

## В этом модуле

Вы продолжите изучение ClearML и примените уже известные вам функции ClearML для решения практической задачи машинного обучения из соревнования с площадки kaggle.com.

Многие специалисты в области данных и машинного обучения участвуют в соревнованиях и хакатонах, поскольку это хорошая возможность отточить свои навыки на учебных (и не только) задачах, узнать новые методы и подходы к решению, познакомиться с экспертами и другими командами и заявить о себе в профессиональном сообществе. К получению медалей и статуса грандмайстера на kaggle.com стремятся даже опытные специалисты, ведь это свидетельствует о высокой квалификации эксперта. Конечно, между соревновательным ML и производственными задачами очень много различий, однако есть и общие черты, в том числе касающиеся MLOps. В следующих юнитах 8-11 вы:

* разберете на практике, как настроить окружение ClearML для небольшой команды, решающей практическую задачу конкурса «Tabular Playground Series» с площадки kaggle.com, используя уже известные вам возможности clearml,
* в заключительном юните найдете описание задания для самостоятельной работы, выполнение которого потребует описанных в модуле навыков.

## Содержание юнитов

|  |
| --- |
| 1. Постановка задачи и подготовка окружения для работы |
| 1. Получение и обработка данных с clearml-data |
| 1. Управление экспериментами |
| 1. Сборка общего пайплайна, оценка результатов и загрузка итогового файла |
| 1. Итоговое задание по модулю |

# Модуль 2. Юнит 8. Постановка задачи и настройка окружения для работы.

## Введение

В этом юните мы сформулируем задачу и дадим указания о настройке ClearML окружения для командной работы над ней.

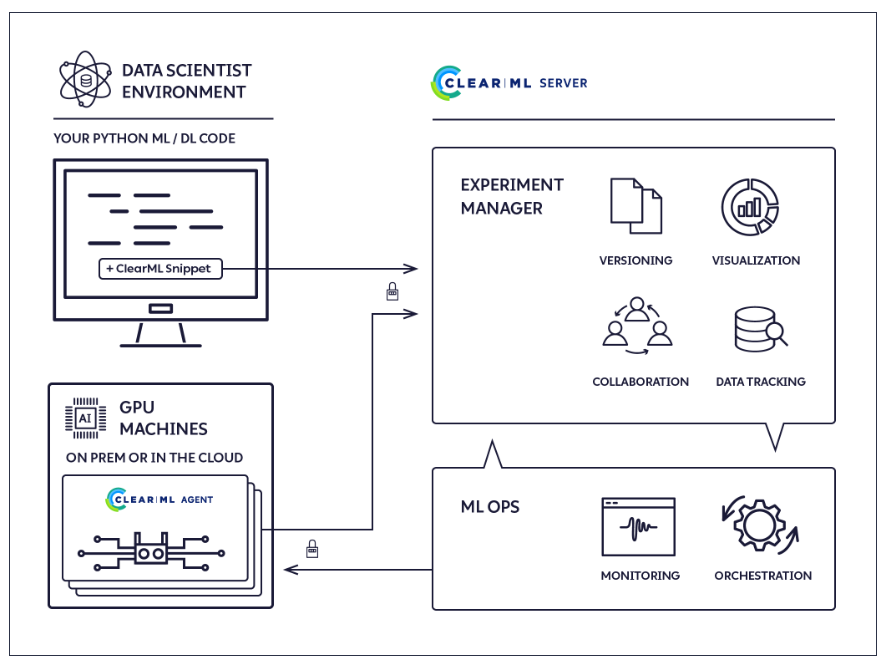
## Содержание

### Описание целевой инфраструктуры и установка программного обеспечения

Для решения практической задачи необходимо создать и настроить основные элементы инфраструктуры ClearML, в нашем случае мы будем пользоваться возможностями облачного сервиса app.clear.ml.

Напомним, основные элементы инфраструктуры ClearML:

* ClearML сервер;
* Пакет clearml, устанавливаемый с помощью pip и содержащий необходимые исполняемые скрипты и библиотеки для интеграции python кода в clearml;
* Clearml-agent для управления конвейерами операций, в том числе вычислениями.

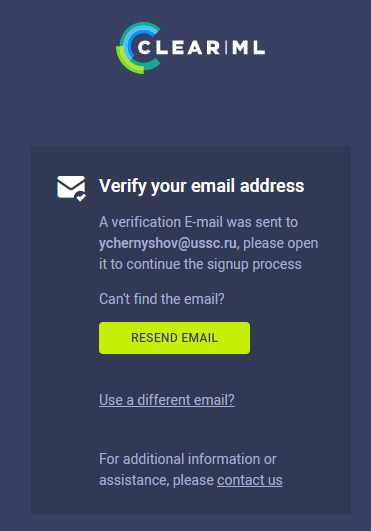


Описание основных элементов инфраструктуры ClearML (схема с официального сайта https://clear.ml/docs/latest/docs).

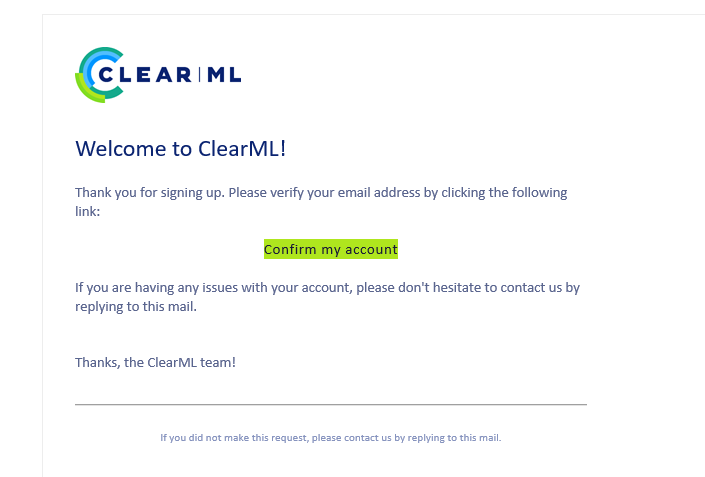
Модули, входящие в состав ClearML

* [ClearML Python Package](https://clear.ml/docs/latest/docs/getting_started/ds/ds_first_steps#install-clearml) (clearml) предназначен для интеграции ClearML в программный код проекта,
* [ClearML Server](https://clear.ml/docs/latest/docs/deploying_clearml/clearml_server) (clearml-server) для сохранения экспериментов, моделей и конвейеров обработки данных, в том числе с использованием удобного графического web-интерфейса. Выполняет роль «панели управления» для MLOps.
* [ClearML Agent](https://clear.ml/docs/latest/docs/clearml_agent) (clearml-agent), выполняет оркестрацию операций MLOps, обеспечивает повторяемость и масштабируемость экспериментов и конвейеров операций.
* [ClearML Data](https://clear.ml/docs/latest/docs/clearml_data/) (clearml-data) обеспечивает управление и версионирование данных.
* [ClearML Serving](https://clear.ml/docs/latest/docs/clearml_serving/) (clearml-serving) — управление моделями, оркестрация и развертывание.
* [ClearML Session](https://clear.ml/docs/latest/docs/apps/clearml_session) (clearml-session) — для запуска удаленных экземпляров Jupyter Notebooks и VSCode. Комбинируется с управляющей панелью clearml-server.

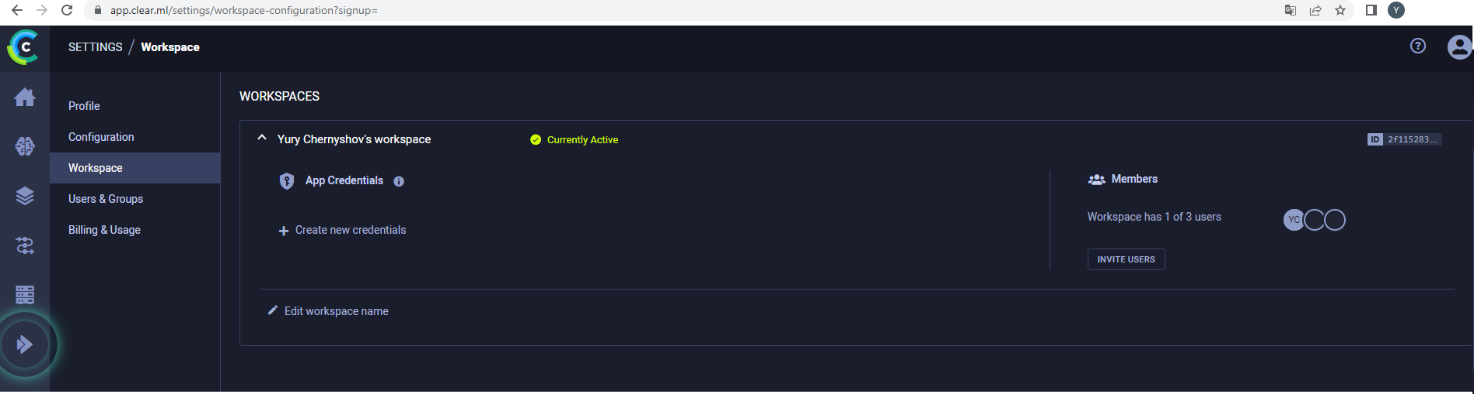
В юнитах 8-11 далее мы будем использовать облачную версию ClearML, которая доступна в Интернет по ссылке <https://app.clear.ml/settings/workspace-configuration>. При открытии этой страницы в браузере система попросит зарегистрироваться или войти под уже зарегистрированным пользовательским именем. При регистрации после заполнения интуитивно понятных данных вы получите сообщение о том, что на ваш адрес электронной почты направлено письмо для подтверждения регистрации.



Письмо с запросом подтверждения регистрации



Кликнув на ссылку «Confirm my account» вы завершите регистрацию, после чего можно открыть пользовательский интерфейс в app.clear.ml, на котором появится ваш пользовательский профиль.



Теперь облачный сервис ClearML готов к настройке для организации работы вашей команды исследователей.

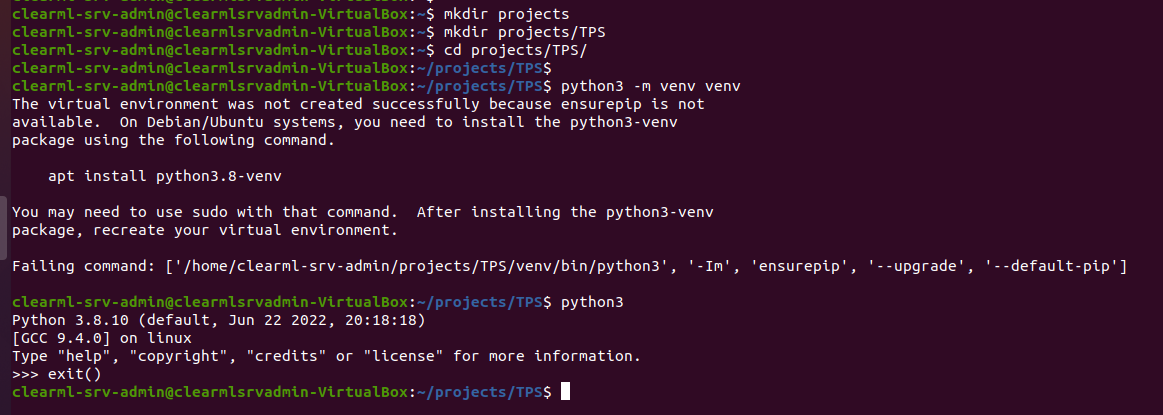
### Постановка задачи и исходные данные

В юнитах 8-11 мы совместно настроим ClearML для работы команды, решающей задачу с площадки соревнований по машинному обучению kaggle.com. Это самая популярная соревновательная площадка, на которой можно найти как учебные задачи, так и конкурсы с солидным денежным призом. Почти все начинающие исследователи данных и специалисты ML начинают с задачи «Titanic Disaster», уже известной вам по предыдущим модулям — это своего рода «Hello, world» в машинном обучении и, конечно же, в Интернете можно найти точные ответы на эту задачу. После того, как задача «Titanic Disaster» оказалась «зарешена до дыр», и по понятным причинам с первой же попытки новые «участники» конкурса «Titanic Disaster» получали 100% результат на неизвестной (закрытой) проверочной выборке, модераторы площадки kaggle.com решили каждый месяц публиковать тренировочный конкурс «Tabular Playground Series», призванного заменить задачу «Titanic Disaster». По замыслу создателей эти соревнования хорошо подходят начинающим исследователям данных, поскольку проще, чем реальные соревнования на kaggle.com, но не такие простые и изученные, как «Titanic Disaster». Опытным участникам соревнований kaggle.com такой формат будет не интересен, создатели «Tabular Playground Series» просят их не занимать строчки в leaderboard, а дать дорогу новичкам. Над задачами могут работать исследовательские команды, но число участников одной команды должно быть не более трех. Каждый месяц в этом соревновании появляется новая задача. Например, задача ноября 2022 года находится тут: <https://www.kaggle.com/competitions/tabular-playground-series-nov-2022>. Предлагаем вам в следующих юнитах объединиться в команды и организовать исследование данных и эксперименты с моделями машинного обучения для того, чтобы попасть в верхние строчки leaderboard конкурса «Tabular Playground Series» ноября 2022, следуя инструкциям в следующих юнитах 8-11, а затем самостоятельно применить свои знания на задании «Tabular Playground Series» от другого месяца 2022 года (по вашему выбору) в командах до трех человек.

Каждое соревнование на kaggle.com имеет свой датасет (открытую часть для обучения и закрытую часть для оценки предоставленных решений и формирования leaderboard), детально описанную задачу, метрику для оценки. Также часто присутствует baseline-решение в виде jupyter-ноутбука. В задаче «Tabular Playground Series» ноября 2022 даны результаты работы различных моделей бинарной классификации в 5000 CSV файлах (общий размер 2.9 Gb) и правильная разметка train\_labels.csv для первой половины строчек файлов. Необходимо, используя блендинг (blending) полученных результатов, постараться улучшить общий результат. Пример построения бейзлайна (baseline) с помощью среднего арифметического оценок можно посмотреть тут: <https://www.kaggle.com/code/inversion/simple-blending-example/notebook.> То, как работает блендинг (смешивание) моделей машинного обучения, описано во многих источниках. Например, в простой форме с примерами можно найти описание тут: [https://habr.com/ru/company/skillfactory/blog/531416](https://habr.com/ru/company/skillfactory/blog/531416/).

### Настройка рабочей инфраструктуры

Будем работать с виртуальной машиной в virtualbox, на которую установим ОС Ubuntu 20.04. Создадим рабочую папку проекта ./projects/TPS (это сокращение от «Tabular Playground Series») и виртуальное окружение venv для установки необходимых библиотек.

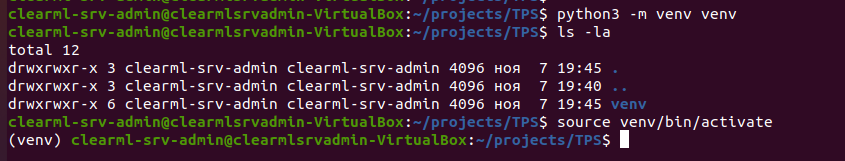


Если модуль venv не установлен, это можно решить командой

**sudo apt install python3.8-venv**

Она пригодится, если у вас используется python версии 3.8. Еще вы можете подобрать установку для вашей конкретной версии интерпретатора python.

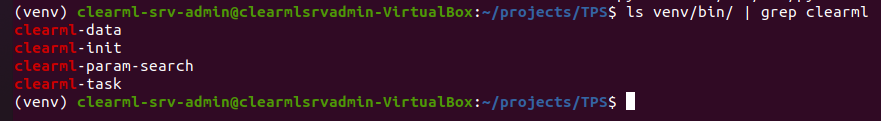
После этого создаем папку виртуального окружения и запускаем его



Теперь давайте в этом виртуальном окружении установим пользовательское окружение ClearML с помощью команды

**pip install clearml**

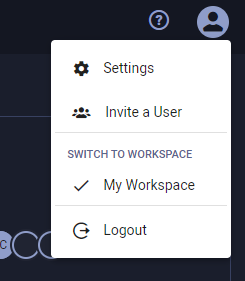
Кроме того, вы можете посмотреть новые добавленные исполняемые скрипты clearml, добавленные в папку venv/bin



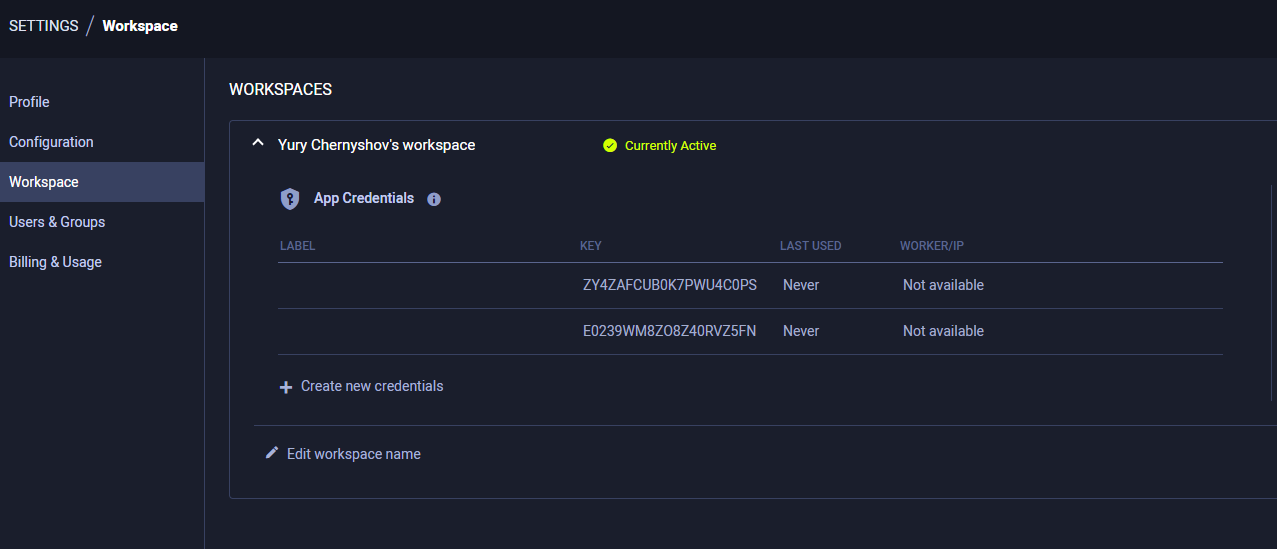
Назначение этих скриптов-команд

* clearml-init — присоединение к серверу,
* clearml-data — работа с данными,
* clearml-param-search — поиск параметров,
* clearml-task — работа с задачами.

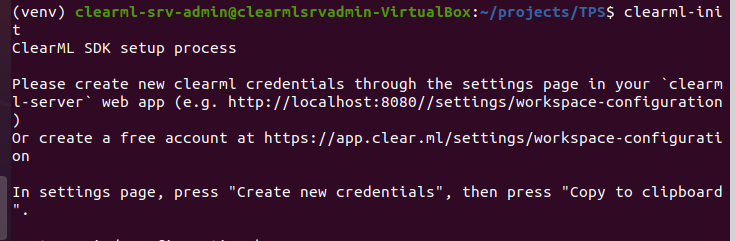
Давайте подключимся к настроенному нами облачному серверу app.clear.ml, для этого перейдем во вкладке «Профиль» в «Settings» (настройки)



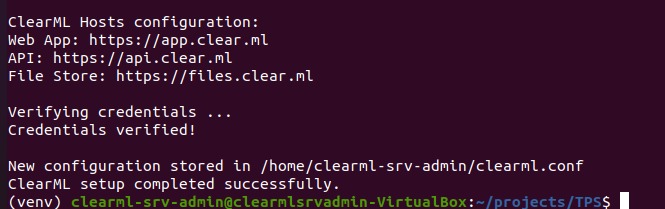
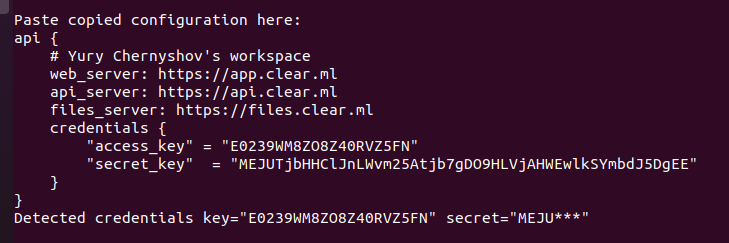
после чего во вкладке «Workspace» выберем «Create new credentials»



И в открывшемся окне скопируем данные, которые нужно будет записать в консоль рабочей машины после вызова clearml-init.



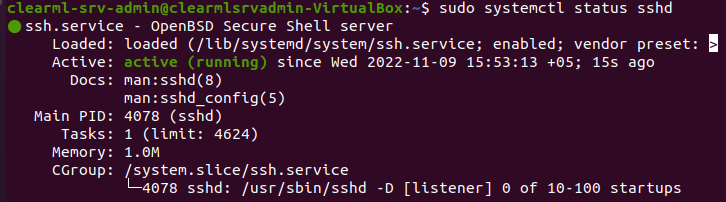
После добавления этих данных рабочая машина подключается к облачному сервису app.clear.ml



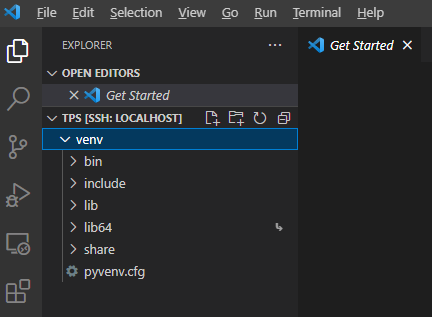
Далее вам понадобится писать и редактировать программный код, поэтому подключите редактор VSCode к виртуальной машине через прокинутый сетевой порт virtualbox с помощью SSH, как мы это делали ранее в предыдущих юнитах. Не забудьте установить ssh сервер на виртуальную машину с помощью команды

**sudo apt-get install openssh-server**

и проверить его текущий статус



В итоге рабочая папка в Visual Studio Code проекта должна выглядеть так:

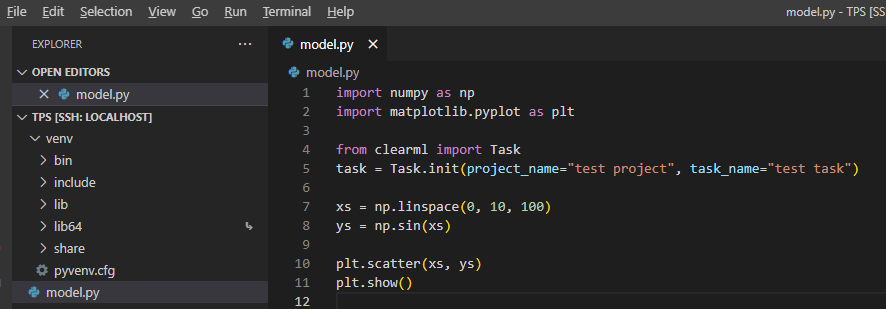


Теперь сюда мы можем добавлять код. Pip пакет clearml позволяет сделать python код видимым в ClearML сервере с помощью «двух строчек кода». Вот эти строчки, добавляемые в код python скрипта:

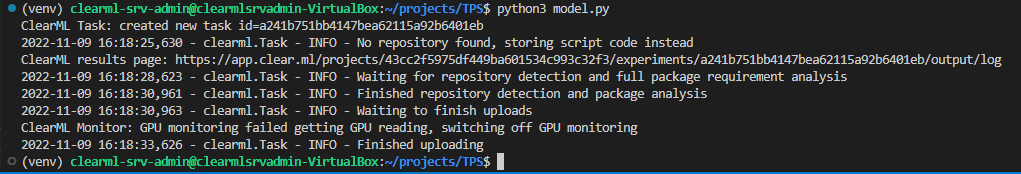
**from clearml import Task**

**task = Task.init(project\_name=”имя проекта”, task\_name=”имя задачи”)**

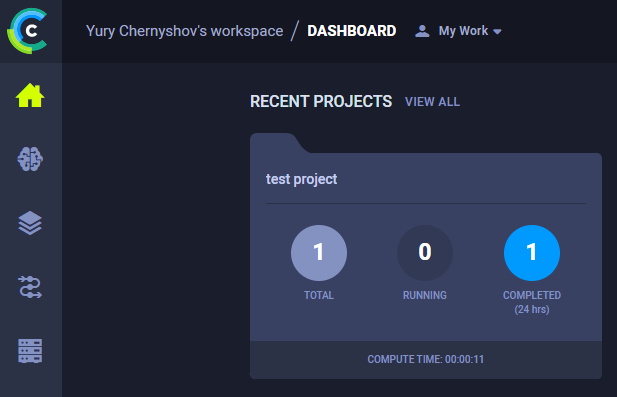
Давайте создадим простой скрипт, возможно, для его выполнения вам понадобится установить в виртуальном окружении venv соответствующие библиотеки



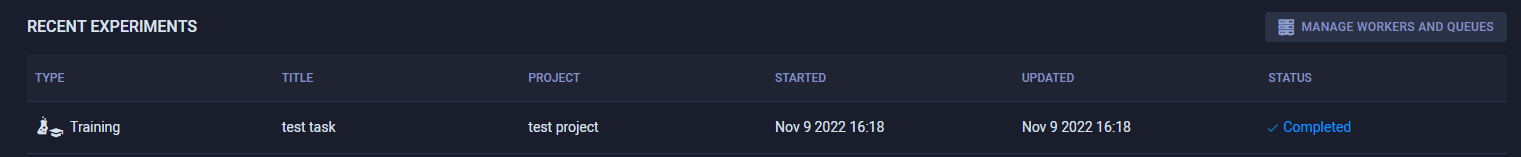
После выполнения этого скрипта интерпретатором вы можете видеть, что данный код теперь попал в поле зрения облачного сервиса clearml



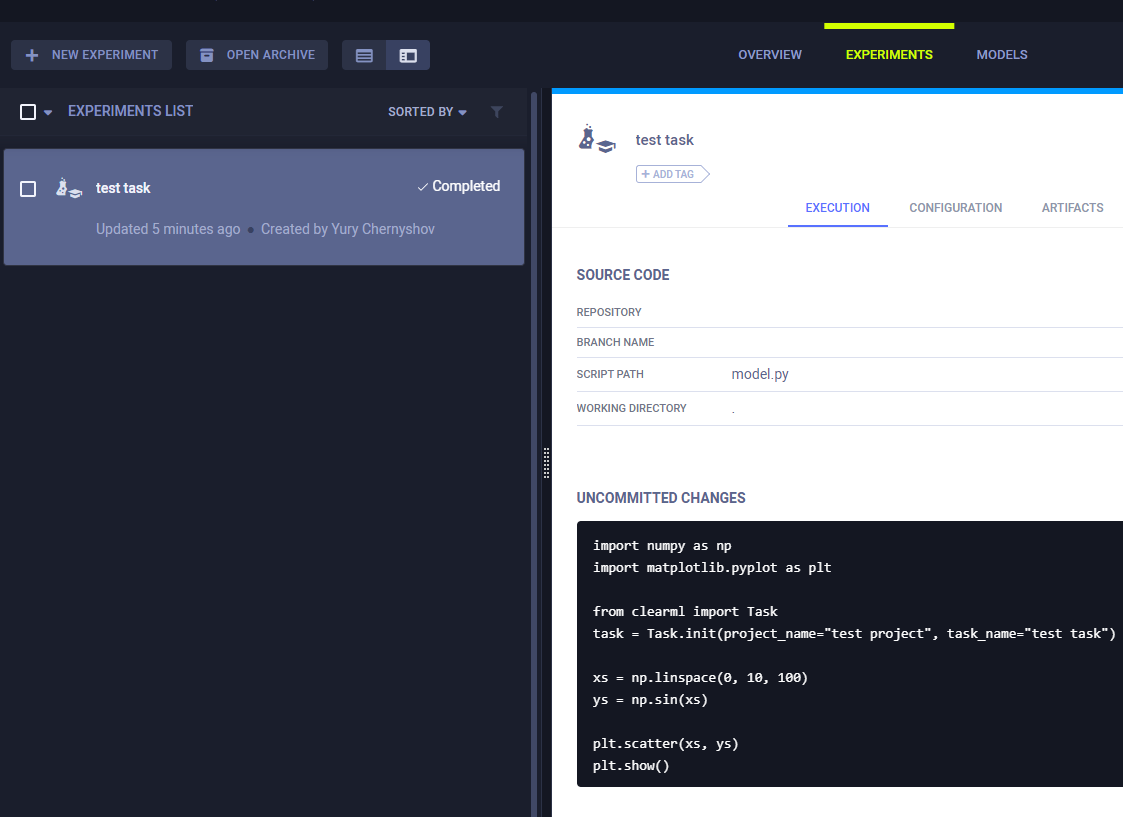
А именно, во вкладке «Dashboard**»** появилась информация о созданном вами проекте



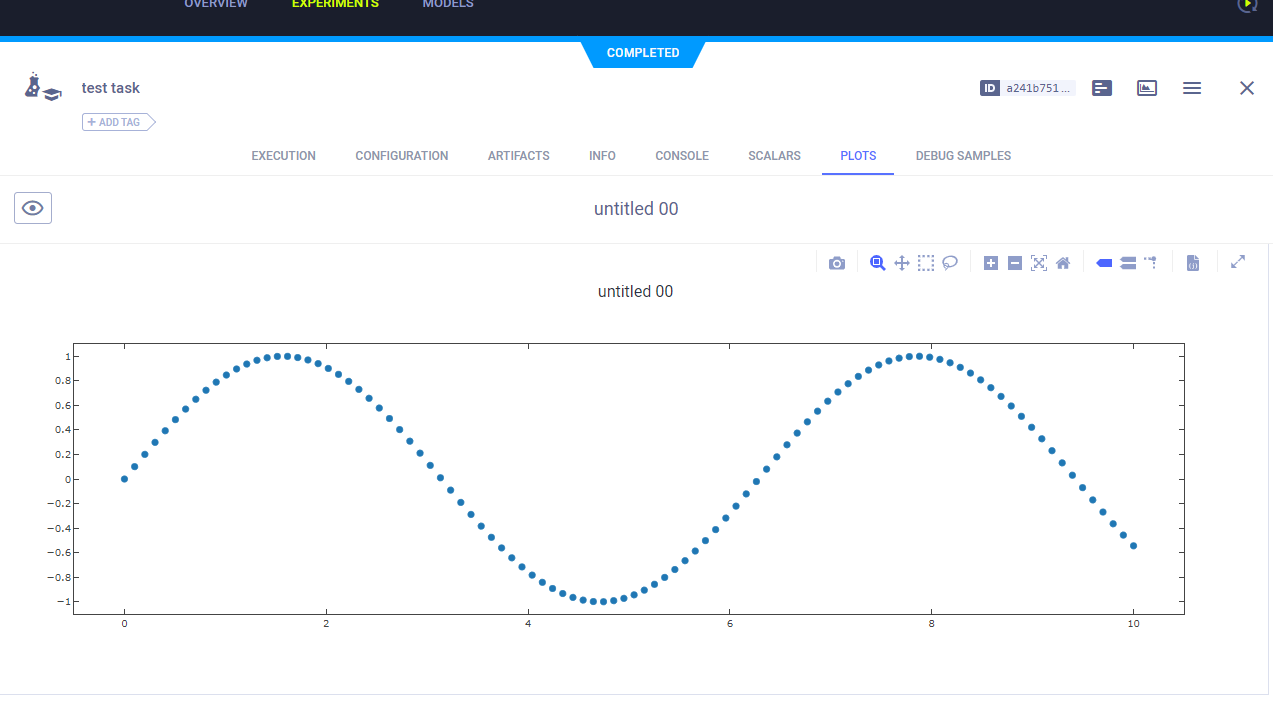
Также вы можете видеть результаты выполнения этого проекта



И детали его содержания, например, код скрипта



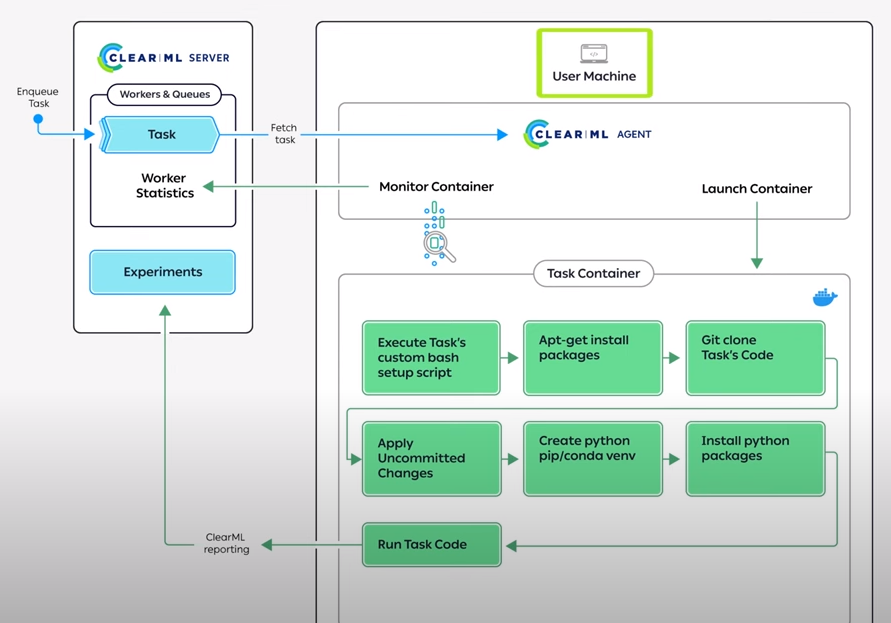
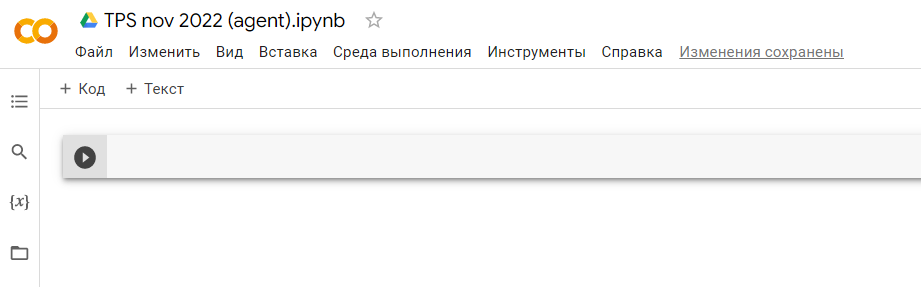
Также, поскольку ClearML сразу берет под наблюдение вывод стандартных библиотек, он автоматически запоминает графики, которые строит библиотека matplotlib



Также вы можете добавлять под наблюдение clearml числовые параметры (scalars), результаты работы программ (артефакты, artifacts), контролировать конфигурацию. Очень многие настройки и свойства вашего проекта clearml обнаруживает самостоятельно, «из коробки», что очень удобно.

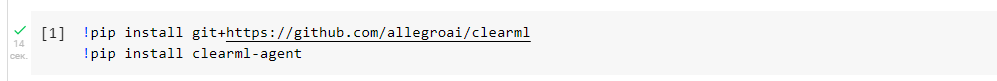
Для управления конвейерами машинного обучения, в том числе для выполнения вычислений, вам понадобится clearml-agent. Давайте используем в качестве вычислительного ресурса clearml-agent вычислительные мощности google colab, подробная инструкция для этого приведена на официальном сайте <https://clear.ml/docs/latest/docs/guides/ide/google_colab.>

Сначала создадим jupyter ноутбук в google colab ()



В итоге вы получили вычислительные мощности для проведения экспериментов на google colab. Бесплатный вариант использования google colab предполагает ограничения на используемые ресурсы, например, вы не сможете использовать нужную вам модель GPU. Однако для учебных примеров и экспериментов, как наш, этого ресурса вполне достаточно.

В соответствии с инструкцией выполним первую ячейку, в которой устанавливаются необходимые пакеты

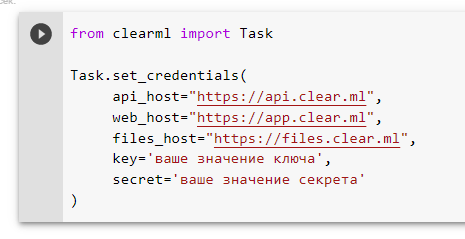


Затем во второй ячейке экспортируем значение системной переменной

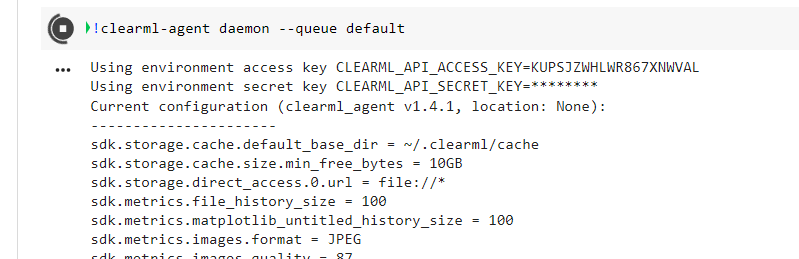


Далее в облачном сервисе app.clear.ml создадим новое подключение. Для этого на странице

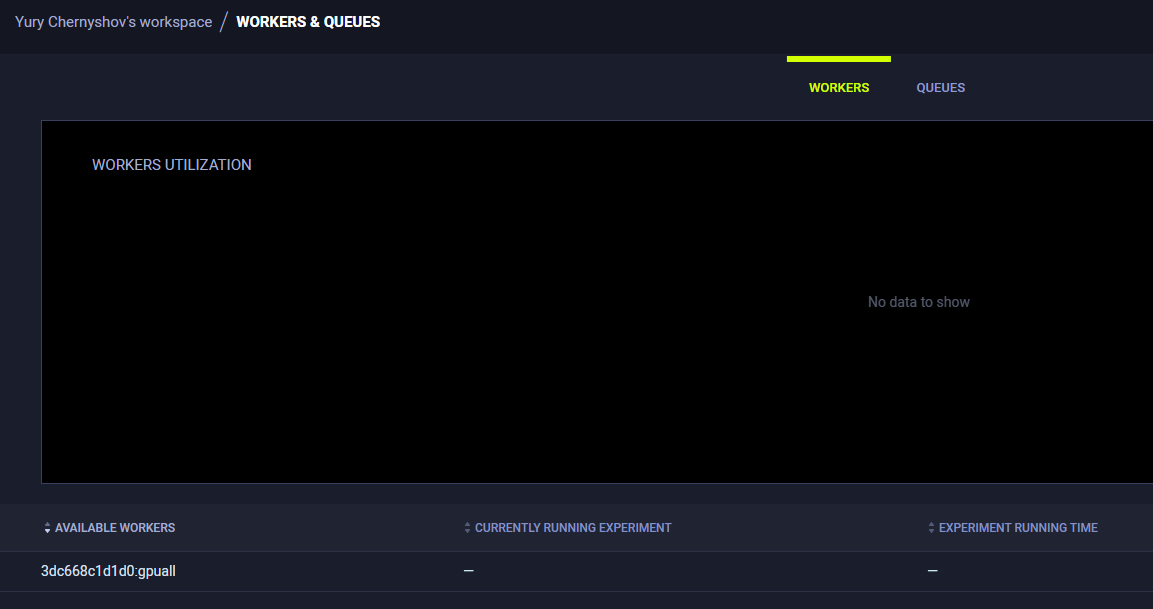
«[Settings](https://app.clear.ml/settings/workspace-configuration)» надо выбрать раздел «WORKSPACE» и создать параметры для нового подключения, нажав на кнопку «Create new credentials», после чего эти параметры можно скопировать.



Теперь вы можете запустить clearml-agent демона (программа, выполняющаяся в фоновом режиме) в очереди выполнения по умолчанию (queue default)

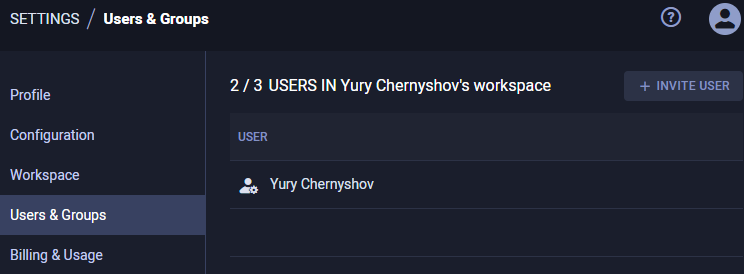


После выполнения этой ячейки соответствующий воркер (worker) появляется в разделе **«**[Workers & Queues](https://clear.ml/docs/latest/docs/webapp/webapp_workers_queues)**»** на сервере app.clear.ml.

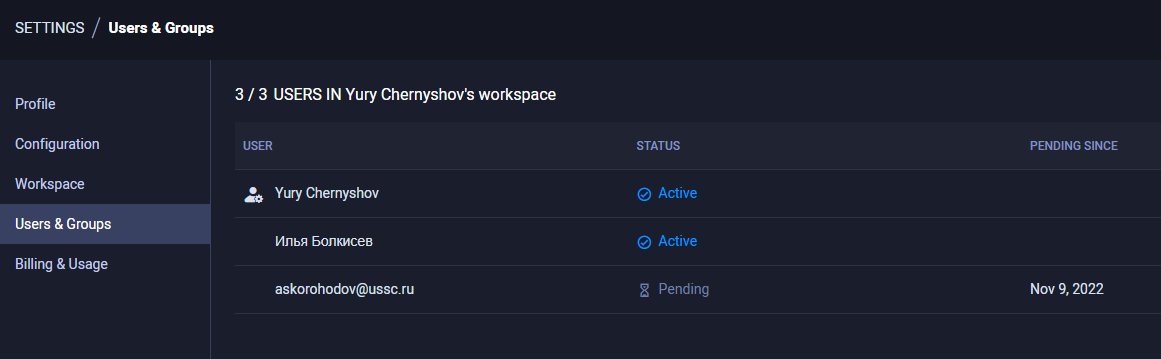


Теперь можно клонировать эксперимент и указать соответствующий ресурс для выполнения эксперимента, при этом при выполнении кода будут задействованы ресурсы google colab.

И, наконец, давайте добавим участников проекта, которые будут участвовать в исследовании данных и проведении экспериментов с моделями машинного обучения. В соревнованиях kaggle.com чем больше участников, тем лучше для проекта, поскольку соревнования предполагают множество экспериментов, а большая команда может проверить больше идей и гипотез. Поэтому в соревнованиях ограничивают количество участников и публикаций результатов. В соревновании «Tabular Playground Series nov 2022» команда может состоять из трех человек. Совершенно случайно это ограничение совпадает с ограничением бесплатной облачной версии ClearML, в облачном сервисе ClearML вы можете добавлять до трех участников в проект. Для этого надо всем участникам зарегистрироваться в сервисе app.clear.ml, а вам добавить участников с помощью кнопки «Invite user»



После этого вы будете видеть всех участников проекта на вкладке «Users&Groups». Пока пользователь не откроет направленную вами ссылку-приглашение он будет в списке пользователей в статусе «Pending», а после открытия ссылки и подтверждения своего участия в проекте пользователь перейдет в статус «Active»



Теперь у вас есть необходимая инфраструктура для организации работы небольшой (до трех человек) команды машинного обучения для решения задачи Tabular Playground Series nov 2022.

## Тест

1. Какие исполняемые файлы добавились при установке clearml? (0.25)
   1. **clearml-data**
   2. clearml-pipeline
   3. **clearml-init**
   4. **clearml-task**
2. Какую задачу вы будете решать в следующих юнитах данного модуля? (0.25)
   1. **Tabular Playground Series November 2022**
   2. Titanic Disaster
   3. MNIST digits recognition
   4. Предсказание стоимости квартир в Бостоне
3. Какая команда устанавливает clearml? (0.25)
   1. apt install clearml
   2. **pip install clearml**
   3. apt-get install clearml
   4. clearml/setup.py
4. Какая команда используется для подключения к облачному сервису ClearML? (0.25)
   1. clearml-start
   2. **clearml-init**
   3. clearml-connect
   4. clearml-cloud

## Итоги/выводы

В этом юните вы создали необходимую инфраструктуру ClearML для использования в следующих юнитах данного модуля, в том числе:

* зарегистрировали аккаунты в облачном сервисе app.clear.ml и настроили его для работы,
* установили pip пакет clearml,
* создали среду для разработки необходимого программного кода с использованием virtualbox, VSCode,
* настроили clearml-agent для выполнения пайплайна операций,
* добавили пользователей в команду.

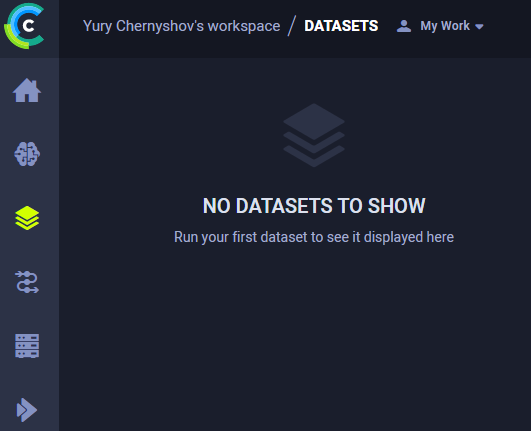
# Модуль 2. Юнит 9. Получение и обработка датасета с clearml-data.

## Введение

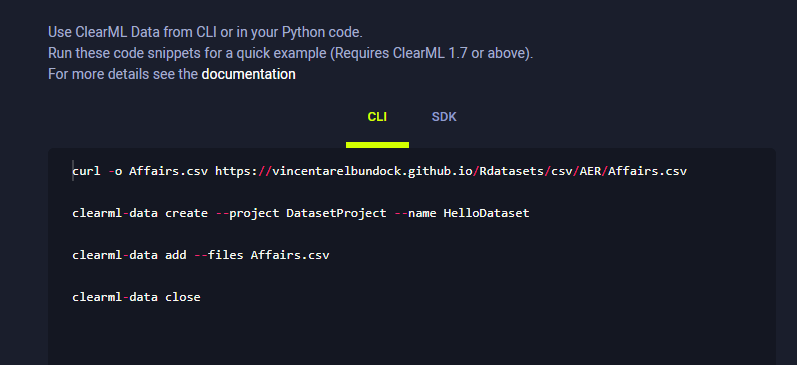
Данные — это важнейшая часть проекта машинного обучения. В соревнованиях на платформе kaggle.com данные являются частью конкурса, участники соревнуются между собой — в том числе, в умении обрабатывать данные и создавать на их основе наиболее эффективные признаки (features). В этом юните вы научитесь интегрировать датасеты из соревнований с площадки kaggle.com в инфраструктуру вашей команды с использованием ClearML.

## Содержание

В панели управления сервиса app.clear.ml вы можете через вкладку Datasets получить доступ для работы с наборами данных (датасетами). По умолчанию в проекте ClearML датасетов нет, поэтому сначала вы увидите уведомление *No datasets to show*.



На этой же вкладке есть пример простого кода, добавив который в код проекта вы сможете автоматизировать получение данных из внешнего источника и добавить данные в проект. Это можно сделать с помощью консольных команд (исполняемых скриптов), пример которых приведен на вкладке CLI:



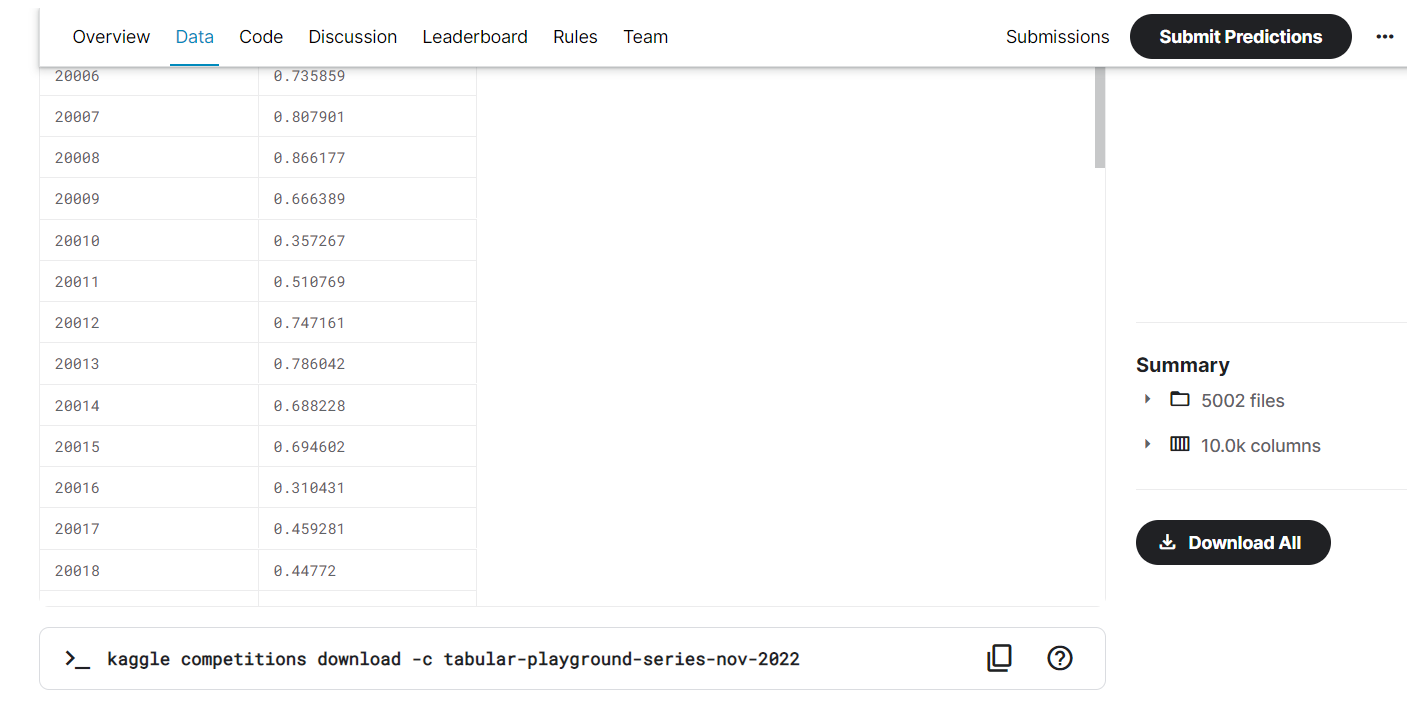
Другой, более функциональный и гибкий вариант загрузки данных в ClearML состоит в использовании SDK (Software Development Kit), набора команд, которые можно выполнять в python скрипте, импортировав специальную библиотеку. Пример такого скрипта и синтаксиса команд вы также можете найти на вкладе Datasets в ClearML в разделе SDK:



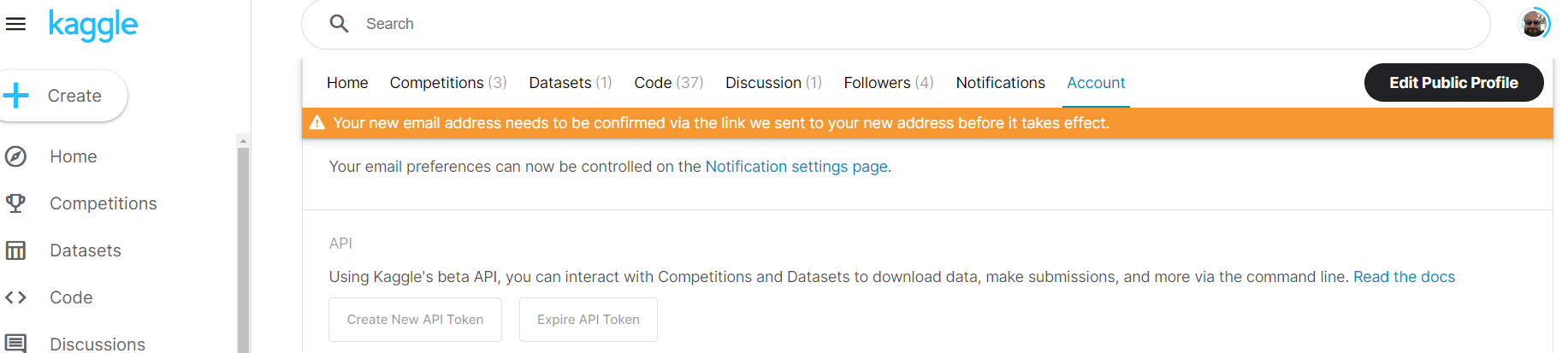
В нашем случае мы будем работать с данными с площадки kaggle.com, которая содержит API для автоматизированного взаимодействия сторонних инструментов с площадкой, в том числе и для закачки датасетов. Это библиотека с открытым исходным кодом, <https://github.com/Kaggle/kaggle-api,> ее можно установить командой в созданном вами виртуальном окружении venv

**pip install kaggle**

Затем вы загрузите данные с помощью инструкций со страницы соревнования: <https://www.kaggle.com/competitions/tabular-playground-series-nov-2022/data>



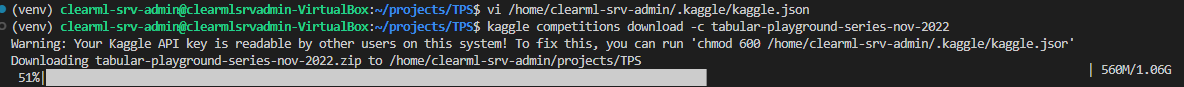
Важно, что для доступа к данным площадки вам нужно создать токен в аккаунте вашего профиля на kaggle.com (инструкция находится здесь https://github.com/Kaggle/kaggle-api).



Нажатие на кнопку *Create New API Token* приведет к созданию kaggle.json файла, который необходимо перенести в папку .kaggle вашей рабочей директории.

Изменяя соответствующим образом команды из примера ClearML по загрузке данных, вам надо подготовить вот такой скрипт, либо последовательно выполнить команды в консоли

**kaggle competitions download -c tabular-playground-series-nov-2022**



**clearml-data create --project “TPS” --name tps-dataset**



Теперь давайте добавим файлы датасета под контроль ClearML. Перед этим не забудьте разархивировать файл — данные с kaggle.com переносятся в виде архива zip. Это можно сделать командой

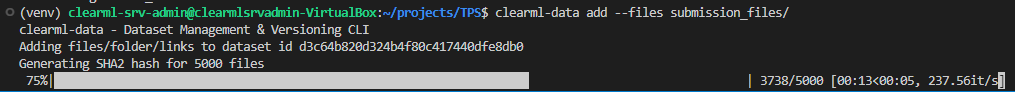
**unzip tabular-playground-series-nov-2022.zip**

Далее с помощью скрипта clearml-data эти файлы ставятся на контроль

**clearml-data add --files submission\_files/**

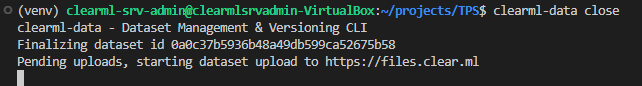
**clearml-data add --files train\_labels.csv**

**clearml-data add --files sample\_submission.csv**

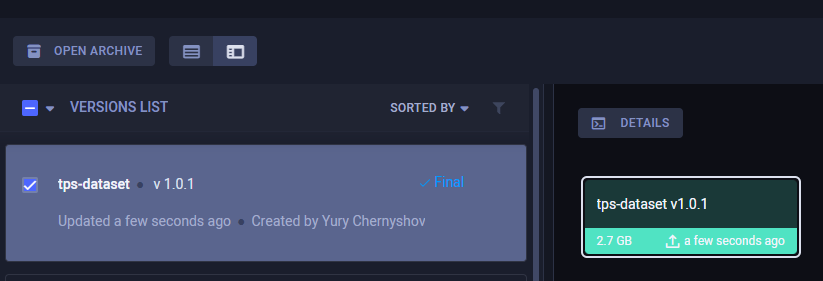


После добавления всех необходимых файлов необходимо вызвать команду

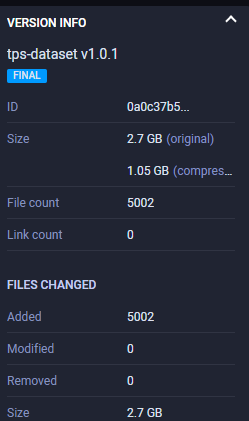
**clearml-data close**



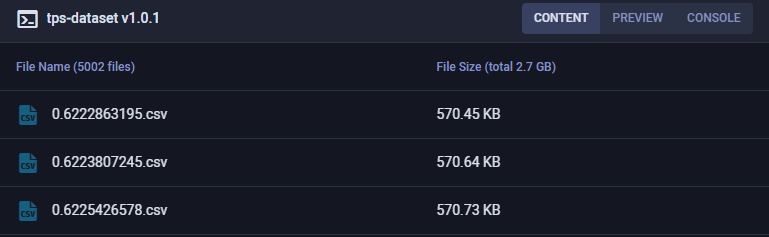
Файлы загрузятся в облачный сервис app.clear.ml, после чего этот датасет можно увидеть в графическом интерфейсе



Также ClearML позволяет получить более детальную информацию по структуре данных



И даже получить доступ к содержимому датасета через графический интерфейс



Теперь датасет задачи привязан к проекту в ClearML, это его составная часть.

## Тест

1. Какой инструмент мы использовали для загрузки датасета с площадки kaggle.com? (0.25)
   1. GPU
   2. **библиотека kaggle со специальным API**
   3. ftp
   4. ssh
2. Какие классы импортируются из SDK модуля clearml для работы с данными? (0.25)
   1. **StorageManager**
   2. Pandas
   3. **Dataset**
   4. Table
3. Какая команда используется для добавления файлов в датасет под контролем clearml? (0.25)
   1. **clearml-data add**
   2. clearml-data file
   3. clearml-data append file
   4. clearml-dataset add
4. Какая команда завершает формирование датасета на clearml? (0.25)
   1. clearml-close
   2. **clearml-data close**
   3. clearml-dataset finish
   4. clearml-data finalize

## Итоги/выводы

В этом юните вы загрузили набор данных (датасет), который необходимо использовать в проекте для решения задачи конкурса «Tabular Playground Series nov 2022» на площадке kaggle.com с помощью специальной библиотеки kaggle, предоставляющей удобный интерфейс для работы с площадкой. Далее штатными средствами ClearML вы добавили файлы датасета в проект, после чего они были загружены в облачный сервис app.clear.ml и стали частью проекта. В следующих юнитах вы проведете эксперименты с этими данными с использованием различных моделей машинного обучения.

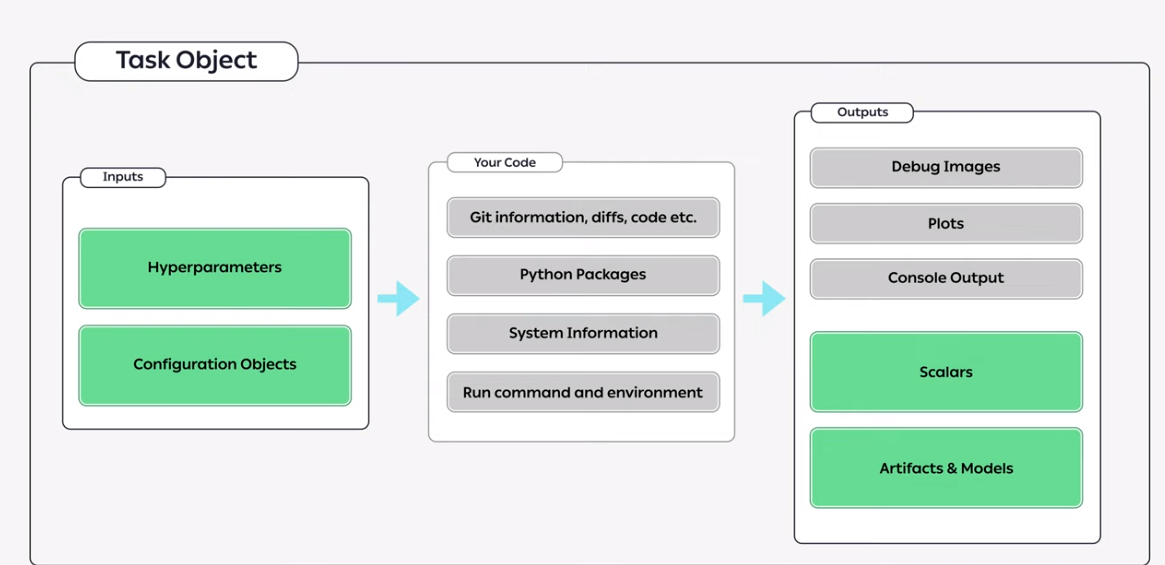
# Модуль 2. Юнит 10. Управление экспериментами.

## Введение

Вы уже знаете, что важнейшие задачи в проекте машинного обучения — воспроизводимость и сравнимость экспериментов, а также быстрое тестирование модели для оценки качества ее работы. Это же в полной мере применимо и для соревнований по машинному обучению, так как при попытке создать модель с наилучшим качеством исследователи проводят множество экспериментов, в которых легко можно запутаться без инструментов для управления этими экспериментами: формирования пайплайна, версионирования, сопоставления, быстрого возврата к удачной конфигурации, кэширования отдельных этапов. Для решения таких задач проекта в этом юните вы настроите контроль и управление экспериментами в ClearML для выполняемого проекта.

## Содержание

Основная сущность в ClearML — Task, который служит как для выполнения и управления экспериментами, так и для последующего вывода решения в продакшн.

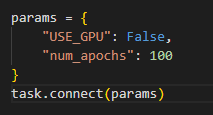


В качестве «Входов» (Inputs) Task работает с гиперпараметрами (Hyperparameters) и конфигурационными настройками (Configuration Objects). Task знает очень многое о вашем коде, причем он собирает эту информацию в автоматическом режиме, незаметно для пользователя. Данные git, информация об установленных python пакетах, системная информация, запускаемые команды и настройки среды выполнения — все это собирается с помощью стандартных утилит и сохраняется в ClearML для последующего использования, в том числе для отображения в графическом интерфейсе. К удобству получения этой информации пользователи ClearML привыкают настолько, что бывают сильно удивлены, не найдя этих инструментов контроля параметров кода и среды выполнения в других инструментах. «Выходами» (Outputs) из Task являются выводы стандартных библиотек, которые ClearML принимает и фиксирует для использования, консольный вывод, а также то, что задает пользователь в python скрипте, например, числовые значения (Scalars), результаты выполнения отдельных этапов (Artifacts), например, обученные модели ML (Models).

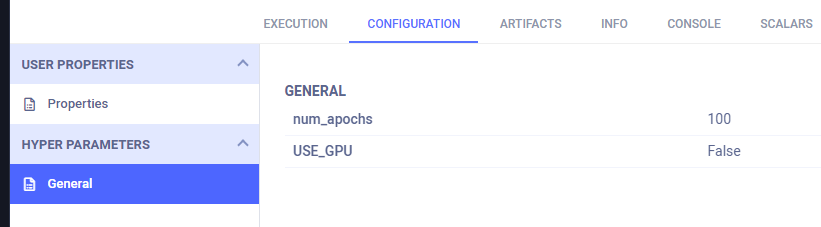
Python код становится Task после добавления в python скрипт следующих строчек:



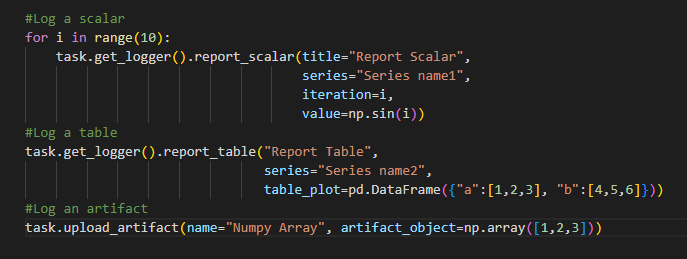
Гиперпараметры модели машинного обучения задаются перед началом обучения и затем не изменяются, в отличие от параметров, изменение которых и составляет суть машинного обучения. Удачное сочетание гиперпараметров может существенно ускорить обучение модели и/или улучшить значения метрик. ClearML предоставляет удобный инструмент для работы с гиперпараметрами. Повторение экспериментов с разными сочетаниями гиперпараметров, моделями, наборами признаков (features) — это основа машинного обучения. Поэтому при исследовании данных и экспериментах с моделями часто возникает ситуация, когда исследователь хочет повторить эксперимент другого участника проекта с другим набором гиперпараметров. Вы можете сохранить гиперпараметры пайплайна следующим образом в python коде:



После чего эти параметры будут доступны в графическом интерфейсе app.clear.ml

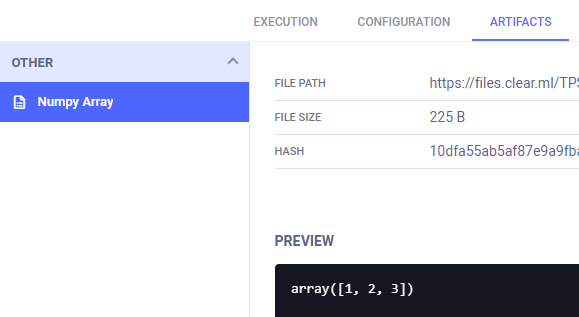


В ходе выполнения эксперимента вы можете выполнять логирование любой информации, например:



После этого загруженные артефакты будут доступны в графическом интерфейсе app.clear.ml

Артефакты:



Численные значения (scalars)

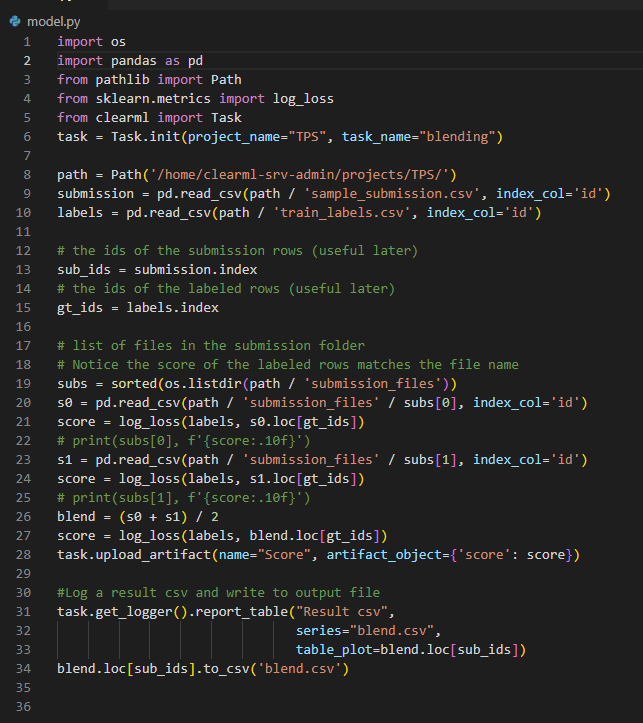


Таблица pandas



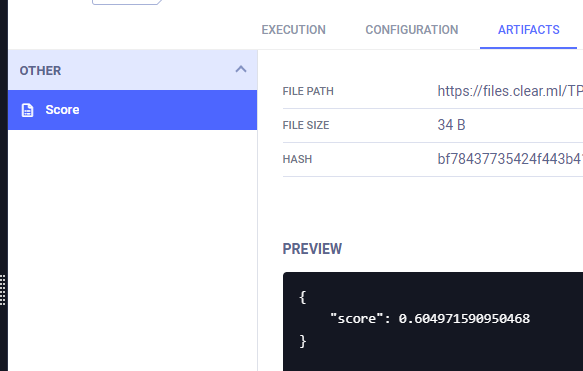
Остальные возможности для логирования в ходе эксперимента можно посмотреть в документации https://clear.ml/docs/latest/docs/guides/reporting/.

Давайте научимся создавать и повторять эксперименты в ClearML. Вот код эксперимента для задачи *TPS nov 2022*, основанный на baseline методе, опубликованном на площадке соревнования организаторами https://www.kaggle.com/code/inversion/simple-blending-example/notebook



Этот эксперимент находится под трекингом ClearML, в частности, вы можете посмотреть результат выполнения работы скрипта

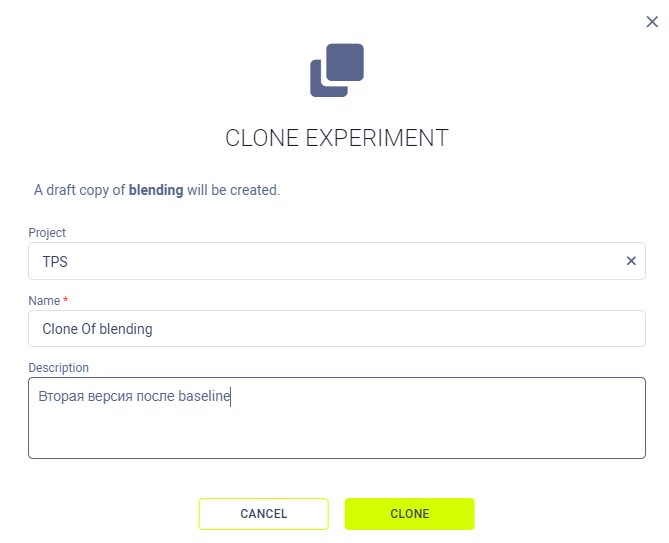
Артефакт, являющийся оценкой качества работы модели (score)



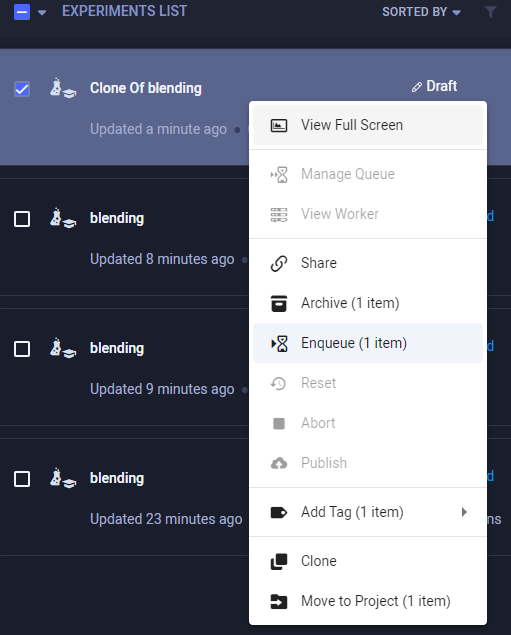
Содержимое файла blend.csv



Этот проект вы можете клонировать, для этого надо выбрать проект, в котором будет храниться эксперимент, а также задать новое имя для эксперимента. А еще можно задать описание для эксперимента.



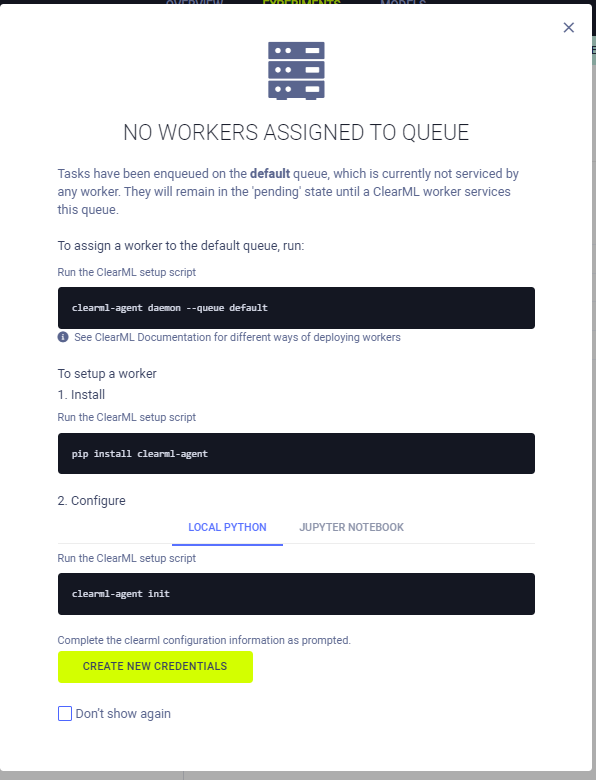
После клонирования эксперимента необходимо его запустить.



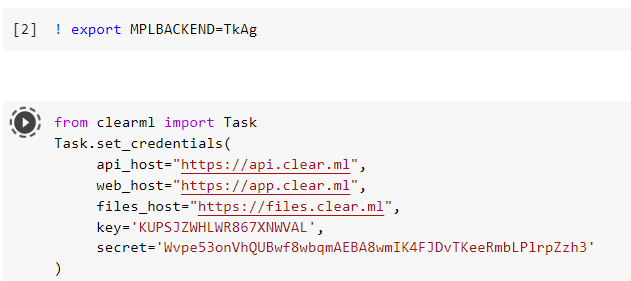
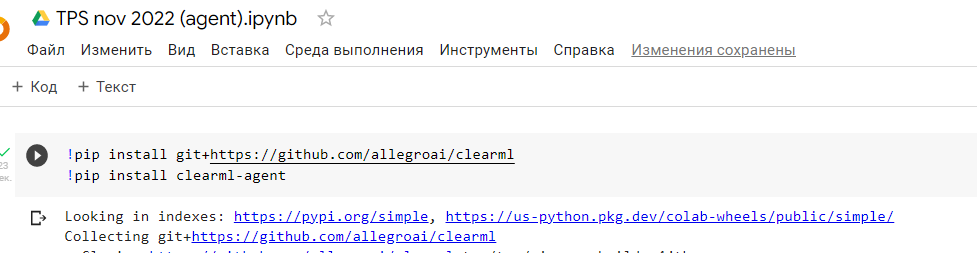
Дальше вам предложат добавить эксперимент в очередь.



Если у приложения нет сервера для обработки очередей, вам придется самим создать его на своем компьютере или сервере. Для этого нужно пошагово выполнить инструкцию и после запуска агента (Agent) отправить эксперимент в очередь и ожидать его обработки.



Clearml-agent запустим на *google colab* уже известным вам способом



после чего за процессом выполнения программы можно следить на вкладке *Console*.

Для командной работы будет полезной функция шеринга экспериментов (можно просто передать ссылку), а также удобное сравнение экспериментов в графическом интерфейсе.

## Тест

1. Какой объект в ClearML основной для управления конвейером операций? (0.25)
   1. **Task**
   2. Process
   3. Procedure
   4. Job
2. Какая информация поступает на вход для Task ClearML? (0.25)
   1. **гиперпараметры**
   2. графики обучения моделей
   3. **данные о концигурации**
   4. ссылка на git
3. Какая параметры у функции report\_scalar()? (0.25)
   1. **series**
   2. **iteration**
   3. scalars
   4. **value**
4. Какая команда сохраняет артефакт выполнения конвейера? (0.25)
   1. execution
   2. **task.upload\_artifact()**
   3. info
   4. scalars

## Итоги/выводы

В этом юните вы научились управлять экспериментами в проекте машинного обучения, логировать информацию и повторять эксперимент.

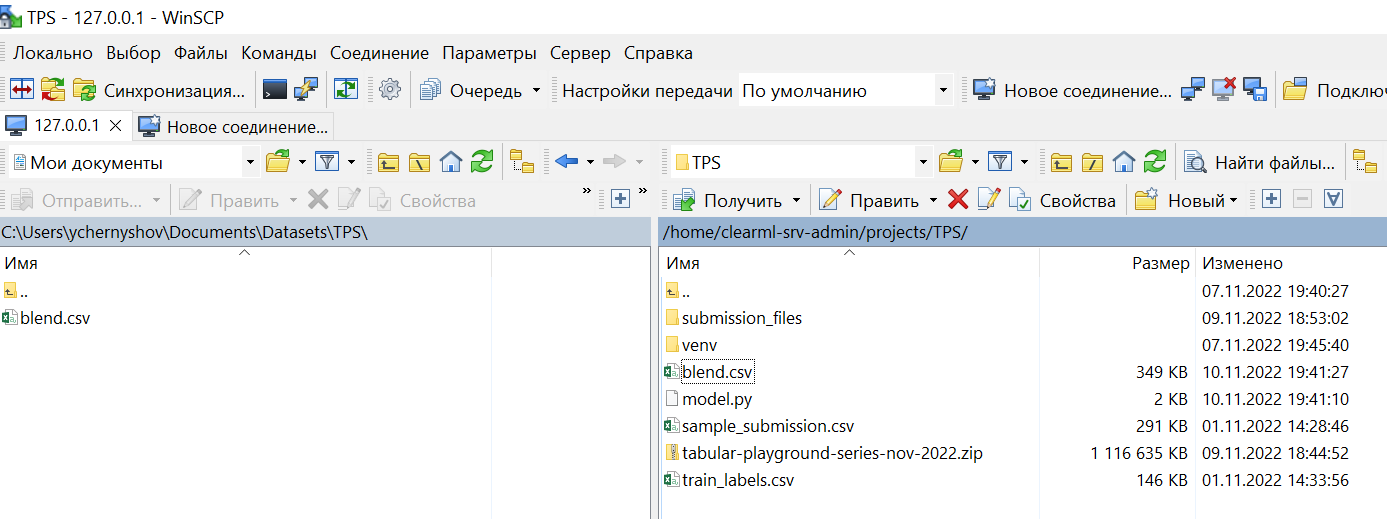
# Модуль 2. Юнит 11. Сборка общего пайплайна, оценка результатов и загрузка итогового файла.

## Введение

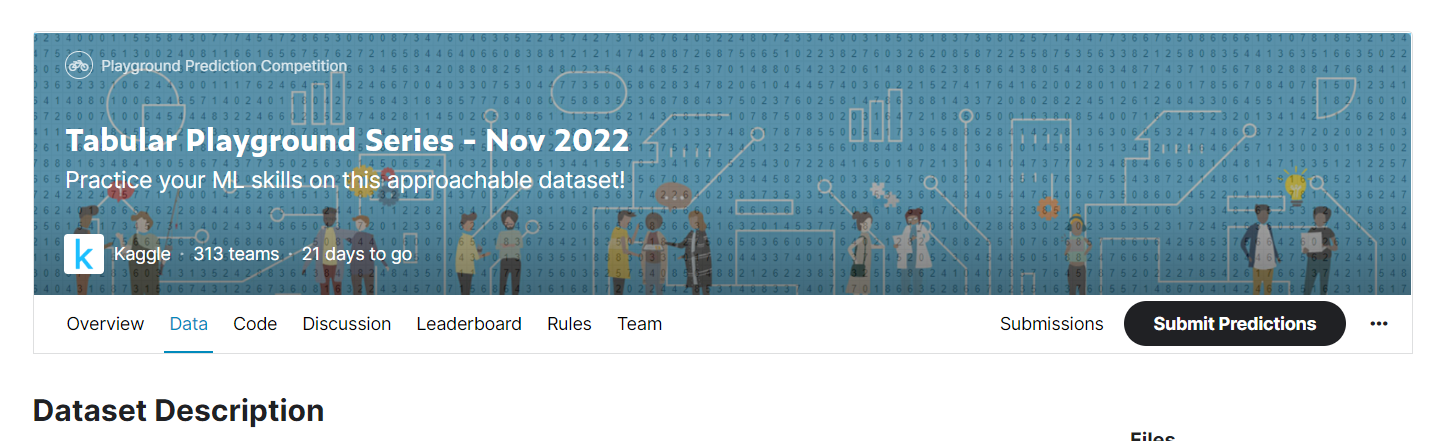
Полученные результаты необходимо опубликовать на площадке конкурса, чтобы получить результат работы построенной вами модели на тестовой (закрытой) выборке данных и, возможно, попасть в leaderboard.

## Содержание

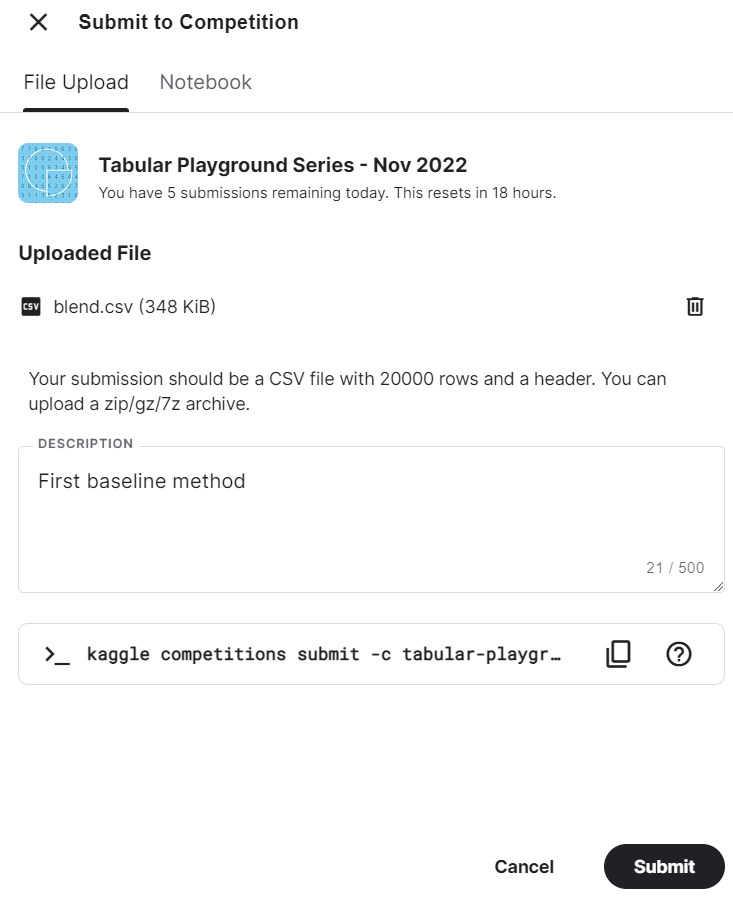
Для предоставления результатов на площадке kaggle.com вы должны представить файл \*.csv. Это можно сделать вручную, загрузив его на площадку. Для этого перенесите файл на локальный компьютер, например, с помощью программы WinSCP



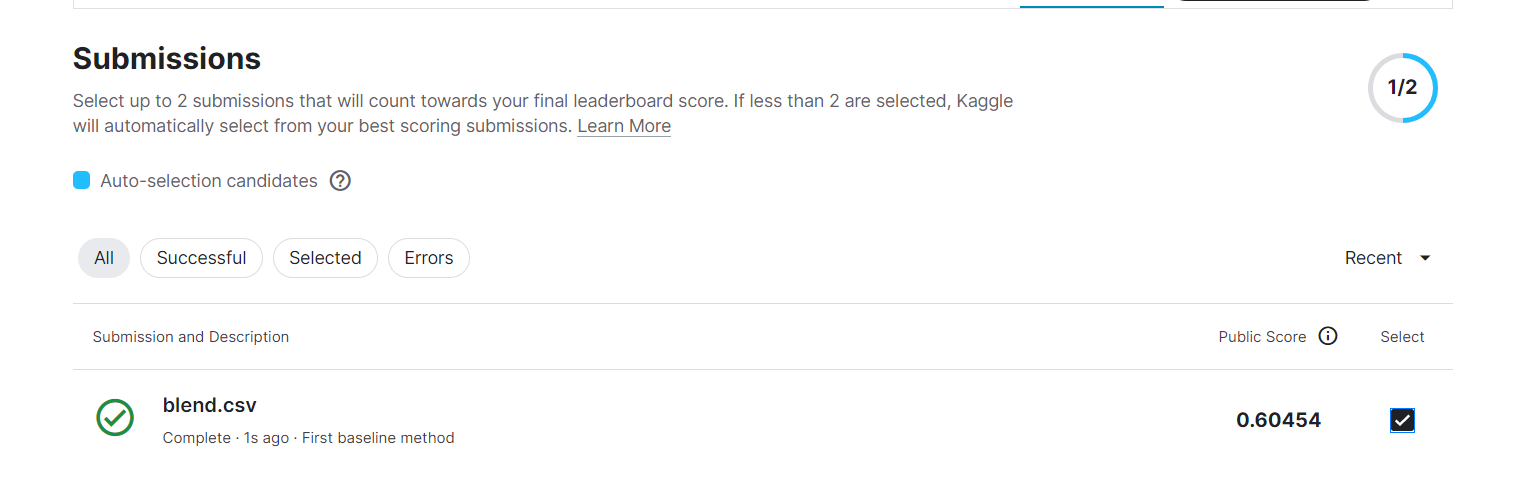
Затем в интерфейсе соревнования на площадке kaggle.com загрузите файл с результатом



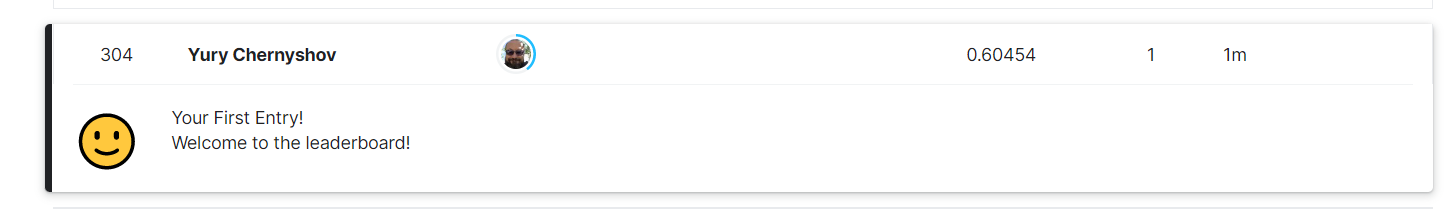
Нажмите *Submit Predictions* и добавьте файл



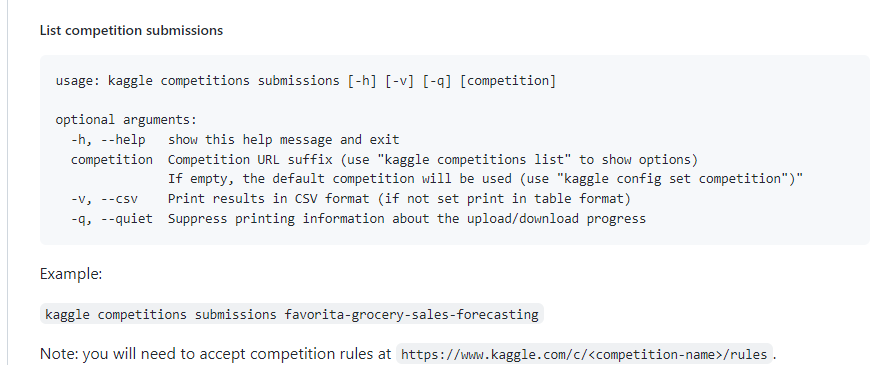
После этого можно будет увидеть оценку вашего результата на закрытой выборке



и вашу позицию в общем leaderboard



Второй, функционально более удобный способ заключается в использовании уже известного вам API, описание находится по адресу <https://github.com/Kaggle/kaggle-api.> Например, информация о том, как осуществлять публикацию результатов:



Публикация производится следующей командой



И, наконец, вы можете посмотреть свой результат в leaderboard



## Тест

1. В каком виде необходимо предоставить результат выполнения задачи в конкурсе? (0.25)
   1. xml,
   2. json,
   3. **csv,**
   4. doc
2. Как можно осуществить публикацию (submission) результатов на площадку kaggle.com? (0.25)
   1. По электронной почте
   2. **Вручную на площадке**
   3. **С помощью API kaggle**
   4. По СМС
3. Какая команда API kaggle опубликует результат на площадке? (0.25)
   1. kaggle result publush
   2. **kaggle competitions submissions**
   3. kaggle submissions put
   4. kaggle finish
4. Как через kaggle API увидеть свой результат? (0.25)
   1. kaggle result
   2. kaggle leaderboard
   3. **kaggle competition leaderboard**
   4. kaggle competition result

## Итоги/выводы

В этом юните вы научились пользоваться API kaggle для того, чтобы взаимодействовать с площадкой соревнования, например, публиковать полученный результат.

# Модуль 2. Юнит 12. Итоговое задание

## Содержание

Вам необходимо разделиться на команды, каждая не более 3 человек и выбрать одно из соревнований *Tabular Playground Series* за 2022 год.

С использованием ClearML необходимо

* Настроить для работы app.clear.ml;
* Загрузить и обработать данные;
* Обучить модель и улучшить ее работу, вы можете использовать для улучшения любые доступные методы: менять гиперпараметры, добавлять новые модели и их ансамбли и так далее.
* Сохранить результаты, загрузить на kaggle.com

Для сдачи задания необходимо приложить на платформу:

* Скриншот трекинга экспериментов из app.clear.ml, отсортированный по убыванию качества работы модели.
* Скриншот с kaggle.com, показывающий достигнутый вашей командой score в leaderboard.

# Итоги/выводы по модулю

Модуль 2 был мы посвятили описанию и практическому использованию инструмента ClearML, который набирает популярность благодаря своему удобству и большому набору полезных функций, автоматизирующих и делающих более удобной работу участников проекта машинного обучения. Сначала мы рассмотрели основные понятия экосистемы ClearML, а затем на нескольких практических задачах рассмотрели применение ClearML в проекте.

В юнитах 8-11 подробно описано использование ClearML для решения задач на площадке соревнований по машинному обучению kaggle.com. Наверняка вы уже знали про нее и, возможно, решали некоторые задачи. Опытные участники соревнований по машинному обучанию знают, что большего успеха можно добиться в команде, поскольку это позволяет задействовать больше вычислительных ресурсов и проверить больше гипотез. В этом случае как раз полезными оказываются подходы MLOps, которые мы и рассмотрели.

# Список источников

Образовательные и обзорные статьи, справочники, FAQ

1. Что такое блендинг в ML <https://habr.com/ru/company/skillfactory/blog/531416/>
2. Feature-weighted linear stacking https://arxiv.org/pdf/0911.0460.pdf.

Официальные страницы программных продуктов, инструментов, утилит

1. Площадка соревнований по ML <https://kaggle.com>.
2. Официальная страница ClearML <https://clear.ml>.

Репозитории и источники, использованные в модуле для практической работы

1. Облачный сервис ClearML https://app.clear.ml/.