Installationsanleitung

Im Folgenden wird der Installationsvorgang der MAL2 Komponente **Fake-Shop Detection - Explainability Dashboard** beschrieben und dessen Funktionsumfang anhand von plakativen Beispielen erörtert. Die Komponente interagiert mit den trainierten Fake-Shop Prediction Modellen und ermöglicht es beliebige Seiten zu validieren. Die Interaktion mit dem Tool erfolgt über ein simples Command-Line Interface, die Ausgabe wird als html Dashboard persistiert. Lime und Shap bieten ein tieferes Verständnis hinsichtlich Nachvollziehbarkeit und Erklärbarkeit der getroffenen Entscheidungen.

Die Anwendung wird als Linux Docker Container gebaut und die UI ist via VNC entweder direkt im Browser oder über einen VNC Client aufrufbar. Docker unterstützt die Host-Betriebssysteme Windows, Mac und Linux. Der Installations-Guide beschreibt die benötigten Schritte unter Linux, die unter Windows analaog durchzuführen sind (Windows spezifische Unterschiede sind spezifisch aufgelistet)

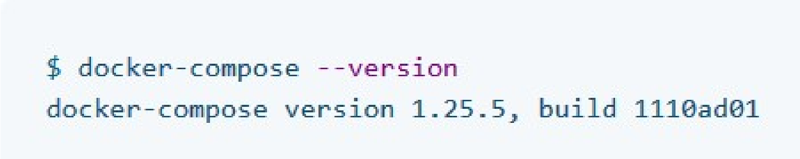
Installing Docker

In 2013, Docker introduced what would become the industry standard for containers. Containers are a standardized unit of software that allows developers to isolate their app from its environment, solving the “it works on my machine” headache. For millions of developers today, Docker is the de facto standard to build and share containerized apps - from desktop, to the cloud.

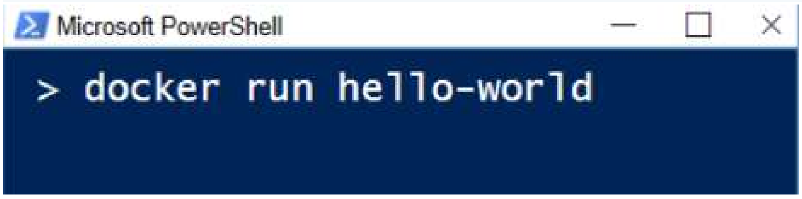
Follow the installation guidelines provided at: https://docs.docker.com/engine/install/

Windows:	Install Docker Desktop https://docs.docker.com/docker-for-windows/
Ubuntu:	Install Docker https://docs.docker.com/engine/install/ubuntu/ and Install Docker Compose https://docs.docker.com/compose/install/

Please note, it is not mandatory to register an account at hub.docker.com. After your setup is completed verify your installation is properly working by running both commands ‘docker run hello-world’ and ‘docker-compose –version’ in your shell. Great now we have all the tooling in place that is necessairy for building the MAL2 Dashboard!



```
$ docker-compose --version
docker-compose version 1.25.5, build 1110ad01
```

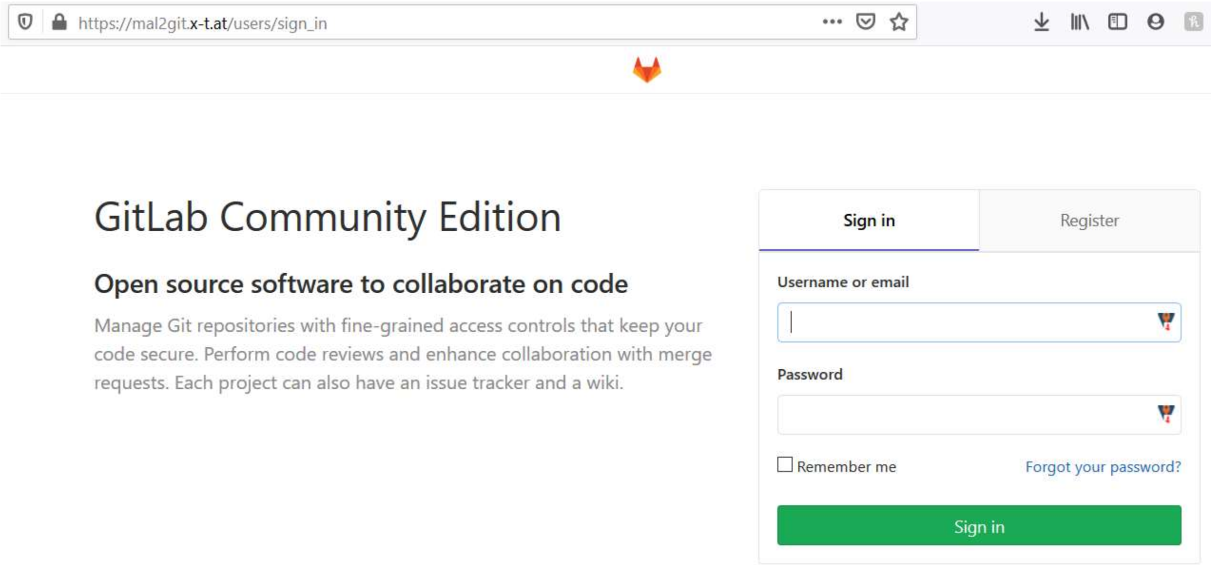


```
> docker run hello-world
```

Checkout the MAL2 dashboard source code

GitLab is a web-based DevOps lifecycle tool that provides a Git-repository manager providing wiki, issue-tracking and continuous integration/continuous deployment pipeline features, etc.

The installation process is designed that the application is directly built from the latest MAL2 source code. MAL2 uses a git repository that’s hosted by the project partner XNET for all software artefacts of the project. The repositories are located at https://mal2git.x-t.at To download the sources you need to **sign up for an account**. Please Register an account at the MAL2 gitlab.

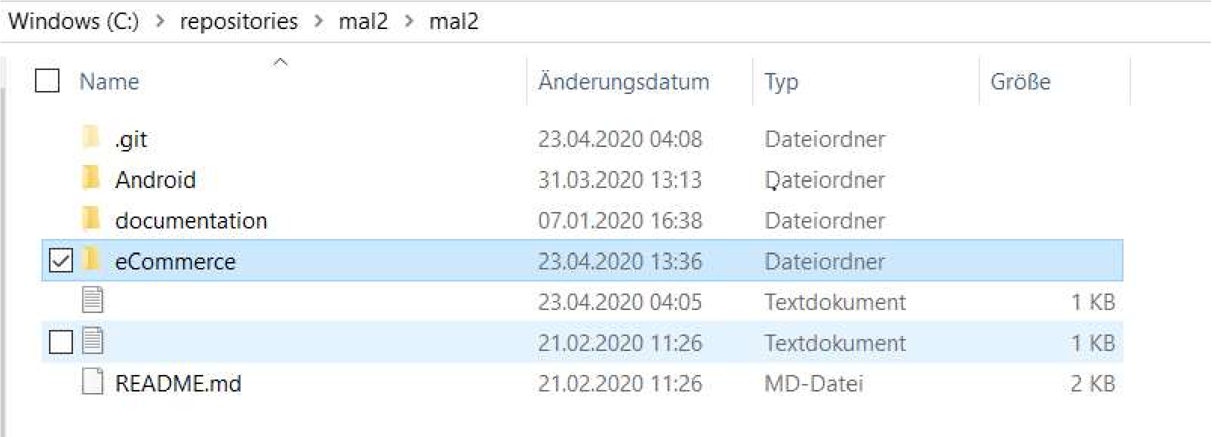


GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Select a location on your local file storage. We’ll be referring to this from now on as mal2_base_dir. For example ‘C:\repositories\mal2\mal2’. After you completed Option a or Option b your file structure should look like this:



Windows (C:) > repositories > mal2 > mal2

□ Name	Änderungsdatum	Typ	Größe
.git	23.04.2020 04:08	Dateiordner	
Android	31.03.2020 13:13	Dateiordner	
documentation	07.01.2020 16:38	Dateiordner	
☑ eCommerce	23.04.2020 13:36	Dateiordner	
	23.04.2020 04:05	Textdokument	1 KB
□	21.02.2020 11:26	Textdokument	1 KB
README.md	21.02.2020 11:26	MD-Datei	2 KB

<u>Option a</u>: If you’re familiar with git: clone the repository and switch to the dashboard branch

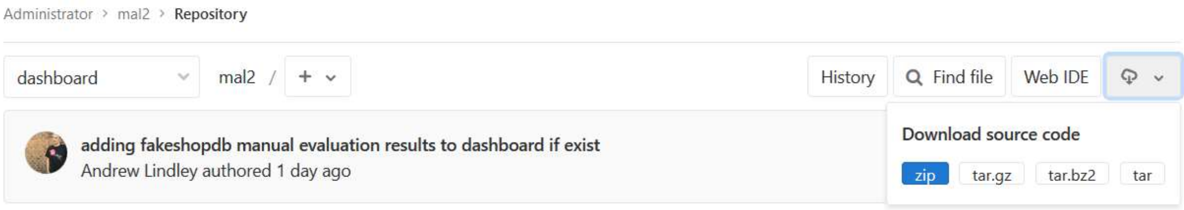
```
git clone https://mal2git.x-t.at/root/mal2.git

git pull
git fetch -all
git reset --hard origin/dashboard
```

Option b: Doanlowding the sources as zip file. Therefore after login proceed to the site

<u>Option b</u>: Doanlowding the sources manually. Therefore after login proceed to the site

https://mal2git.x-t.at/root/mal2/tree/dashboard

Download the zip file (icon next to Web IDE) and extract the files to your mal2_base_dir



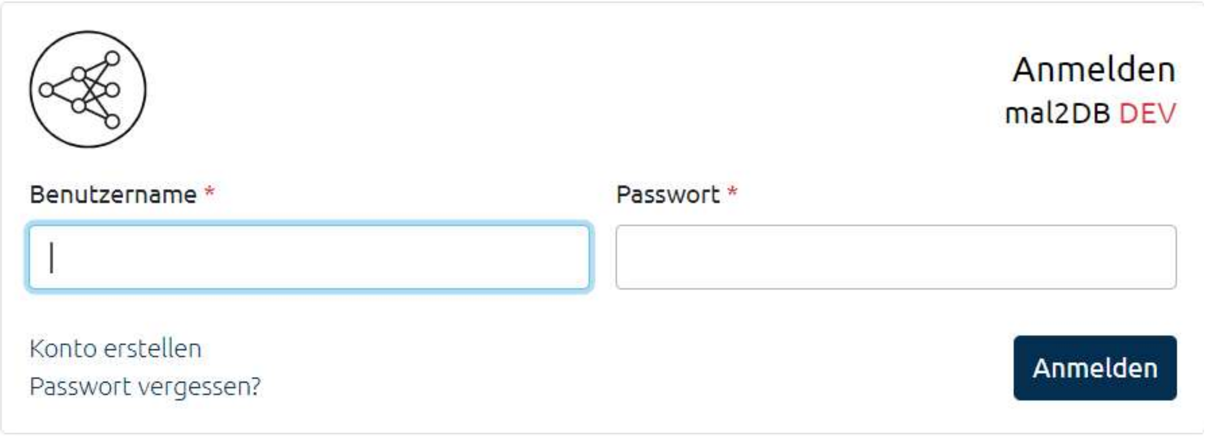
adding fakeshopdb manual evaluation results to dashboard if exist
Andrew Lindley authored 1 day ago

Fake-Shop Database Interaction (optional)

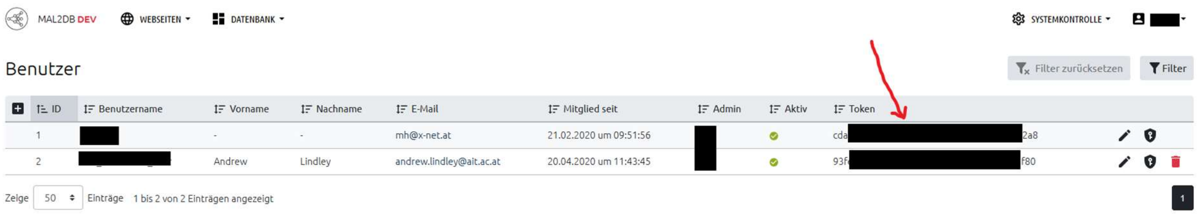
Besides the verification of arbitrary sites (web shops) against the trained prediction models, the MAL2 dashboard is able to pull data from the Fake-Shop Database[1] to include these results of the manual expert evaluation process. Manual verification is performed by Watchlist-Internet[2] based on the MAL2 checklist. To interact with the Fake-Shop DBs API an access token is required.

Please note: A default access token is already provided. No further action required. Skip to next section.

Optional: To access the details of the manual inspection please register an account at https://db-dev.malzwei.at/users/signin/?next=/users/



To change your access token: After login you're able to receive your access token from the management console at https://db-dev.malzwei.at/users/ (see red arrow) – copy that String Token

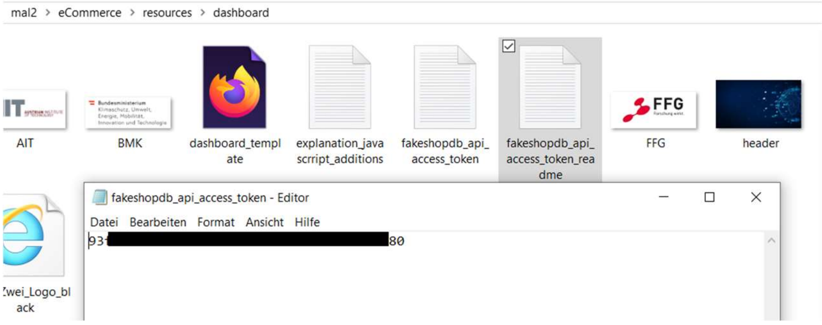


Benutzer

ID	Benutzername	Vorname	Nachname	E-Mail	Mitglied seit	Admin	Aktiv	Token
1	[redacted]	-	-	mh@x-net.at	21.02.2020 um 09:51:56	[redacted]	✓	cda[redacted]2a8
2	[redacted]	Andrew	Lindley	andrew.lindley@ait.ac.at	20.04.2020 um 11:43:45	[redacted]	✓	93f[redacted]f80

Zeige 50 Einträge 1 bis 2 von 2 Einträgen angezeigt

Switch to your local directory mal2_base_dir`\eCommerce\resources\dashboard`
create a new file called `fakeshopdb_api_access_token.txt`
insert the Token String and save the file.


mal2 > eCommerce > resources > dashboard
AIT
BMK
dashboard_template
explanation_javascript_additions
fakeshopdb_api_access_token
fakeshopdb_api_access_token_readme
FFG
header
Zwei_Logo_black
fakeshopdb_api_access_token - Editor
Datei Bearbeiten Format Ansicht Hilfe

```
93[redacted]80
```

[1] https://db-dev.malzwei.at/db/fake_shop/

[2] https://www.watchlist-internet.at/

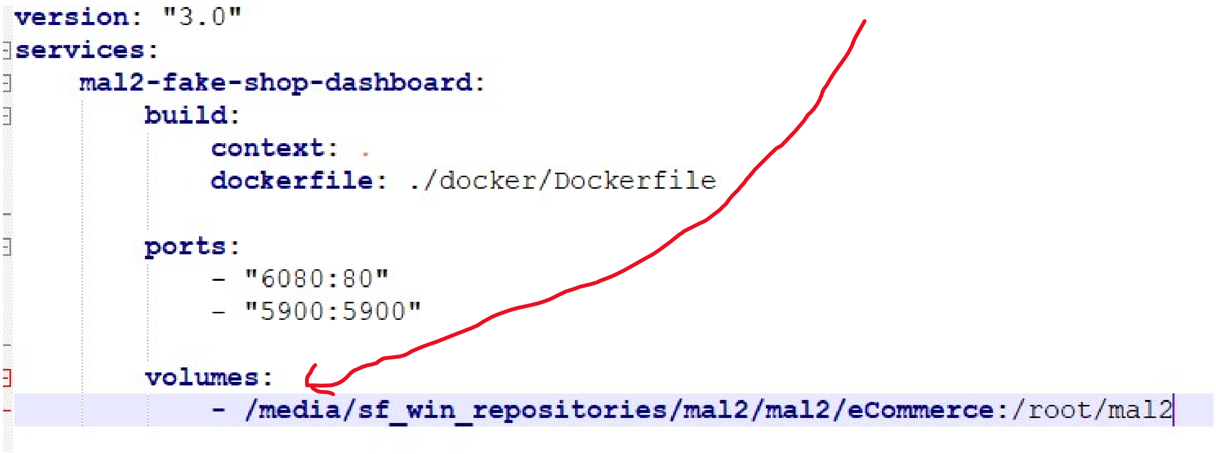
Build the MAL2 Dashboard Virtual Machine image

All the required dependencies and steps to automatically assemble and build the application are provided within a Dockerfile located in \eCommerce\docker\Dockerfile. All information on launching the system (ports, mounted volumes, etc.) are provided in the \eCommerce\docker-compose.yml. If you'd like to setup the system locally checkout these files as they contain all required step on build the component locally.

There's one final configuration we need to adjust before we can start building the VM image.

Open the file `docker-compose.yml` with a text editor (e.g. Windows: Notepad) and replace the String '/media/sf_win_repositories/mal2/mal2/eCommere' with the location of your mal2_base_dir`\eCommerce` directory on your local computer. Make sure to save the file.

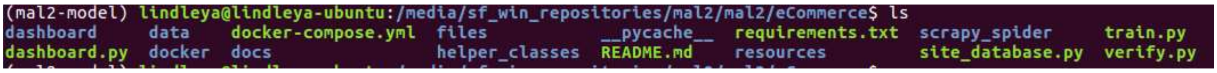
Please make sure to use the full path (e.g. C:\daten\mal2\eCommerce). This folder on your local computer will be mounted as volume (drive) in the virtual machine and allows you to persist the dashboards you created even when shutting down the VM.



```
version: "3.0"
services:
    mal2-fake-shop-dashboard:
        build:
            context: .
            dockerfile: ./docker/Dockerfile

        ports:
            - "6080:80"
            - "5900:5900"

        volumes:
            - /media/sf_win_repositories/mal2/mal2/eCommerce:/root/mal2
```

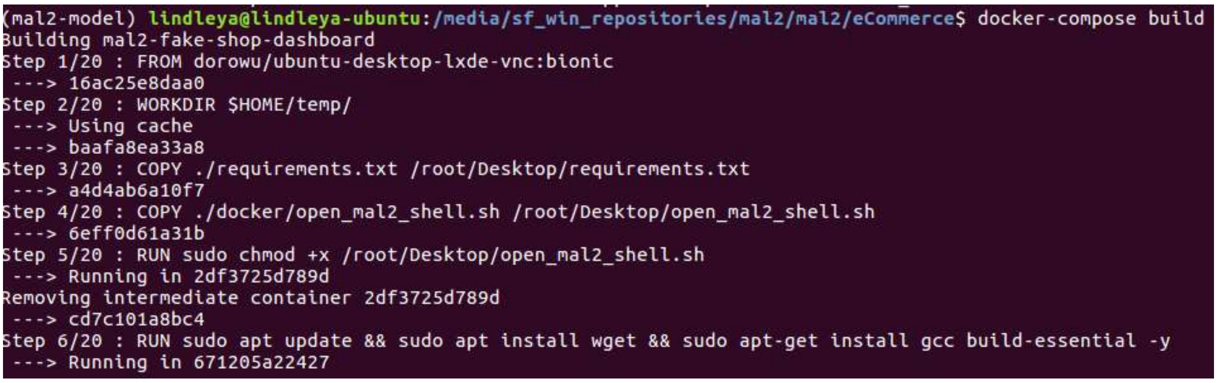
Open your Docker Desktop App and access the Microsoft Powershell / Linux Bash Shell

change directory to mal2_base_dir`\eCommerce` Typing 'ls' should list the following items including the `docker-compose.yml` file



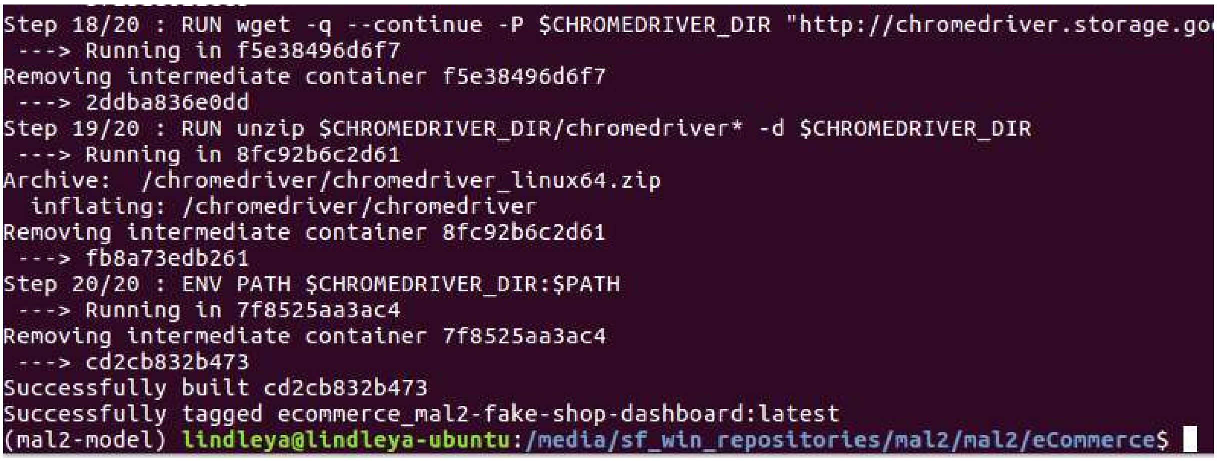
```
(mal2-model) lindleya@lindleya-ubuntu:/media/sf_win_repositories/mal2/mal2/eCommerce$ ls
dashboard     data    docker-compose.yml  files           __pycache__  requirements.txt  scrapy_spider     train.py
dashboard.py  docker  docs                helper_classes  README.md    resources         site_database.py  verify.py
```

Start the build process of the virtual machine image by calling
`'docker-compose build'`
You're able to follow the build process (20 steps) in your console. No manual intervention required.
This may take up to 5-15 minutes depending on the speed of your internet provider



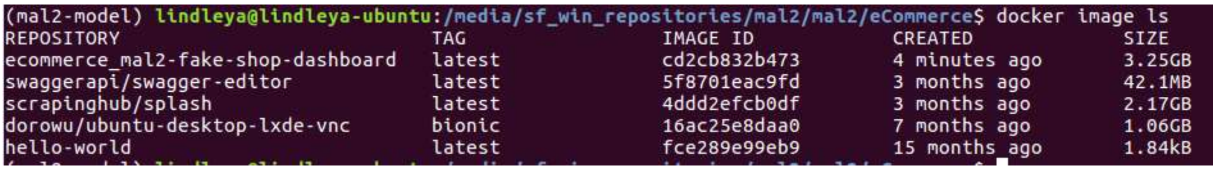
```
(mal2-model) lindleya@lindleya-ubuntu:/media/sf_win_repositories/mal2/mal2/eCommerce$ docker-compose build
Building mal2-fake-shop-dashboard
Step 1/20 : FROM dorowu/ubuntu-desktop-lxde-vnc:bionic
 ---> 16ac25e8daa0
Step 2/20 : WORKDIR $HOME/temp/
 ---> Using cache
 ---> baafa8ea33a8
Step 3/20 : COPY ./requirements.txt /root/Desktop/requirements.txt
 ---> a4d4ab6a10f7
Step 4/20 : COPY ./docker/open_mal2_shell.sh /root/Desktop/open_mal2_shell.sh
 ---> 6eff0d61a31b
Step 5/20 : RUN sudo chmod +x /root/Desktop/open_mal2_shell.sh
 ---> Running in 2df3725d789d
Removing intermediate container 2df3725d789d
 ---> cd7c101a8bc4
Step 6/20 : RUN sudo apt update && sudo apt install wget && sudo apt-get install gcc build-essential -y
 ---> Running in 671205a22427
```

Once the process has completed you'll see the statement
`Successfully tagged ecommerce_mal2-fake-shop-dashboard:latest`



```
Step 18/20 : RUN wget -q --continue -P $CHROMEDRIVER_DIR "http://chromedriver.storage.go
 ---> Running in f5e38496d6f7
Removing intermediate container f5e38496d6f7
 ---> 2ddba836e0dd
Step 19/20 : RUN unzip $CHROMEDRIVER_DIR/chromedriver* -d $CHROMEDRIVER_DIR
 ---> Running in 8fc92b6c2d61
Archive:  /chromedriver/chromedriver_linux64.zip
  inflating: /chromedriver/chromedriver
Removing intermediate container 8fc92b6c2d61
 ---> fb8a73edb261
Step 20/20 : ENV PATH $CHROMEDRIVER_DIR:$PATH
 ---> Running in 7f8525aa3ac4
Removing intermediate container 7f8525aa3ac4
 ---> cd2cb832b473
Successfully built cd2cb832b473
Successfully tagged ecommerce_mal2-fake-shop-dashboard:latest
(mal2-model) lindleya@lindleya-ubuntu:/media/sf_win_repositories/mal2/mal2/eCommerce$ 
```

You can check out the size of the created virtual machine image calling
`docker image ls`



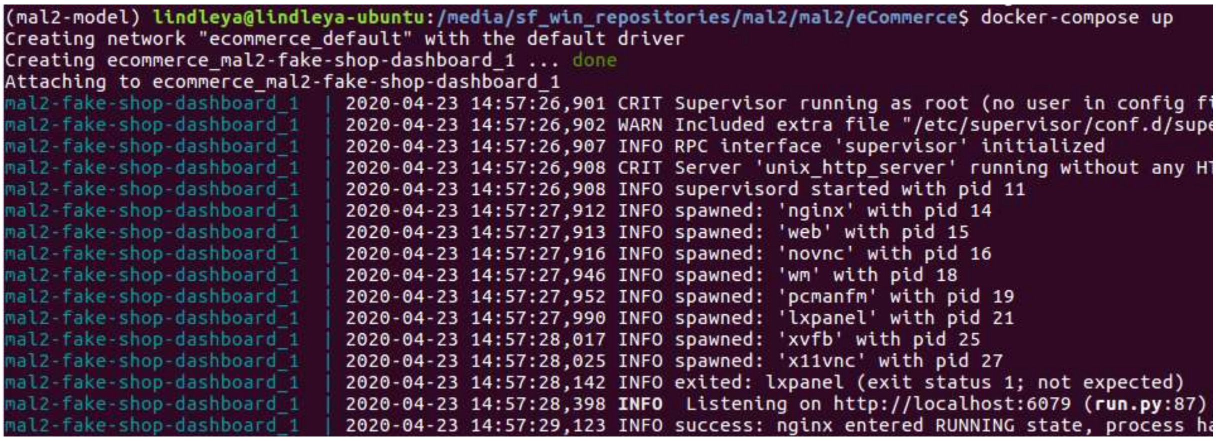
```
(mal2-model) lindleya@lindleya-ubuntu:/media/sf_win_repositories/mal2/mal2/eCommerce$ docker image ls
REPOSITORY                          TAG        IMAGE ID       CREATED          SIZE
ecommerce_mal2-fake-shop-dashboard  latest     cd2cb832b473   4 minutes ago    3.25GB
swaggerapi/swagger-editor           latest     5f8701eac9fd   3 months ago     42.1MB
scrapinghub/splash                  latest     4ddd2efcb0df   3 months ago     2.17GB
dorowu/ubuntu-desktop-lxde-vnc      bionic     16ac25e8daa0   7 months ago     1.06GB
hello-world                         latest     fce289e99eb9   15 months ago    1.84kB
```

Building the virtual machine is only required once, as long as non of the dependencies have changed.

Start the MAL2 Dashboard Virtual Machine

Open your Docker Desktop App and access the Microsoft Powershell / Linux Bash Shell

change directory to mal2_base_dir`\eCommerce` and start the VM by calling
`docker-compose up`
This should only take seconds and look similar to the Screenshot below



```
(mal2-model) lindleya@lindleya-ubuntu:/media/sf_win_repositories/mal2/mal2/eCommerce$ docker-compose up
Creating network "ecommerce_default" with the default driver
Creating ecommerce_mal2-fake-shop-dashboard_1 ... done
Attaching to ecommerce_mal2-fake-shop-dashboard_1
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:26,901 CRIT Supervisor running as root (no user in config fi
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:26,902 WARN Included extra file "/etc/supervisor/conf.d/supe
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:26,907 INFO RPC interface 'supervisor' initialized
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:26,908 CRIT Server 'unix_http_server' running without any HT
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:26,908 INFO supervisord started with pid 11
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:27,912 INFO spawned: 'nginx' with pid 14
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:27,913 INFO spawned: 'web' with pid 15
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:27,916 INFO spawned: 'novnc' with pid 16
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:27,946 INFO spawned: 'wm' with pid 18
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:27,952 INFO spawned: 'pcmanfm' with pid 19
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:27,990 INFO spawned: 'lxpanel' with pid 21
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:28,017 INFO spawned: 'xvfb' with pid 25
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:28,025 INFO spawned: 'x11vnc' with pid 27
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:28,142 INFO exited: lxpanel (exit status 1; not expected)
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:28,398 INFO  Listening on http://localhost:6079 (run.py:87)
mal2-fake-shop-dashboard_1  | 2020-04-23 14:57:29,123 INFO success: nginx entered RUNNING state, process ha
```

Congratulations!

Let's move on to read more about how to using the application

Please note to stop a running docker container either precc `Strg + C` or if the service was launched in the background call `docker-compose down`

Benutzerhandbuch

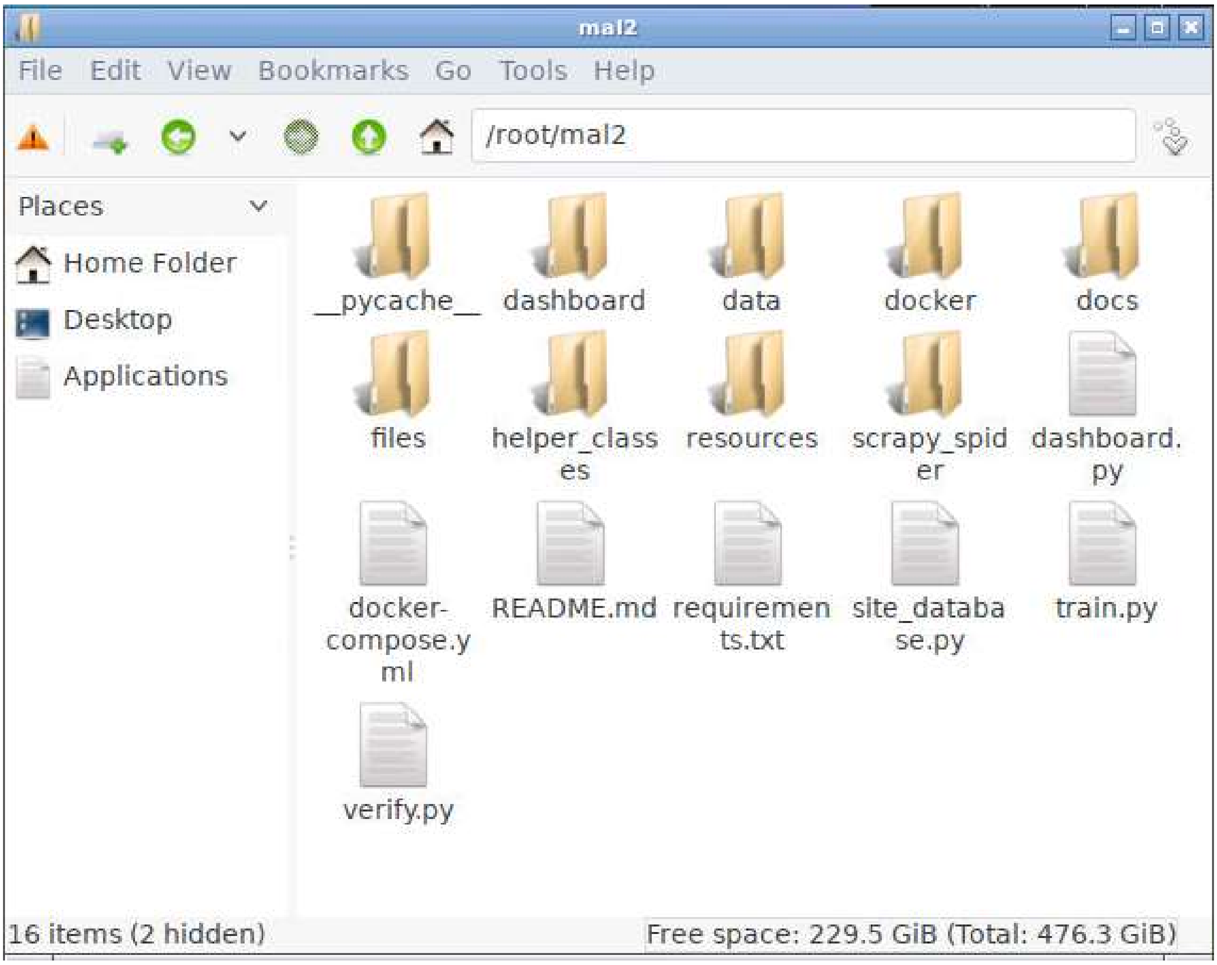
Es existieren zwei Möglicheiten um auf die zuvor gestartete Virtuelle Maschine zuzugreifen.

Option1: direkt im Browser durch Aufruf der Adresse http://127.0.0.1:6080/#/

Option2: durch Verwendung eines VNC Clients[3] unter 127.0.0.1 auf Port 5900

Um zu überprüfen ob das Verzeichnis des lokalen Computers korrekt geladen wurde öffne den Dateimanager. Klick auf das schwarze Symbol in der linken unteren Ecke und wähle `System Tools > File Manager PCManFM`

Folgende Dateien (deines PCs) sollten in der Virtuellen Maschine unter `/root/mal2` vorhanden sein



16 items (2 hidden) Free space: 229.5 GiB (Total: 476.3 GiB)

Anm: Sollte dies nicht der Fall sein gehe zurück zu Schritt „Build the MAL2 Dashboard Virtual Machine image“, stoppe die VM, editiere das docker-compose.yml File und führe ‚docker-compose build‘ aus

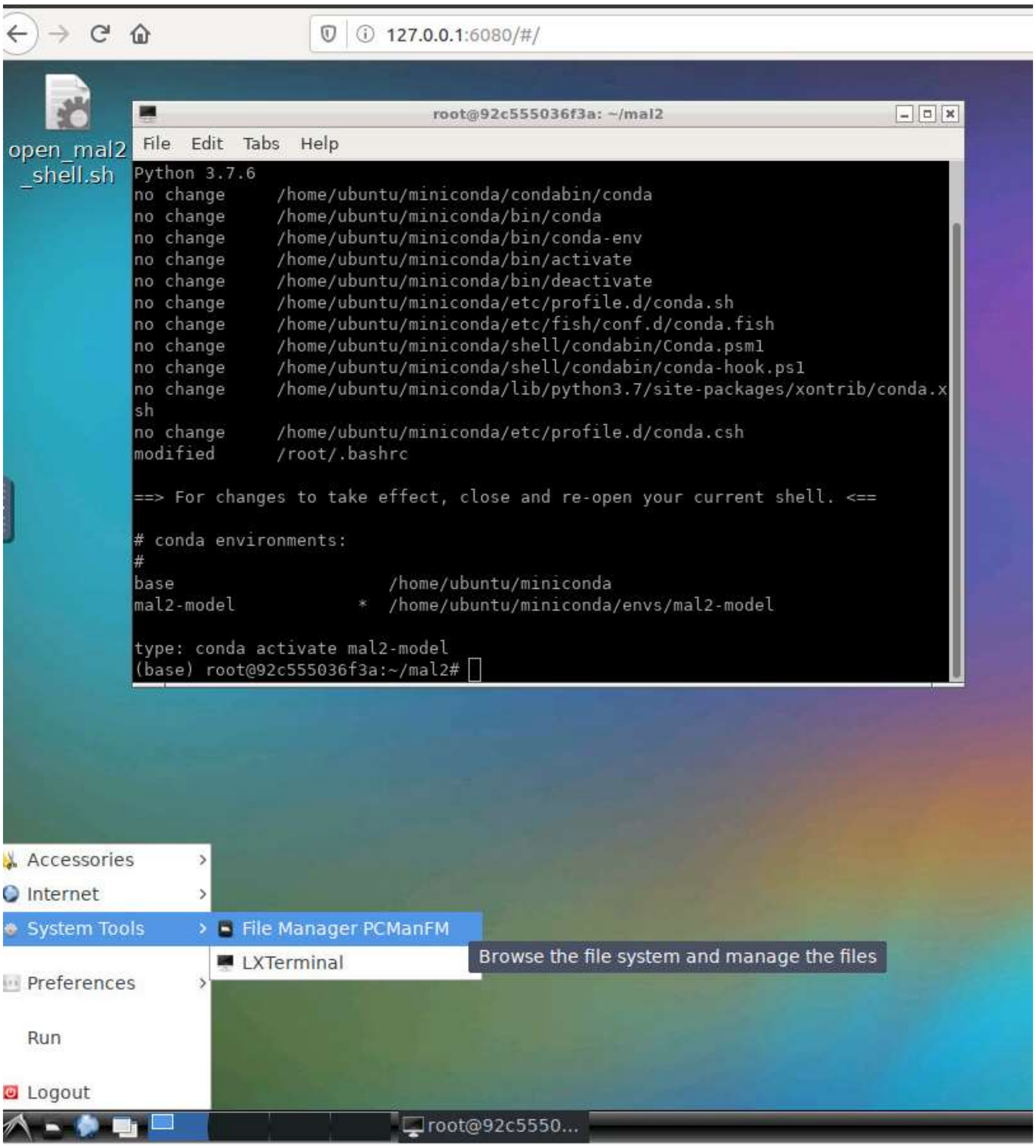
[3] https://www.realvnc.com/de/connect/download/viewer/

Verwendung des Dashboard Builders

Mit Hilfe des Dashboard Builders ist es möglich den Fake-Score von beliebigen Seiten auf Basis der trainierten Machine Learning Modelle zu ermitteln. Aktuell stehen hierfür drei Modelle bereit. Während es sich bei dem Modell Single_Tree um einen work in progress handelt, dessen Prognosen vernachlässigt werden sollten liefern die Modelle XGBoost und Random Forrest hervorrangende Detektionsraten, Precision und Recall von etwa 90% auf dem Trainingsdatensatz.

Öffne dazu am Desktop der Virtuellen Maschine mit einem Doppelklick das Script:

“`open_mal2_shell.sh`” and bestätige die Ausführung mit “Execute in Terminal”



```
Python 3.7.6
no change     /home/ubuntu/miniconda/condabin/conda
no change     /home/ubuntu/miniconda/bin/conda
no change     /home/ubuntu/miniconda/bin/conda-env
no change     /home/ubuntu/miniconda/bin/activate
no change     /home/ubuntu/miniconda/bin/deactivate
no change     /home/ubuntu/miniconda/etc/profile.d/conda.sh
no change     /home/ubuntu/miniconda/etc/fish/conf.d/conda.fish
no change     /home/ubuntu/miniconda/shell/condabin/Conda.psm1
no change     /home/ubuntu/miniconda/shell/condabin/conda-hook.ps1
no change     /home/ubuntu/miniconda/lib/python3.7/site-packages/xontrib/conda.x
sh
no change     /home/ubuntu/miniconda/etc/profile.d/conda.csh
modified      /root/.bashrc

==> For changes to take effect, close and re-open your current shell. <==

# conda environments:
#
base                     /home/ubuntu/miniconda
mal2-model            *  /home/ubuntu/miniconda/envs/mal2-model

type: conda activate mal2-model
(base) root@92c555036f3a:~/mal2# 
```

Aktivierte das conda python environment durch die Eingabe von

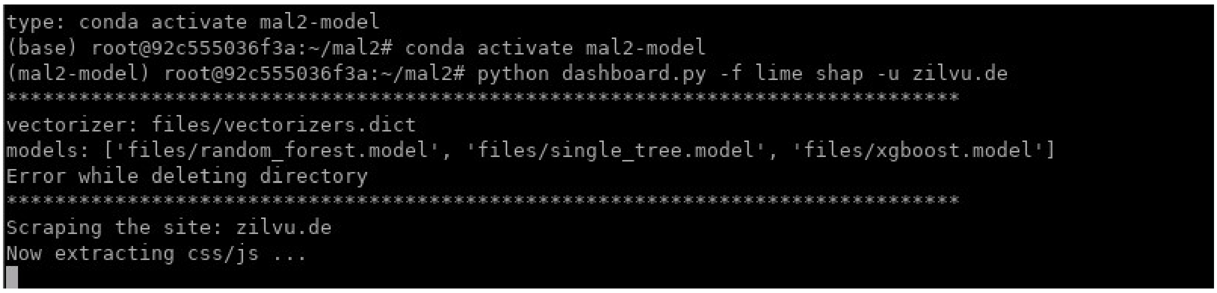
```
conda activate mal2-model
```

was durch ‚(mal2-model)‘ in der linken Ecke der shell bestätigt wird

Beim Dashboard Builder handelt es sich um eine simple Command-Line Anwendung zur Interaktion mit den Python Modulen. Die Ausgabe des Überprüfungsergebnisses erfolgt als html Datei.

Um den shop ‚elekronio.de‘ zu überprüfen starte den dashboard builder über

```
python dashboard.py -u zilvu.de
```



```
type: conda activate mal2-model
(base) root@92c555036f3a:~/mal2# conda activate mal2-model
(mal2-model) root@92c555036f3a:~/mal2# python dashboard.py -f lime shap -u zilvu.de
******************************************************************************
vectorizer: files/vectorizers.dict
models: ['files/random_forest.model', 'files/single_tree.model', 'files/xgboost.model']
Error while deleting directory
******************************************************************************
Scraping the site: zilvu.de
Now extracting css/js ...
```

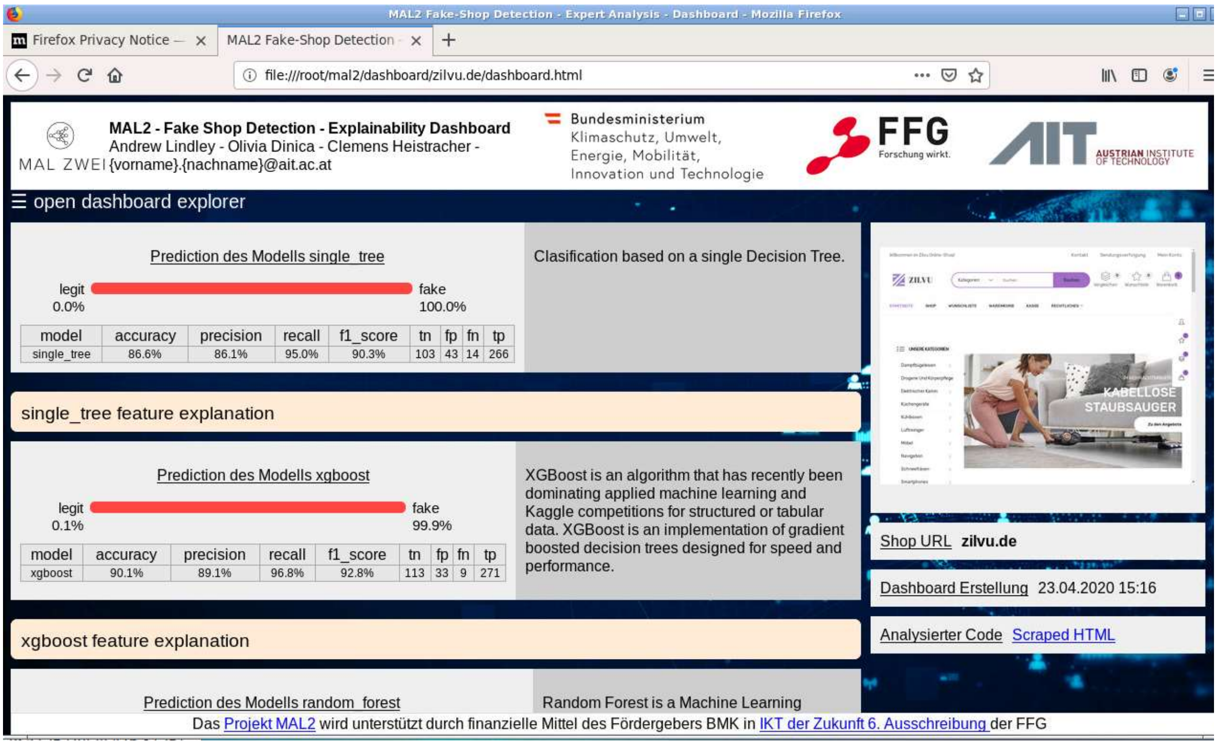
Der Fortschritt der einzelnen Schritte wird in der Console vermerkt und mit der Meldung `created dashboard at /root/mal2/dashboard/zilvu.de/dashboard.html` beendet

Du findest das dashboard nun auf deinem lokalen Computer unter

mal2_base_dir`/eCommerce/dashboard/zilvu.de/dashboard.html`

sowie in der virtuellen Maschine unter

`/root/mal2/dashboard/zilvu.de/dashboard.html`



MAL2 Fake-Shop Detection - Expert Analysis - Dashboard - Mozilla Firefox
Firefox Privacy Notice —
MAL2 Fake-Shop Detection
file:///root/mal2/dashboard/zilvu.de/dashboard.html
MAL ZWEI
Bundesministerium Klimaschutz, Umwelt, Energie, Mobilität, Innovation und Technologie
FFG Forschung wirkt.
AIT AUSTRIAN INSTITUTE OF TECHNOLOGY
☰ open dashboard explorer
KABELLOSE STAUBSAUGER

MAL2 - Fake Shop Detection - Explainability Dashboard
Andrew Lindley - Olivia Dinica - Clemens Heistracher - {vorname}.{nachname}@ait.ac.at

Prediction des Modells single_tree

legit 0.0% — fake 100.0%

model	accuracy	precision	recall	f1_score	tn	fp	fn	tp
single_tree	86.6%	86.1%	95.0%	90.3%	103	43	14	266

Clasification based on a single Decision Tree.

single_tree feature explanation

Prediction des Modells xgboost

legit 0.1% — fake 99.9%

model	accuracy	precision	recall	f1_score	tn	fp	fn	tp
xgboost	90.1%	89.1%	96.8%	92.8%	113	33	9	271

XGBoost is an algorithm that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

xgboost feature explanation

Prediction des Modells random_forest

Random Forest is a Machine Learning

Shop URL zilvu.de

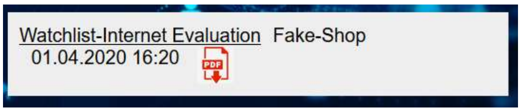
Dashboard Erstellung 23.04.2020 15:16

Analysierter Code Scraped HTML

Das Projekt MAL2 wird unterstützt durch finanzielle Mittel des Fördergebers BMK in IKT der Zukunft 6. Ausschreibung der FFG

Optionale Parameter beim Aufruf von dashboard.py	
`-f lime shap`	Explainability Grafiken zum Verstädnis der Entscheidungspfade generieren
`--use-cache`	Scraping des html nur falls noch nicht lokal vorhanden
`--submit-results`	Ergebnisse von XGBoost an die Fake-Shop DB melden (über 60% Threshold)

Ist eine abgefragte Seite in der MAL2 Fake-Shop Datenbank bereits als Fake-Shop klassifiziert worden, so wird dies in den Metadaten des jeweiligen Dashboards (rechts im Bild) gekennzeichnet und die Detailergebnisse der Überprüfung verlinkt. Um auf die Ergebnisse zuzgreifen wird ein gültiger Account für die fake-shop Datenbank benötigt.



Watchlist-Internet Evaluation Fake-Shop
01.04.2020 16:20