

Démarche Travail pratique

Objectif:

Partie-3 : Faire des prédictions avec le modèle et tester la méthode predict

Partie-4 : Validation des données dans le modèle

Partie-5 : Ajout d'ingénierie des fonctionnalités dans le pipeline

Partie-6 : Gestion des versions et journalisation

Partie-7 : Créer un package

Partie 8 : Préparation de Rest-API de service de modèle

Partie-9 Ajout de la configuration et de la journalisation et tests à l'API Flask

Partie-10 : Ajout du point de prédiction à l'API Flask

Partie-11 : Ajout de version a l'application flask.

Partie-12 : Ajout de Schéma de validation à l'API Flask

Démarche :

Partie-3 : Faire des prédictions avec le modèle et tester la méthode predict

Après avoir modifié le code de pipeline, vérifiez le staus :

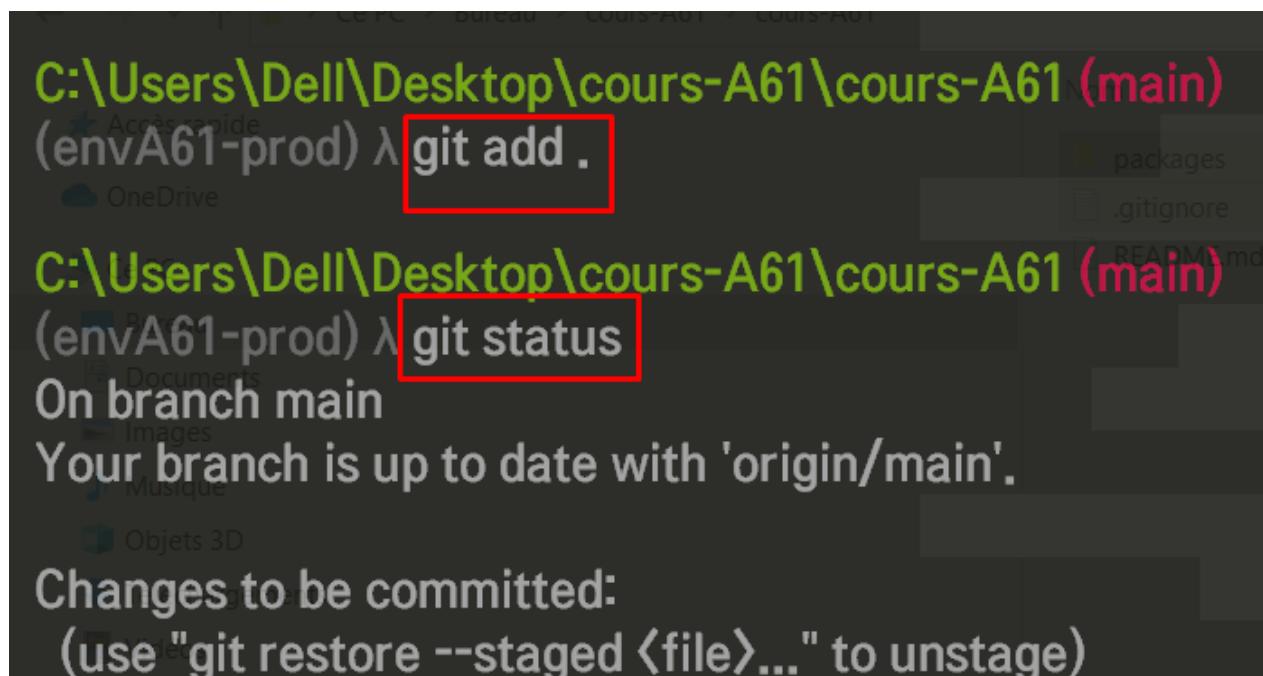
Git status

Git log

Git log -p

Tapez les commandes suivantes, comme les montrent les captures d'écrans ci-dessous :

```
git add .  
git status  
git commit -m 'Partie-3-Ajout prédictions et test'  
git push -u origin main
```



```
C:\Users\Del\lDesktop\cours-A61\cours-A61 (main)  
(envA61-prod) λ git add .  
  
C:\Users\Del\lDesktop\cours-A61\cours-A61 (main)  
(envA61-prod) λ git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
Changes to be committed:  
(use "git restore --staged <file>..." to unstage)
```

Ajoutez;

```
git commit -m 'Partie-3-Ajout prédictions et test'
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git commit -m " Partie-3- Ajout predictions et tests"
[main d6fc4d6] Partie-3- Ajout predictions et tests
```

puis

```
C:\Users\Dell\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git push -u origin main
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 8 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 2.11 KiB | 1.05 MiB/s, done.
Total 13 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/cegep2020/cours-A61.git
 bbe8649..d6fc4d6 main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Pour faire de prediction et test la methode predict

On a ajouté une ligne à la fin de fichier **Tox** pour faire des tests

Exécutez

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\regression_model (main)
(envA61-prod) λ tox -r
Expliquer le code
regression_model recreate: C:\Users\Dell\Desktop\cours-A61\cours-A61\p
ackages\regression_model\.tox\regression_model
Pour faire de prediction et test la methode predict
regression_model installdeps: -rrequirements.txt
Tox Ajout d'une ligne à la fin pour faire des tests
```

Pour recréer l'environnement, tapez la commande ci-dessous :

```
(envA61-prod) \tox -r
regression_model recreate: C:\Users\DELL\Desktop\cours-A61\cours-A61\pac
ackages\regression_model\.tox\regression_model
regression_model installdeps: -rrequirements.txt
regression_model installed: atomicwrites==1.4.0,attrs==20.3.0,colorama
==0.4.4,joblib==0.14.1,more-itertools==8.6.0,numpy==1.18.5,packaging==
20.8,pandas==0.25.3,pluggy==0.13.1,py==1.10.0,pyparsing==2.4.7,pytest=
=5.4.3,python-dateutil==2.8.1,pytz==2020.5,scikit-learn==0.22.2.post1,
scipy==1.6.0,six==1.15.0,wcwidth==0.2.5
regression_model run-test-pre: PYTHONHASHSEED='464'
regression_model run-test: commands[0] | python regression_model/train
_pipeline.py
saved pipeline
regression_model run-test: commands[1] | pytest tests
=====
platform win32 -- Python 3.8.1, pytest-5.4.3, py-1.10.0, pluggy-0.13.1
cachedir: .tox\regression_model\.pytest_cache
rootdir: C:\Users\DELL\Desktop\cours-A61\cours-A61\packages\regression_
_model
collected 1 item
tests\test_predict.py .
[100%]
=====
1 passed in 1.70s =====
summary _____
```

Partie-4- Validation des données dans le modèle

- Valider les données avant la prediction
- Ajout d'un module de validation et l'appelé au niveau du module **predict**
- Mettre à jour nos tests pour tester le nouveau module validation

```
(envA61-prod) λ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory
)
      modified: packages/regression_model/regression_model/predict
.py
      modified: packages/regression_model/tests/test_predict.py
Partie-4- Validation des données

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    packages/regression_model/regression_model/processing/validation.py
Partie-5-Ajout prediction
git add .
git commit -m "Partie-4-Ajout validation"
git push -u origin main

no changes added to commit (use "git add" and/or "git commit -a")
```

```
C:\Users\DELL\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified: packages/regression_model/regression_model/predict
.py
    new file: packages/regression_model/regression_model/process
ing/validation.py
    modified: packages/regression_model/tests/test_predict.py
```

```
C:\Users\DELL\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git commit -m "Partie-4- Ajout bvalidation de données"
```

```
[main f18e49b] Partie-4- Ajout bvalidation de données
 3 files changed, 50 insertions(+), 1 deletion(-)
  create mode 100644 packages/regression_model/regression_model/process
ing/validation.py
```

```
C:\Users\DELL\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git push -u origin main
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 1.59 KiB | 816.00 KiB/s, done.
Total 10 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/cegep2020/cours-A61.git
  d6fc4d6..f18e49b main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

```

ain)
(envA61-prod) λ tox
regression_model installed: atomicwrites==1.4.0,attrs==20.3.0,colorama
==0.4.4,joblib==0.14.1,more-itertools==8.6.0,numpy==1.18.5,packaging==
20.8,pandas==0.25.3,pluggy==0.13.1,py==1.10.0,pyparsing==2.4.7,pytest=
=5.4.3,python-dateutil==2.8.1,pytz==2020.5,scikit-learn==0.22.2.post1,
scipy==1.6.0,six==1.15.0,wcwidth==0.2.5
regression_model run-test-pre: PYTHONHASHSEED='884'
regression_model run-test: commands[0] | python regression_model/train
_pipeline.py
saved pipeline
regression_model run-test: commands[1] | pytest tests

```

```

=====
 test session starts =====
platform win32 -- Python 3.8.1, pytest-5.4.3, py-1.10.0, pluggy-0.13.1
    / config.py
  datasets
cachedir: .tox\regression_model\.pytest_cache
rootdir: C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\regression
_model
collected 2 items
    /__init__.py
tests\test_predict.py ..
    / preprocessors.py
    / validation.py
    / trained_models
===== 2 passed in 1.63s =====
summary
regression_model: commands succeeded
congratulations :)
    / train_pipeline.py

```

Partie-5 Ajout d'ingénierie des fonctionnalités dans le pipeline

- Ajouter un module feature et l'appeler au niveau du module pilene
- Diviser preprocessinfg et feautre pour simplifier le test et la compresnetion du code.

Note : on peut créer un package tout entier pour l'ingénierie de données et l'importer dedans ou même on peut appeler api tout entière pour nous faire le travail....

```
C:\Users\Del\l\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) ⊞ git status
On branch main
Your branch is up to date with 'origin/main'.
  * config.py
  * datasets
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified: packages/regression_model/regression_model/pipeline.py
    modified: packages/regression_model/regression_model/processing/preprocessors.py
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    packages/regression_model/regression_model/processing/features.py
no changes added to commit (use "git add" and/or "git commit -a")
```

```
C:\Users\DELL\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) > git add .
C:\Users\DELL\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) > git commit -m "Partie-5- Ajout ingénierie des fonctionnalités dans le pipeline"
[main 43a0a4b] Partie-5- Ajout ingénierie des fonctionnalités dans le pipeline
3 files changed, 37 insertions(+), 31 deletions(-)
create mode 100644 packages/regression_model/regression_model/process
ing/features.py
```

```
C:\Users\DELL\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) > git push -u origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 1.05 KiB | 1.05 MiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 4 local objects.
To https://github.com/cegep2020/cours-A61.git
  f18e49b..43a0a4b  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

```
(envA61-prod) λ tox
regression_model installed: atomicwrites==1.4.0 attrs==20.3.0 colorama
==0.4.4 joblib==0.14.1 more-itertools==8.6.0 numpy==1.18.5 packaging==
20.8 pandas==0.25.3 pluggy==0.13.1 py==1.10.0 pyparsing==2.4.7 pytest==
5.4.3 python-dateutil==2.8.1 pytz==2020.5 scikit-learn==0.22.2.post1,
scipy==1.6.0 six==1.15.0 wcwidth==0.2.5
regression_model run-test-pre: PYTHONHASHSEED='482'
regression_model run-test: commands[0] | python regression_model/train
_pipeline.py
saved pipeline
regression_model run-test: commands[1] | pytest tests
===== test session starts =====
platform win32 -- Python 3.8.1, pytest-5.4.3, py-1.10.0, pluggy-0.13.1

cachedir: .tox\regression_model\.pytest_cache
rootdir: C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\regression
_model
```

```
===== test session starts =====
platform win32 -- Python 3.8.1, pytest-5.4.3, py-1.10.0, pluggy-0.13.1
cachedir: .tox\regression_model\.pytest_cache
rootdir: C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\regression
_model
collected 2 items
tests\test_predict.py .. [100%]
===== 2 passed in 1.59s =====
summary
regression_model: commands succeeded
congratulations :)
```

Partie-6- Gestion des versions et journalisation

- Gestion des versions est crédible pour la reproductibilité
- Les logs on peut les formatter et les afficher directement dans la console ou bien les rediriger vers un fichier ou bien une base de données, on peut même alimenter avec des rapports...
- Lors d'execution tox, on va voir plus logs compréhensibles.

Git status

```
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
              modified: packages/regression_model/regression_model/__init__.py
              modified: packages/regression_model/regression_model/config/config.py
              modified: packages/regression_model/regression_model/pipeline.py
              modified: packages/regression_model/regression_model/predict.py
              modified: packages/regression_model/regression_model/process/datas_management.py
              modified: packages/regression_model/regression_model/process/features.py
              modified: packages/regression_model/regression_model/process/preprocessors.py
              modified: packages/regression_model/regression_model/train_pipeline.py

Untracked files:
(use "git add <file>..." to include in what will be committed)
    packages/regression_model/regression_model/VERSION
    packages/regression_model/regression_model/config/logging_config.py
    packages/regression_model/regression_model/processing/errors.p
```

```
C:\Users\... cours-A61 (main)
(envA61-prod) λ git add .

C:\Users\... cours-A61 (main)
(envA61-prod) λ git commit -m " Partie-6-Gestion des versions et journalisation"
[main-1744ad3] Partie-6-Gestion des versions et journalisation
11 files changed, 121 insertions(+), 12 deletions(-)
create mode 100644 packages/regression_model/regression_model/VERSION

create mode 100644 packages/regression_model/regression_model/config/
logging_config.py
create mode 100644 packages/regression_model/regression_model/process
ing/errors.py
```

```
C:\Users\... cours-A61 (main)
(envA61-prod) λ git push -u origin main
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 8 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (18/18), 3.35 KiB | 1.12 MiB/s, done.
Total 18 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), completed with 8 local objects.
To https://github.com/cegep2020/cours-A61.git
  43a0a4b..1744ad3 main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Comme vous remarquez il y a plus infos logs

```
C:\Users\Del\l\Desktop\cours-A61\cours-A61\packages\regression_model (main)
(envA61-prod) λ tox
regression_model installed: atomicwrites==1.4.0,attrs==20.3.0,colorama
==0.4.4,joblib==0.14.1,more-itertools==8.6.0,numpy==1.18.5,packaging==
20.8,pandas==0.25.3,pluggy==0.13.1,py==1.10.0,pyparsing==2.4.7,pytest=
=5.4.3,python-dateutil==2.8.1,pytz==2020.5,scikit-learn==0.22.2.post1,
scipy==1.6.0,six==1.15.0,wcwidth==0.2.5
regression_model run-test-pre: PYTHONHASHSEED='712'
regression_model run-test: commands[0] | python regression_model/train
    pipeline.py
2021-01-09 03:24:41,664 — regression_model.processing.data_management
— INFO — save_pipeline:33 — saved pipeline: lasso_regression_output_v0.
1.0.pkl
regression_model run-test: commands[1] | pytest tests
```

Suite

```
===== test session starts =====
platform win32 -- Python 3.8.1, pytest-5.4.3, py-1.10.0, pluggy-0.13.1
2020-03-29 09:36
                               README.md
                               2021-01-08 17:36
cachedir: .tox\regression_model\.pytest_cache
rootdir: C:\Users\Del\l\Desktop\cours-A61\cours-A61\packages\regression
_model
_images
_musique
collected 2 items
    tests\test_predict.py .. [100%]
=====
2 passed in 1.37s
summary
regression_model: commands succeeded
congratulations :)
```

Comme vous avez remarqué quand vous exécutez **tox** vous ne voyez pas les logs, pour les ajouter, modifier le fichier **.tox.ini** comme suit :

```

13 [testenv]
14 install_command = pip install {opts} {packages}
15 deps =
16   -rrequirements.txt
17
18 setenv =
19   PYTHONPATH=.
20
21 commands =
22   python regression_model/train_pipeline.py
23   pytest -s tests
24

```

File Explorer:

- __pycache__
- /* _init_.py
- /* data_management.py
- /* errors.py
- /* features.py
- /* preprocessors.py
- /* validation.py
- trained_models
 - /* _init_.py
 - /* pipeline.py
 - /* predict.py
 - /* preprocessors.py
 - /* train_pipeline.py
- VERSION
- tests
 - __pycache__
 - /* _init_.py
 - /* test_predict.py
- requirements.txt
- tox.ini
- .gitignore
- README.md

Sortie :

```

tests\test_predict.py 2021-01-09 03:37:53,976 — regression_model.predict — INFO — make_prediction:28 — Making predictions with model version: 0.1.0 Inputs: Id M$SubClass M$Zoning LotFrontage LotArea ... MiscVal MoSold \
0 1461 20 RH 80 11622 ... 0
6
YrSold SaleType SaleCondition
0 2010 WD Normal

[1 rows x 80 columns] Predictions: {'predictions': array([112475.83215
879]), 'version': '0.1.0'}
.2021-01-09 03:37:54,114 — regression_model.predict — INFO — make_prediction:28 — Making predictions with model version: 0.1.0 Inputs:
Id M$SubClass M$Zoning LotFrontage LotArea ... MiscVal MoSold \
0 1461 20 RH 80.0 11622 ... 0
6
1 1462 20 RL 81.0 14267 ... 12500
6
2 1463 60 RL 74.0 13830 ... 0
3
3 1464 60 RL 78.0 9978 ... 0
6
4 1465 120 RL 43.0 5005 ... 0
1
...

```

On peut voir input et output logs

Remarque : pourquoi on ne voit pas le log du module train_pipeline?

```
...  
/* data_management.py  
/* errors.py  
/* features.py  
/* preprocessors.py  
/* validation.py  
► trained_models  
/* __init__.py  
/* pipeline.py  
/* predict.py  
/* preprocessors.py  
/* train_pipeline.py  
└ VERSION  
tests  
26     # transform the target  
27     y_train = np.log(y_train)  
28  
29     pipeline.price_pipe.fit(X_train[config.FEATURES], y_train)  
30  
31     _logger.info(f"saving model version: {_version}")  
32     save_pipeline(pipeline_to_persist=pipeline.price_pipe)  
33  
34  
35     if __name__ == "__main__":  
36         run_training()  
37
```

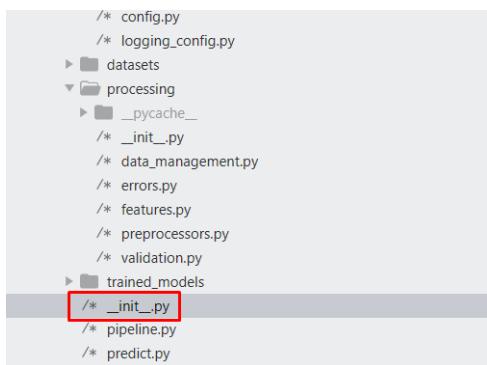
La réponse, puisque tox appelle le fichier **train_pipeline** directement, comme vous voyez ici

```
...  
/* logging_config.py  
► datasets  
▼ processing  
► __pycache__  
/* __init__.py  
/* data_management.py  
/* errors.py  
/* features.py  
/* preprocessors.py  
/* validation.py  
► trained_models  
/* __init__.py  
/* pipeline.py  
/* predict.py  
/* preprocessors.py  
/* train_pipeline.py  
└ VERSION  
tests  
► __pycache__  
/* __init__.py  
/* test_predict.py  
10  
11  
12     _logger = logging.getLogger(__name__)  
13  
14  
15     def run_training() -> None:  
16         """Train the model."""  
17  
18         # read training data  
19         data = load_dataset(file_name=config.TRAINING_D  
20  
21         # divide train and test  
22         X_train, X_test, y_train, y_test = train_test_s  
23             data[config.FEATURES], data[config.TARGET],  
24         ) # we are setting the seed here  
25  
26         # transform the target  
27         y_train = np.log(y_train)  
28
```

Le **__name__** ici est présentement main

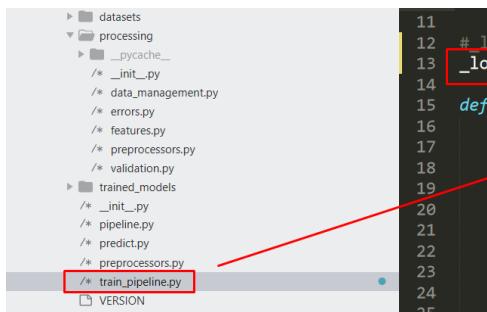
```
...  
/* config.py  
/* features.py  
/* preprocessors.py  
/* validation.py  
► trained_models  
/* __init__.py  
/* pipeline.py  
/* predict.py  
/* preprocessors.py  
/* train_pipeline.py  
└ VERSION  
tests  
28  
29  
30  
31     pipeline.price_pipe.fit(X_train[config.FEATURES], y_train)  
32  
33     _logger.info(f"saving model version: {_version}")  
34     save_pipeline(pipeline_to_persist=pipeline.price_pipe)  
35     if __name__ == "__main__":  
36         run_training()  
37
```

Et le logger quand, il est configuré et appelé `regression_model`, comme suit :



```
9 # Configure logger for use in package
10 logger = logging.getLogger(__name__)
11 logger.setLevel(logging.DEBUG)
12 logger.addHandler(logging_config.get_console_handler())
13 logger.propagate = False
14
15
16 with open(VERSION_PATH, 'r') as version_file:
17     __version__ = version_file.read().strip()
18
```

Si on remplace `__name__` par `regression_model` comme suit :



```
11 # logger = logging.getLogger(__name__)
12 _logger = logging.getLogger("regression_model")
13
14 def run_training() -> None:
15     """Train the model."""
16
17     # read training data
18     data = load_dataset(file_name=config.TRAINING_DATA_FILE)
19
20     # divide train and test
21     X_train, X_test, y_train, y_test = train_test_split(
22         data[config.FEATURES], data[config.TARGET], test_size=0.1, random_state=0
23     ) # we are setting the seed here
24
```

On obtient: on peut voir maintenant les deux lignes de logs...

```
20.8,pandas==0.25.3,pluggy==0.13.1,py==1.10.0,pyparsing==2.4.7,pytest=
=5.4.3,python-dateutil==2.8.1,pytz==2020.5,scikit-learn==0.22.2.post1,
scipy==1.6.0,six==1.15.0,wcwidth==0.2.5
regression_model run-test-pre: PYTHONHASHSEED='794'
regression_model run-test: commands[0] | python regression_model/train
_pipeline.py
2021-01-09 03:54:06,752 -- regression_model -- INFO --run_training:31 -- s
aving model version: 0.1.0
2021-01-09 03:54:06,755 -- regression_model.processing.data_management
-- INFO --save_pipeline:33 -- saved pipeline: lasso_regression_output_v0.
1.0.pkl
regression_model run-test: commands[1] | pytest tests
===== test session starts =====
platform win32 -- Python 3.8.1, pytest-5.4.3, py-1.10.0, pluggy-0.13.1
    / pipeline.py
cachedir: .tox\regression_model\.pytest_cache
rootdir: C:\Users\DELL\Desktop\cours-A61\cours-A61\packages\regression
_model
collected 2 items
tests\test_predict.py .. [100%]
===== 2 passed in 1.28s =====
```

Partie-7 Créer un package

```
C:\Users\...| Aide rapide | Accès rapide | Dell\...\cours-A61\cours-A61\packages (main)
(envA61-prod) λ git status
```

On branch main

Your branch is up to date with 'origin/main'.

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

)

modified: regression_model/requirements.txt

modified: regression_model/tox.ini

Untracked files:

(use "git add <file>..." to include in what will be committed)

regression_model/MANIFEST.in

regression_model/setup.py

no changes added to commit (use "git add" and/or "git commit -a")

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages (main)
(envA61-prod) λ git add .
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages (main)
(envA61-prod) λ git status
```

On branch main

Your branch is up to date with 'origin/main'.

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

```
new file: regression_model/MANIFEST.in
modified: regression_model/requirements.txt
new file: regression_model/setup.py
modified: regression_model/tox.ini
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages (main)
(envA61-prod) λ git commit -m "Partie-7-Ajout Package Building"
[main e625786] Partie-7-Ajout Package Building
 4 files changed, 115 insertions(+), 2 deletions(-)
 create mode 100644 packages/regression_model/MANIFEST.in
 create mode 100644 packages/regression_model/setup.py
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages (main)
(envA61-prod) λ git push -u origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 1.95 KiB | 1.95 MiB/s, done.
Total 8 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/cegep2020/cours-A61.git
  1744ad3..e625786 main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

```
C:\Users\DELL\Desktop\cours-A61\cours-A61\packages\regression_model (main)
(envA61-prod) tox
regression_model installed: atomicwrites==1.4.0,attrs==20.3.0,colorama==0.4.4,joblib==0.14.1,more-itertools==8.6.0,numpy==1.18.5,packaging==20.8,pandas==0.25.3,pluggy==0.13.1,py==1.10.0,pyparsing==2.4.7,pytest==5.4.3,python-dateutil==2.8.1,pytz==2020.5,scikit-learn==0.22.2,post1,scipy==1.6.0,six==1.15.0,wcwidth==0.2.5
regression_model run-test-pre: PYTHONHASHSEED='279'
regression_model run-test: commands[0] | python regression_model/train_pipeline.py
2021-01-09 19:25:04,495 -- regression_model.processing.data_management -- INFO -- save_pipeline:33 -- saved pipeline: lasso_regression_output_v0.1.0.pkl
regression_model run-test: commands[1] | pytest tests/
===== test session starts =====
platform win32 -- Python 3.8.1, pytest-5.4.3, py-1.10.0, pluggy-0.13.1
cachedir: .tox\regression_model\.pytest_cache
rootdir: C:\Users\DELL\Desktop\cours-A61\cours-A61\packages\regression_model
collected 2 items

tests\test_predict.py .. [100%]

===== 2 passed in 1.39s =====
summary
regression_model: commands succeeded
congratulations :)
```

```
creating 'dist\regression_model-0.1.0-py3-none-any.whl' and adding 'build\bdist.win-amd64\wheel' to it
adding 'regression_model/VERSION'
adding 'regression_model/__init__.py'
adding 'regression_model/pipeline.py'
adding 'regression_model/predict.py'
adding 'regression_model/preprocessors.py'
adding 'regression_model/train_pipeline.py'
adding 'regression_model/config/__init__.py'
adding 'regression_model/config/config.py'
adding 'regression_model/config/logging_config.py'
adding 'regression_model/datasets/__init__.py'
adding 'regression_model/datasets/test.csv'
adding 'regression_model/datasets/train.csv'
adding 'regression_model/processing/__init__.py'
adding 'regression_model/processing/data_management.py'
adding 'regression_model/processing/errors.py'
adding 'regression_model/processing/features.py'
adding 'regression_model/processing/preprocessors.py'
adding 'regression_model/processing/validation.py'
adding 'regression_model/trained_models/__init__.py'
adding 'regression_model/trained_models/lasso_regression_output_v0.1.0.pkl'
adding 'regression_model-0.1.0.dist-info/METADATA'
adding 'regression_model-0.1.0.dist-info/WHEEL'
adding 'regression_model-0.1.0.dist-info/top_level.txt'
adding 'regression_model-0.1.0.dist-info/RECORD'
removing build\bdist.win-amd64\wheel
summary
install_locally: commands succeeded
congratulations :)
```

```
C:\Users\Del\\Desktop\cours-A61\cours-A61\packages\regression_model (main)
(envA61-prod) \ tox -e install_locally
install_locally create: C:\Users\Del\\Desktop\cours-A61\cours-A61\packages\regression_model\.tox\install_locally
install_locally installdeps: -rrequirements.txt
install_locally installed: atomicwrites==1.4.0,attrs==20.3.0,colorama==0.4.4,joblib==0.14.1,more-itertools==8.6.0,numumpy==1.18.5,packaging==20.8,pandas==0.25.3,pluggy==0.13.1,py==1.10.0,pyparsing==2.4.7,pytest==5.4.3,python-dateutil==2.8.1,pytz==2020.5,scikit-learn==0.22.2.post1,sckipy==1.6.0,six==1.15.0,wcwidth==0.2.5
install_locally run-test-pre: PYTHONHASHSEED='369'
install_locally run-test: commands[0] | python regression_model/train_pipeline.py
2021-01-09 19:38:49,751 — regression_model.processing.data_management — INFO — save_pipeline:33 — saved pipeline: lasso_regression_output_v0.1.0.pkl
install_locally run-test: commands[1] | python setup.py sdist bdist_wheel
running sdist
running egg_info
creating regression_model.egg-info
writing regression_model.egg-info\PKG-INFO
writing dependency_links to regression_model.egg-info\dependency_links.txt
writing requirements to regression_model.egg-info\requires.txt
writing top-level names to regression_model.egg-info\top_level.txt
```

Pour le moment on a juste créé le package, dans la suite on va voir comment distribuer et utiliser ce package.

On va installer le package localement

Vous devez naviguer où le package est créé.

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\regression_model (main)
utiliser ce package..
(envA61-prod) λ pip install -e .
Obtaining file:///C:/Users/Dell/Desktop/cours-A61/cours-A61/packages/regression_model
Requirement already satisfied: numpy<1.19.0,>=1.18.1 in c:\users\dell\desktop\cours-a61\enva61-prod\lib\site-packages (from regression-model==0.1.0) (1.18.5)
Requirement already satisfied: pandas<0.26.0,>=0.25.3 in c:\users\dell\desktop\cours-a61\enva61-prod\lib\site-packages (from regression-model==0.1.0) (0.25.3)
Requirement already satisfied: scikit-learn<0.23.0,>=0.22.1 in c:\users\dell\desktop\cours-a61\enva61-prod\lib\site-packages (from regression-model==0.1.0) (0.22.2.post1)
Requirement already satisfied: joblib<0.15.0,>=0.14.1 in c:\users\dell\desktop\cours-a61\enva61-prod\lib\site-packages (from regression-model==0.1.0) (0.14.1)
```

Ou bien

```
C:\Users\Dell\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ pip install -e packages\regression_model\|
```

Exploitation

```
C:\Users\DELL\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) \ python
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.

>>> import regression_model
>>>
>>> regression_model.__version__
'0.1.0'
>>> |
```

Note : Dans la suite, on va voir qu'est-ce qu'on peut faire avec ce package

Partie 8 : Preparation de Rest-API de service de modèle

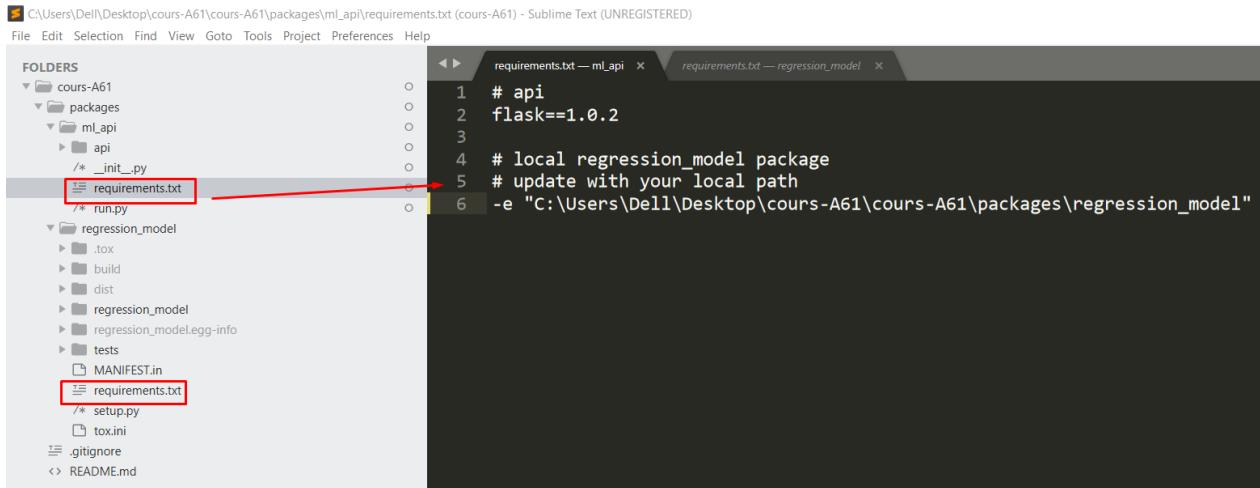
```
C:\Users\DELL\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) \ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory
)
      modified: packages/regression_model/setup.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    packages/ml_api/

no changes added to commit (use "git add" and/or "git commit -a")
```

Vous devez avoir deux fichiers de requirement pour l'API et le modèle.



Installez les modules nécessaires de l'API comme suit :

```
C:\Users\...ml_api(main)
(envA61-prod) \ls
__init__.py api requirements.txt run.py

C:\Users\...ml_api (main)
(envA61-prod) \ pip install -r requirements.txt
```

Maintenant exécutez et vérifiez que tout fonctionne correctement, run app.

Avant, on doit configurer une variable d'environnement pour indiquer à flask , le point d'entrée pour démarrer, comme suit :

```
C:\Users\DELL\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) \ set FLASK_APP=run.py
```

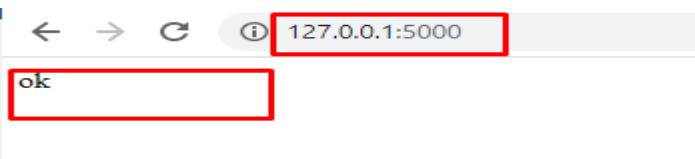
Installez les modules nécessaires de l'API.

Puis run

```
C:\Users\DELL\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) \ python run.py
* Serving Flask app "ml_api" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
         Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git add .
C:\Users\Dell\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git commit -m "Developper- REST API-Flask"
[main 7a00929] Developper- REST API-Flask
7 files changed, 45 insertions(+), 4 deletions(-)
create mode 100644 packages/ml_api/__init__.py
create mode 100644 packages/ml_api/api/__init__.py
create mode 100644 packages/ml_api/api/app.py
create mode 100644 packages/ml_api/api/controller.py
create mode 100644 packages/ml_api/requirements.txt
create mode 100644 packages/ml_api/run.py
C:\Users\Dell\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git push -u origin main
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 4 threads.
```

Sortie :



Partie-9 Ajout de la configuration et de la journalisation et tests à l'API Flask

```
C:\Users\Del\l\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory
)
      modified: packages/ml_api/api/app.py
      modified: packages/ml_api/api/controller.py
      modified: packages/ml_api/requirements.txt
      modified: packages/ml_api/run.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    packages/ml_api/api/config.py
    packages/ml_api/tests/

no changes added to commit (use "git add" and/or "git commit -a")
```

Tester

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) λ ls
__init__.py __pycache__ api requirements.txt run.py tests
    conftest.py
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) λ [pytest tests]
=====
test session starts =====
platform win32 -- Python 3.8.1, pytest-5.4.3, py-1.10.0, pluggy-0.13.1
<> README.md
rootdir: C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api
collected 1 item

tests\test_controller.py . [100%]

=====
1 passed in 0.04s =====
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git add .
C:\Users\Dell\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git commit -m " Ajout- configuration, journalisation et tests au flask api"
[main e552144] Ajout- configuration, journalisation et tests au flask api
8 files changed, 108 insertions(+), 6 deletions(-)
create mode 100644 packages/ml_api/api/config.py
create mode 100644 packages/ml_api/tests/__init__.py
create mode 100644 packages/ml_api/tests/conftest.py
create mode 100644 packages/ml_api/tests/test_controller.py

C:\Users\Dell\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git push -u origin main
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (13/13), 2.29 KiB | 780.00 KiB/s, done.
Total 13 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/cegep2020/cours-A61.git
    7a00929..e552144 main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Partie 10. Ajout de prédiction à l'API Flask

```
Fichier Accueil Insertion Conception Mise en page Références Publipostage Révision
C:\Users\Dell\Desktop\cours-A61\cours-A61 (main)
(envA61-prod) λ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory
)
modified: packages/ml_api/api/controller.py
modified: packages/ml_api/requirements.txt
modified: packages/ml_api/tests/test_controller.py

no changes added to commit (use "git add" and/or "git commit -a")
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) λ ls
__init__.py __pycache__ api logs requirements.txt run.py tests
    / app.py

C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) λ pytest tests\
=====
test session starts =====
platform win32 -- Python 3.8.1, pytest-5.4.3, py-1.10.0, pluggy-0.13.1
    /+ conftest.py
rootdir: C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api
collected 2 items
    /+ run.py
    / regression_model
tests\test_controller.py ...
    / build
=====
[100%] 2 passed in 28.80s =====
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) λ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
(use "git restore --staged <file>..." to unstage)
    modified: api/controller.py
    modified: requirements.txt
    modified: tests/test_controller.py
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) λ git commit -m "Ajout-prédiction-test à notre Flask api"
[main 7234637] Ajout-prédiction-test à notre Flask api
 3 files changed, 49 insertions(+), 2 deletions(-)
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) λ git push -u origin main
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 1.49 KiB | 1.49 MiB/s, done.
Total 9 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/cegep2020/cours-A61.git
  e552144..7234637 main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Partie-11- Ajout de version a notre application flask.

```
C:\Users\Del\l\Desktop\cours-A61\cours-A61\packages\ml_api (ma  
(envA61-prod) ) git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
Changes not staged for commit:  
(use "git add <file>..." to update what will be committed)  
(use "git restore <file>..." to discard changes in working directory)  
)  
modified: api/__init__.py  
modified: api/controller.py  
modified: tests/test_controller.py  
  
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
VERSION  
  
no changes added to commit (use "git add" and/or "git commit -a")
```

Exécuter les tests

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) > ls
VERSION  __pycache__  logs  run.py
__init__.py  api  requirements.txt  tests
/* run.py
VENV_NAME regression_model
response.json
run.py
=====
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) > pytest tests\
===== test session starts =====
platform win32 -- Python 3.8.1, pytest-5.4.3, py-1.10.0, pluggy-0.13.1
    regression_model-0.1.0.tar.gz
rootdir: C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api
collected 3 items
tests\test_controller.py ...
[100%]
===== 3 passed in 2.10s =====
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) > python run.py
2021-01-10 14:59:57,201 — api.app — DEBUG — create_app:18 — Application
instance created
* Serving Flask app "ml_api" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
2021-01-10 14:59:58,580 — api.app — DEBUG — create_app:18 — Application
instance created
* Debugger is active!
* Debugger PIN: 254-811-199
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



```
{  
    "api_version": "0.2.0",  
    "model_version": "0.1.0"  
}
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)  
(envA61-prod) \ git commit -m "Ajout de version a notre api"  
[main 787d5d7] Ajout de version a notre api  
4 files changed, 27 insertions(+)  
create mode 100644 packages/ml_api/VERSION  
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)  
(envA61-prod) \ git push -u origin main  
Enumerating objects: 18, done.  
Counting objects: 100% (18/18), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (9/9), done.  
Writing objects: 100% (10/10), 1.19 KiB | 609.00 KiB/s, done.  
Total 10 (delta 4), reused 0 (delta 0), pack-reused 0  
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.  
To https://github.com/cegep2020/cours-A61.git  
 7234637..787d5d7 main -> main  
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Partie-12 Ajout de Schéma de validation à l'API Flask

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) \ git status
On branch main
Your branch is up to date with 'origin/main'.

  tests
    logs
      test_validation.py
  api
    config.py
    controller.py
    requirements.txt
  tests
    test_controller.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory
)
      modified:   api/config.py
      modified:   api/controller.py
      modified:   requirements.txt
      modified:   tests/test_controller.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      api/validation.py
      test_data_predictions.csv
      tests/test_validation.py
      requirements.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) \ pip install -r requirements.txt
```

```
C:\Users\Dell\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) \ pytest tests
```

```
packages\ml_api\tests\test_controller.py ...
packages\ml_api\tests\test_validation.py .
=====
4 passed in 1.42 seconds =====
```

```
C:\Users\DELL\Desktop\cours-A61\cours-A61\packages\ml_api (main)
(envA61-prod) \ git commit -m "Ajout validation du schéma d'API"
```

```
[main 1124d87] Ajout validation du schéma d'API
```

```
7 files changed, 696 insertions(+), 8 deletions(-)
```

```
create mode 100644 packages/ml_api/api/validation.py
```

```
create mode 100644 packages/ml_api/test_data_predictions.csv
```

```
create mode 100644 packages/ml_api/tests/test_validation.py
```

```
tests
/* __init__.py
C:\Users\DELL\Desktop\cours-A61\cours-A61\packages\ml_api (main)
```

```
(envA61-prod) \ git push -u origin main
```

```
Enumerating objects: 22, done.
```

```
Counting objects: 100% (22/22), done.
```

```
Delta compression using up to 8 threads
```

```
Compressing objects: 100% (13/13), done.
```

```
Writing objects: 100% (13/13), 9.02 KiB | 3.01 MiB/s, done.
```

```
Total 13 (delta 6), reused 0 (delta 0), pack-reused 0
```

```
remote: Resolving deltas: 100% (6/6), completed with 5 local objects.
```

```
To https://github.com/cegep2020/cours-A61.git
```

```
 787d5d7..1124d87 main -> main
```

```
Branch 'main' set up to track remote branch 'main' from 'origin'.
```