Improving the performance of a slow click CLI

Tom Rochette <tom.rochette@coreteks.org>

July 24, 2025 — daae079c

1 Problem

My click CLI is slow, even just to show the help. How do I make it go faster?

2 Solution

In most cases, the reason your click CLI is slow is that you have large imports at the top of the files where you have declared your commands.

The typical pattern is as follows:

cli.py

```
from train import train
from predict import predict

@click.group()
def cli():
    pass

cli.add_command(predict)
cli.add_command(train)
```

train.py

```
import click
import pandas as pd
import torch

@click.command()
def train():
    pass
```

predict.py

```
import click
import pandas as pd
import torch

@click.command()
def predict():
    pass
```

Notice that in both these files we import `pandas` and `torch`, which can account for a large chunk of script execution time simply due to importing them. You can verify that by simply running `python -X importtime` `train.py 2>tuna.log` and using tuna (run `tuna tuna.log`) to inspect the results and convince yourself.

The suggested pattern is to move the imports inside of the function itself, as such:

train.py

```
import click

@click.command()
def train():
    import pandas as pd
    import torch

    pass
```

predict.py

```
import click

@click.command()
def predict():
    import pandas as pd
    import torch

    pass
```

This will shave off a large amount of time spent importing those packages (`pandas` and `torch`). They will only be loaded when you need to run the command itself, not every time you invoke the CLI.

Another pattern which is more complicated is to move the logic of the functions in separate files. This is done to avoid the common mistake that will happen over time that developers will add more logic in those command files, adding imports at the top of the file and slowing the CLI again. By moving the complete implementation to a separate file, you can have the imports at the top of the file and it is not possible to make this mistake again.

train.py

```
import click

@click.command()
def train():
    from train_implementation import train
    train()
```

train_implementation.py

```
import pandas as pd
import torch

def train():
    # Implementation is now here
    pass
```