Metrics tracking with DVC

When conducting different experiments, we need a setup for tracking the metrics resulting from those experiments. DVC provides functionality for this.

First of all, we can define the metrics for our experiment in the `dvc.yaml` file. Much like we specify dependencies and outputs for each stage, we can set specific files that contain metrics and plots for our experiments:

```
stages:
    train:
        cmd: python train.py --config=params.yaml
        deps:
        - features.csv
        outs:
        - model.pkl
        metrics:
        - metrics.json:
            cache: false
        plots:
        - auc.json:
            cache: false
```

In the example above, we specify that `metrics.json` is an output of the `train` stage that contains metrics. When setting `cache: false`, we tell DVC not to track this file but instead keep it as part of our Git history.

Whether or not we want to track metrics files with Git or DVC largely depends on the size of the file. Small files are fine to track in Git, while DVC should track larger files.

DVC provides a few commands specifically for working with metrics files. `dvc metrics show` simply displays the contents of our metrics file in our terminal. For example:

```
dvc metrics show

Path                        f1_score
reports/metrics.json        0.93056
```

```
dvc metrics show --json
{
  "": {
    "data": {
      "reports/metrics.json": {
        "data": {
          "f1_score": 0.9305555555555555
        }
      }
    }
  }
}
```

We can use different flags to specify which metric specifically we would like to show.

Another useful command is `dvc metrics diff`. This command displays the differences in metrics between different experiments, which allows you to quickly compare the performance of two experiments. By default, DVC shows a comparison of the previous two experiments, but you can also specify the commit hashes of the experiments you would like to compare.

```
dvc metrics diff
```

```
Path          Metric    HEAD    workspace  Change
metrics.json  AUC       0.9643  0.9671     0.0028
metrics.json  TP        527     531        4
```