6002B Machine Learning CW Setup Command Line MacOs/Linux

Getting started with tsml using Git command line and Intellij

This guide is written for users to download the **tsml** code base and switch to the **ml6002b-coursework** branch using git. It is intended for users to follow if they want to make sure they are downloading the code correctly or are new to git/GitHub. If you are already familiar with git/GitHub, then feel free to follow your usual workflow to clone tsml and switch to the ml6002-coursework branch.

Pre-requisites

You need to:

1. Have a GitHub account. You can use an existing account if you have one or you can sign up at https://github.com/join
2. Have Git installed which you can find instruction on how to do it here https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

The screenshots provided may not look exactly like your terminal. This is fine, it doesn't matter how your terminal looks. You will want to put the commands specified after the $ in your terminal. In addition the commands will be provided in text so you can copy and paste them into your terminal (ctrl v may not work for pasting into terminal so you may need to right click and click paste them manually into terminal).

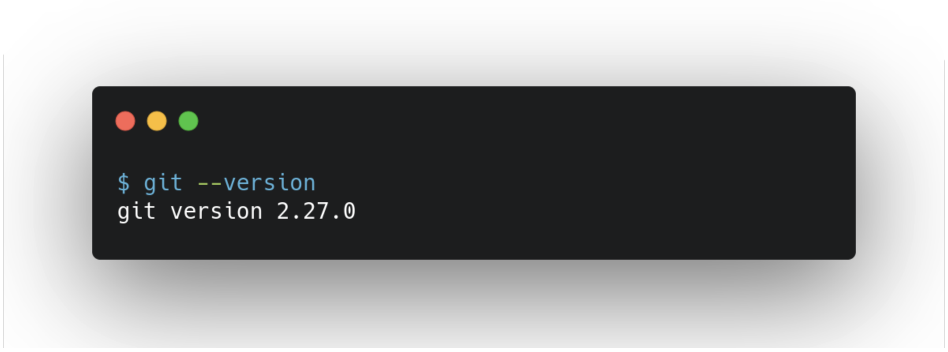
Setting up git with tsml

1) First open terminal

2) Check your git version by typing

```
git --version
```

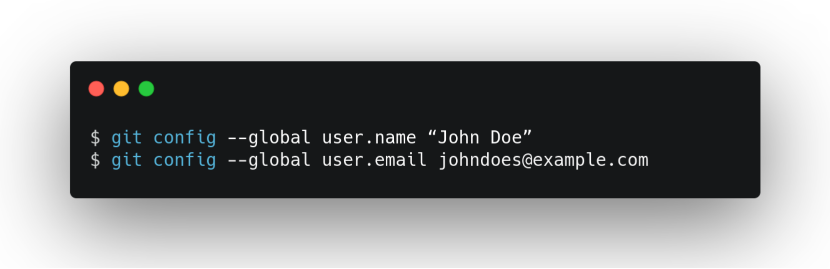
This should give output of your current version below (it doesn't matter what the version is and don't worry if it isn't the same as mine). If you want to update your git google it.



```
$ git --version
git version 2.27.0
```

3) Now we will set your identity. Your identity is important because every Git commit uses this information and it’s immutably baked into the commits you start creating. To do this we’re going to enter the following commands but change the name from John Doe to your name and johndoe@example.com to your email address (ideally the one you set your github account up with).

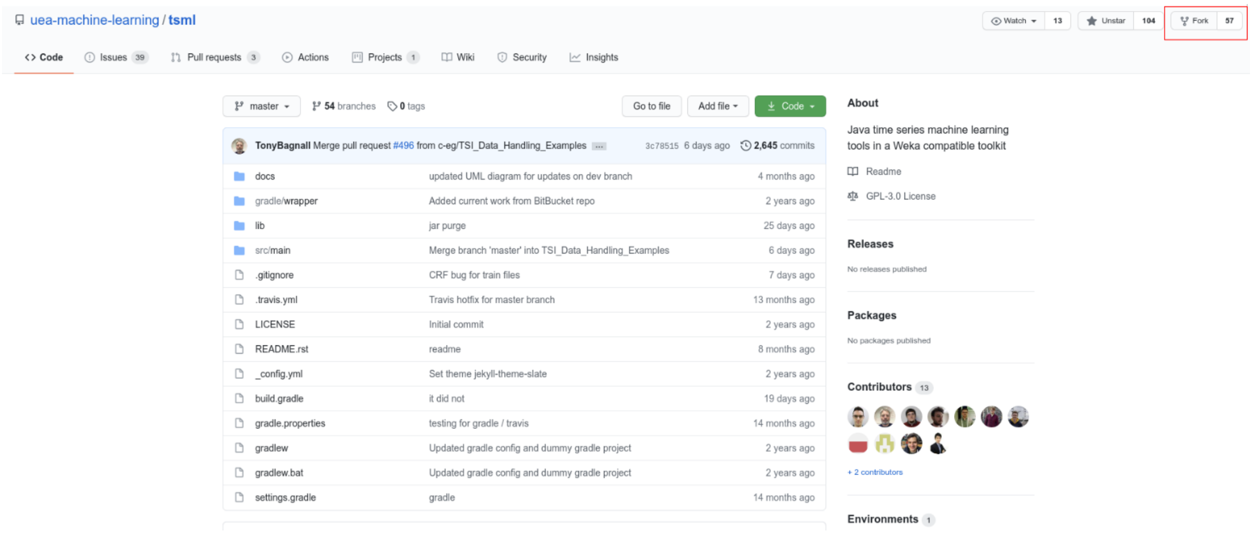
   git config --global user.name “John Doe”
   git config --global user.email johndoes@example.com



```
$ git config --global user.name “John Doe”
$ git config --global user.email johndoes@example.com
```

4) This next step is **optional** but will make it so you don’t need to sign into GitHub every time you want to do any pull or pushes to a repository. There are two ways of doing this. The first (probably the easier method) is to cache your credentials so that you don’t need to sign in every time. Instruction to do this can be found here: https://docs.github.com/en/github/using-git/caching-your-github-credentials-in-git.
   The second method involves setting up an ssh key to Github and instructions for that can be found here (this is a better long term and more secure solution).
   https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh

5) Now we will fork the tsml repository. Go to the tsml GitHub page found here:
   https://github.com/uea-machine-learning/tsml



TonyBagnall Merge pull request #496 from c-eg/TSI_Data_Handling_Examples … 3c78515 6 days ago 2,645 commits

docs	updated UML diagram for updates on dev branch	4 months ago
gradle/wrapper	Added current work from BitBucket repo	2 years ago
lib	jar purge	25 days ago
src/main	Merge branch 'master' into TSI_Data_Handling_Examples	6 days ago
.gitignore	CRF bug for train files	7 days ago
.travis.yml	Travis hotfix for master branch	13 months ago
LICENSE	Initial commit	2 years ago
README.rst	readme	8 months ago
_config.yml	Set theme jekyll-theme-slate	2 years ago
build.gradle	it did not	19 days ago
gradle.properties	testing for gradle / travis	14 months ago
gradlew	Updated gradle config and dummy gradle project	2 years ago
gradlew.bat	Updated gradle config and dummy gradle project	2 years ago
settings.gradle	gradle	14 months ago

About

Java time series machine learning tools in a Weka compatible toolkit

Readme

GPL-3.0 License

Releases

No releases published

Packages

No packages published

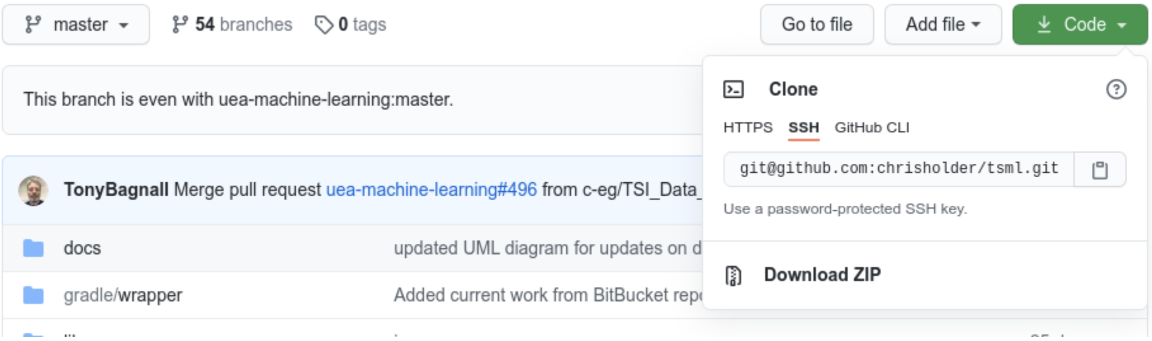
Contributors 13

+ 2 contributors

Environments 1

Click the ‘Fork’ button (should be top right) as seen in the screenshot above. You may be promoted asking where the repository should be forked, click on your account. This should then open the repository but it will be under your personal repositories.

6) Now the repo is forked we can clone it for local development. To do this for the repository you just forked (you can validate you’re on the correct repository as in the top left it should say ‘<your username>/tsml’) click the ‘Code’ button shown below.



This branch is even with uea-machine-learning:master.

TonyBagnall Merge pull request uea-machine-learning#496 from c-eg/TSI_Data_

docs	updated UML diagram for updates on d
gradle/wrapper	Added current work from BitBucket rep

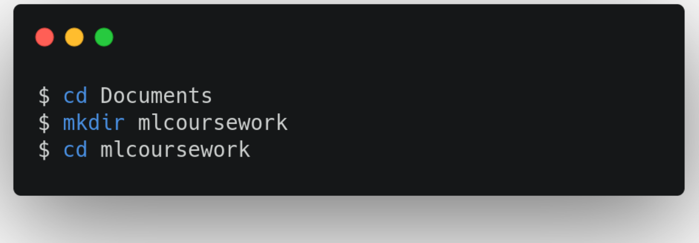
```
git@github.com:chrisholder/tsml.git
```

Use a password-protected SSH key.

As shown this should bring up a drop down. The link you copy will depend on if you setup SSH or not. If you setup SSH (outlined in step 4) go to the SSH tab and click the clipboard button to copy the url. If you didn’t set up ssh click the HTTPS option and click the clipboard button to copy the url.

7) Before we clone the repository locally we’re going to create a directory for our ML coursework to exist in. I decided to create a directory in my Documents and cd into it using the following commands

```
cd Documents
mkdir mlcoursework
cd mlcoursework
```



```
$ cd Documents
$ mkdir mlcoursework
$ cd mlcoursework
```

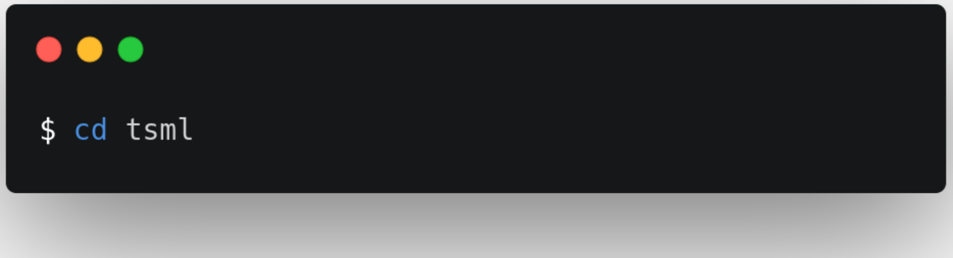
8) Now that we're inside our ‘mlcoursework’ directory we can clone the repository using the command below and pasting the https or ssh link we got from the forked repository above. You may be promoted for your github username and password. If so just enter them and hit enter.

```
git clone <your link goes here>
```

```
$ git clone https://github.com/username/tsml.git
```

9) You’ll probably see a few lines of output and an increasing percentage. To complete the clone it may take a minute or two depending on your internet connection. Once completed we should be able to cd into the directory like so

```
cd tsml
```



```
$ cd tsml
```

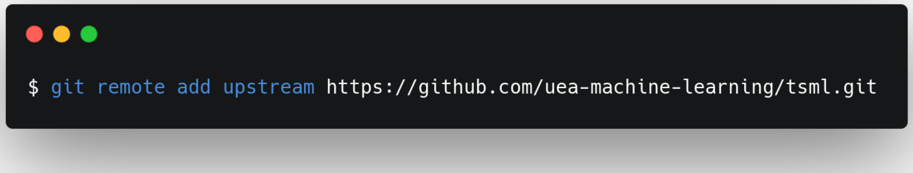
10) We are now inside the directory that contains all of the code for tsml. What we're now going to do is switch to a branch you can develop your coursework on.
GitHub repositories can have multiple branches. The idea behind branches is that different features can be developed simultaneously for the same code base without changes clashing with one another. The simplest projects having a single branch called master (this is being phased out and renamed as main in new repositories, however). A branch can be made that uses main/master, allowing a user to change the code in the branch without affecting the base. Once all changes are completed, the new branch can be merged back into main/master to update it with the new features, and then the separate branch is closed.

You do not need to know too much about branches for the coursework assignment except that there is a specific branch that has been made for the assignment. This branch has been setup so you can easily use and develop your coursework using tsml

Switching to this branch will update the code that is saved on your machine to match what is in the ml6002b branch, It is also good practice to pull code regularly to make sure that the base code is up to date - instructions to do this will be shown below. Please note that you will never need to push anything to the tsml project/ml6002bcoursework branch – your work should be done in private and you should just use the code from this branch as a base.

11) Before we switch we're going to make sure we have the latest version of tsml. To do this we need to tell git where the original repository is (this is called setting an upstream). To do this enter the following command

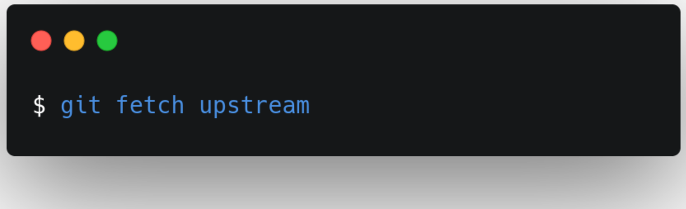
git remote add upstream https://github.com/uea-machine-learning/tsml.git



```
$ git remote add upstream https://github.com/uea-machine-learning/tsml.git
```

12) We can now check we have the most up to date version by running the following. You should run this command everytime you come back to the project to make sure you have the latest version

    git fetch upstream

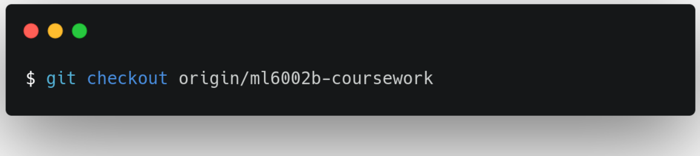


```
$ git fetch upstream
```

    You may see a long output which may include lots of 'new branch' output. This means we've synced all the active branches and code currently on tsml

13) We can now switch to the ml6002b-coursework branch which is setup for the coursework. To do this enter the following command

    git checkout origin/ml6002b-coursework



```
$ git checkout origin/ml6002b-coursework
```

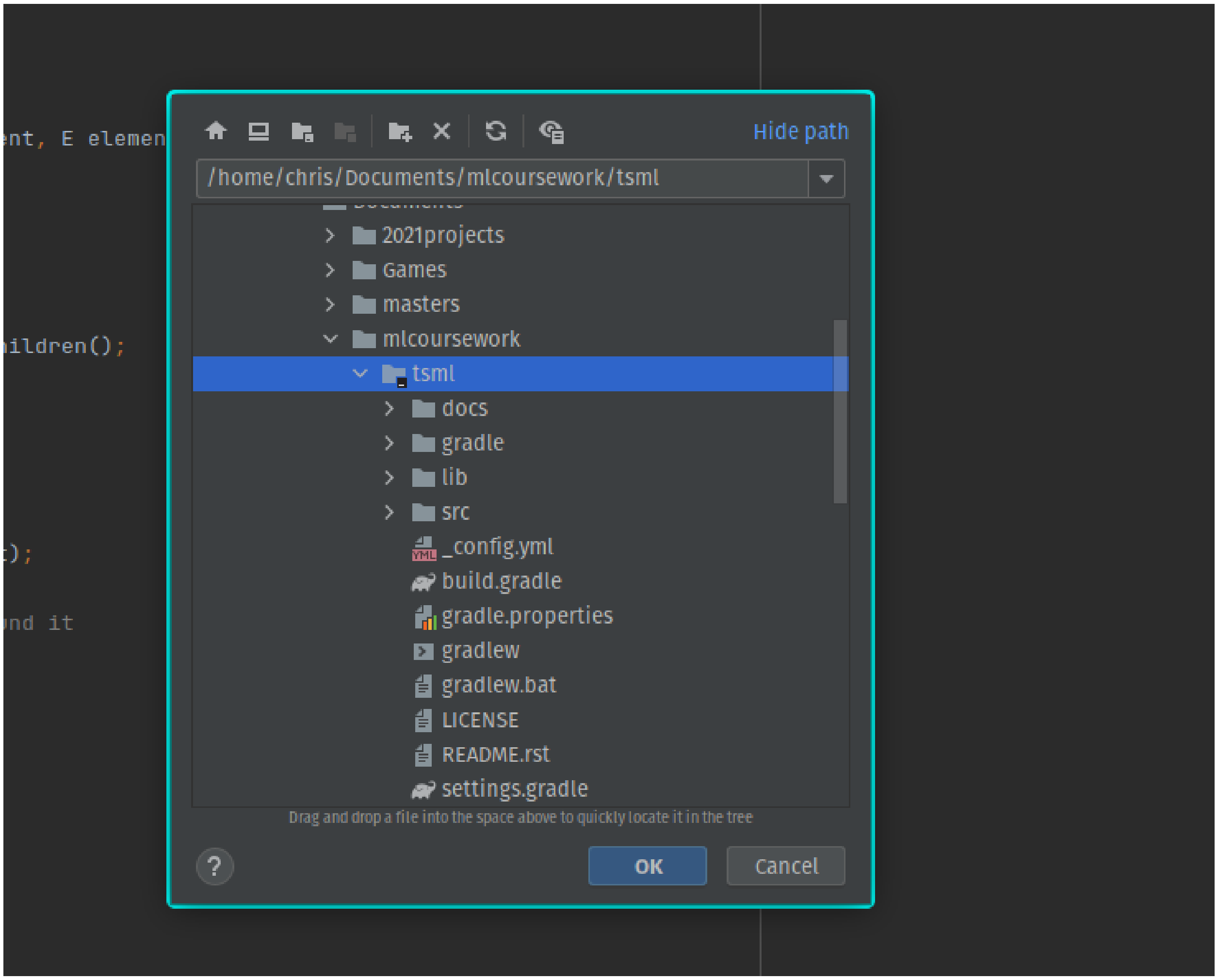
NOTE: we can switch to any of the active branches by doing git checkout origin/<branch name here>.

14) This will update all our local files to be in on the ml6002b-coursework branch so we can now open IntelliJ and begin development

Setting up IntelliJ for development

NOTE: You don’t have to use IntelliJ and are free to use whatever environment you want but the steps below are only for IntelliJ and you’ll have to setup any other IDE on your own.

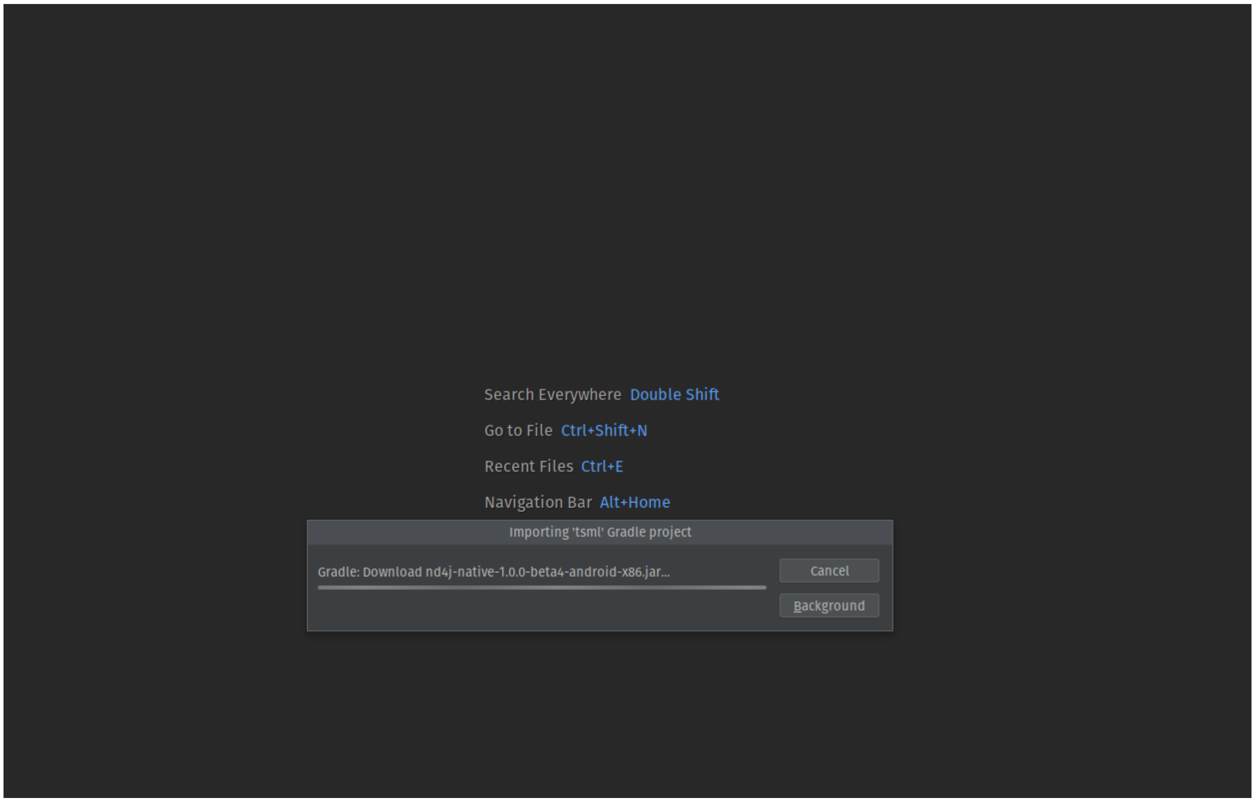
1) Open the tsml directory in IntelliJ. If you followed the steps above it should be in Documents/mlcoursework/tsml. Open it in the root directory (i.e. select tsml dir as shown)



/home/chris/Documents/mlcoursework/tsml

You may get a prompt asking if you trust this project just say yes to anything like this.

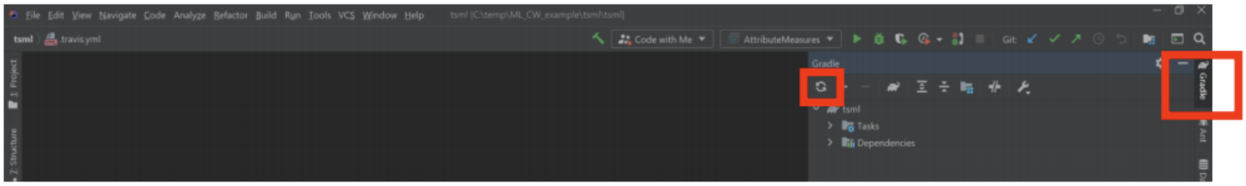
2) After opening you may need to wait a minute or two while all the libraries are downloaded by Gradel - It will looking something like this



Importing 'tsml' Gradle project

Gradle: Download nd4j-native-1.0.0-beta4-android-x86.jar...

3) Once completed the middle loading bar will disappear. To confirm the installation it is also a good idea to press the Gradle button and rebuild (this is required sometimes but will likely not do anything extra for you)

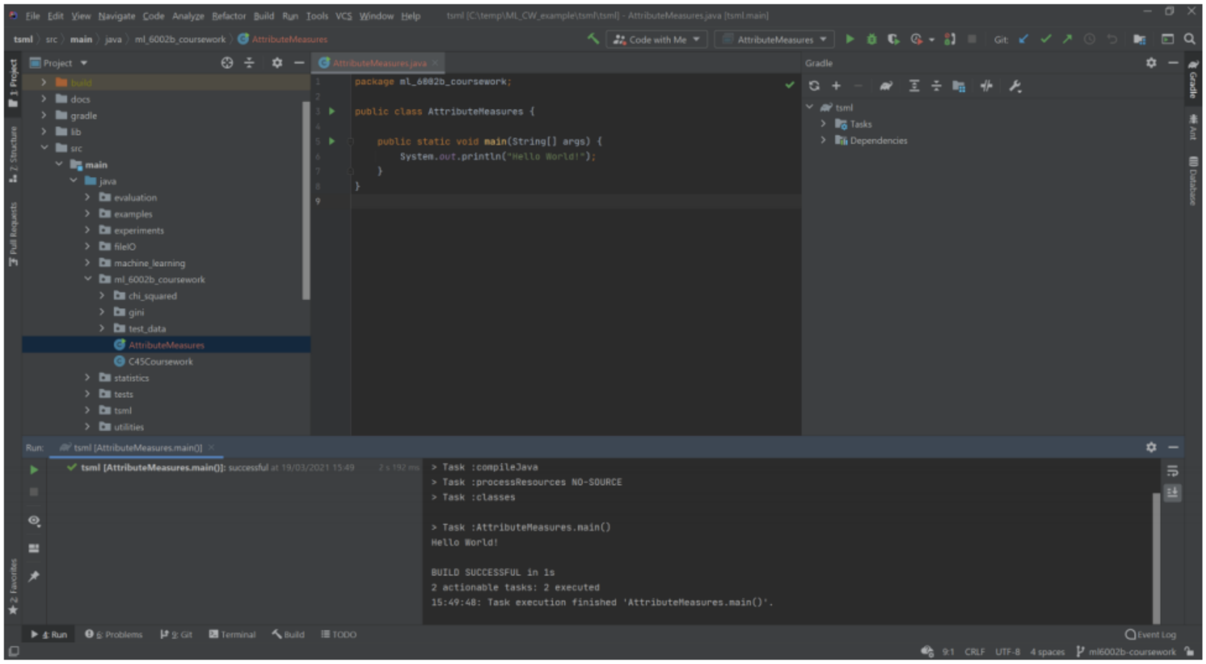

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
tsml [C:\temp\ML_CW_example\tsml\tsml]
tsml
travis.yml
Code with Me
AttributeMeasures
Git:
1: Project
2: Structure
Gradle
tsml
Tasks
Dependencies
Gradle
Ant

4) We should be ready to code now. You will find the package that you need for the CW under **src -> main -> java -> ml_6002b_coursework**.

If the ml_6002b_coursework please scroll to the bottom of the doc and see instructions on how to get it to appear

To test the installation in the screenshot below I have made an AttributeMeasures class and just included a simple “Hello World!” main method to check the installation.

The first time the build will take a while to compile (approx 1-2 mins) so don’t panic if it takes time or seems slow. Subsequent runs should be much faster once the initial run has completed as a number of one-off background operations will not be required again.



```
package ml_6002b_coursework;

public class AttributeMeasures {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

tsml [AttributeMeasures.main()]: successful at 19/03/2021 15:49

```
> Task :compileJava
> Task :processResources NO-SOURCE
> Task :classes

> Task :AttributeMeasures.main()
Hello World!

BUILD SUCCESSFUL in 1s
2 actionable tasks: 2 executed
15:49:48: Task execution finished 'AttributeMeasures.main()'.
```

5) Once you have gotten this far you will now be ready to start implementing the coursework assignment - good luck!

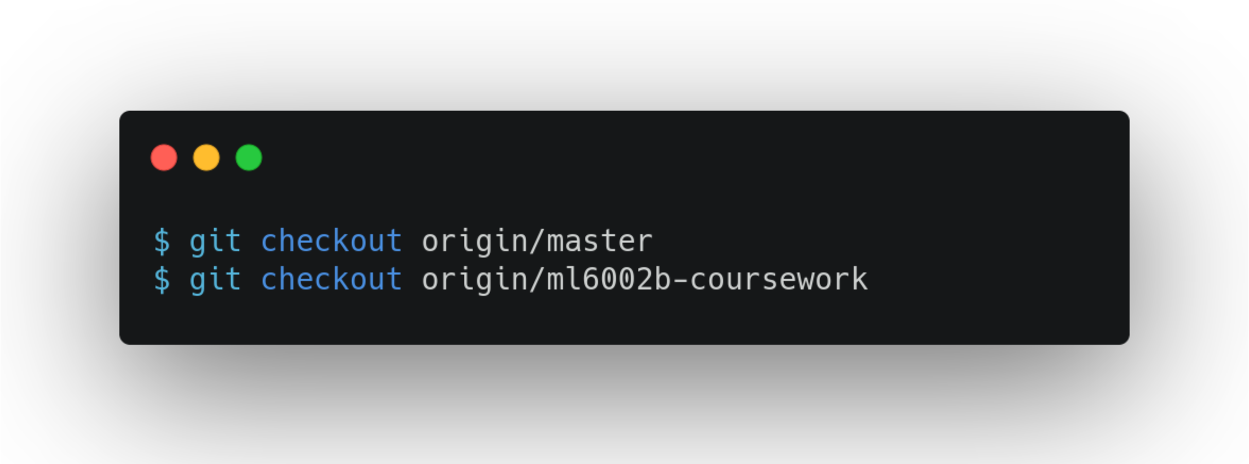
Missing ml_6002b_coursework directory

If you came to open the directory and it wasn't there don't panic this is probably nothing you've done wrong just git being weird about merges and branches. To fix this we just need to checkout to master by doing

```
git checkout origin/master
```

And then come back to the branch

```
git checkout origin/ml6002b-coursework
```



```
$ git checkout origin/master
$ git checkout origin/ml6002b-coursework
```

If you see any warnings when doing this just ignore them. After you do this the directory should appear.