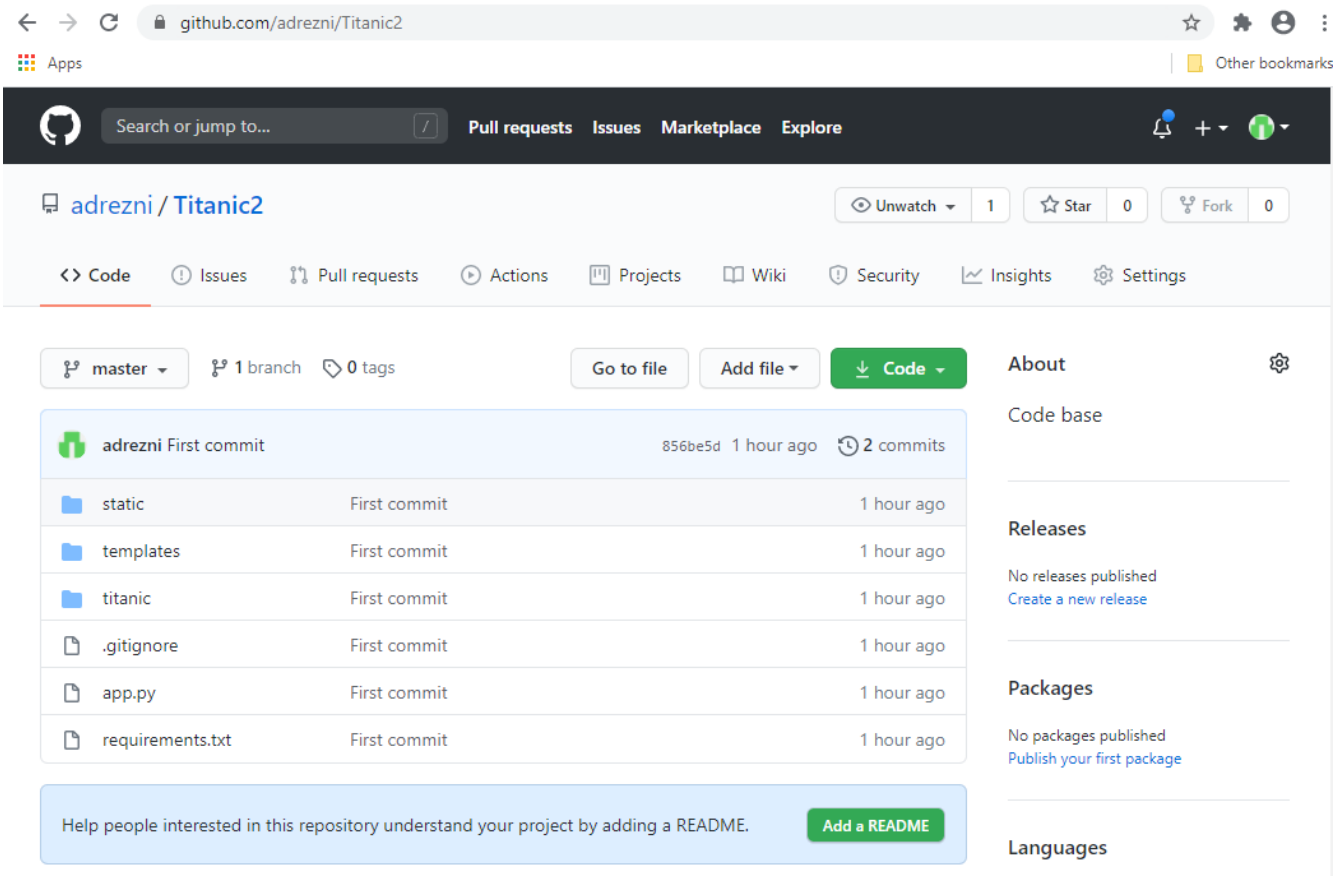
Installation of Demo Titanic2 on Ubuntu

Download project from GitHub

Use the URL: github.com/adrezni/titanic2 and click on the green download button labeled Code. This will download the project as a zip file.



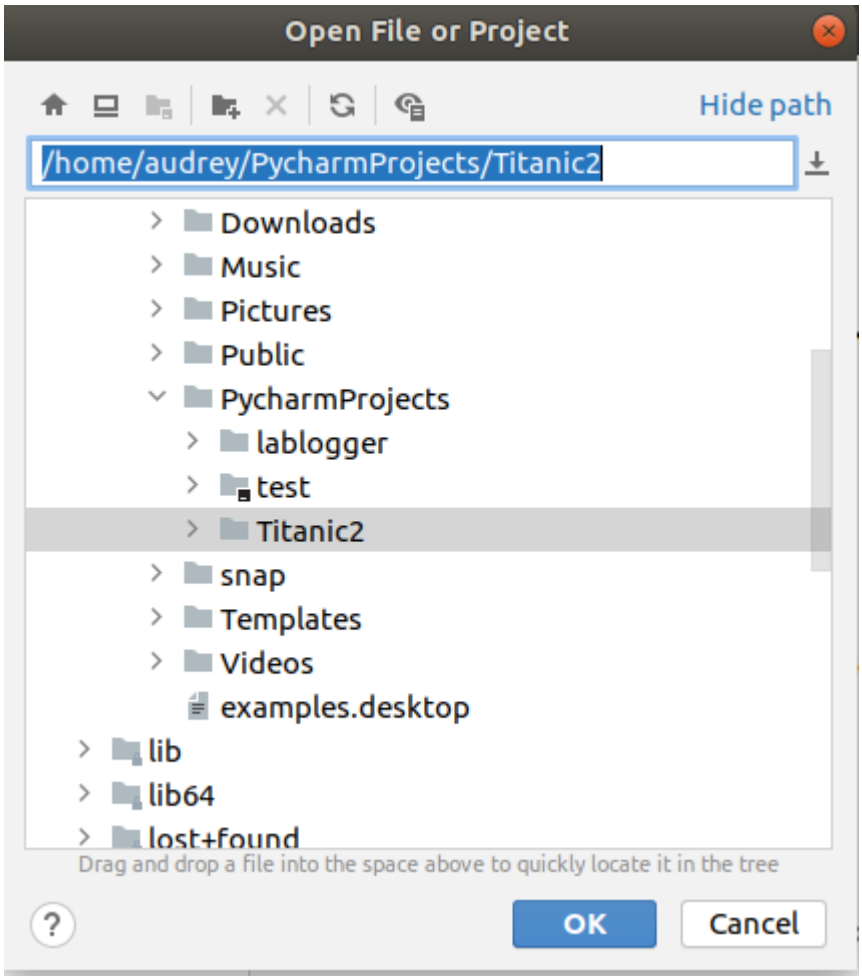
Unzip project to some project folder

Extract the project zip file to a folder where you like to place your PyCharm projects. Once the project is extracted, the unexpanded project looks like:

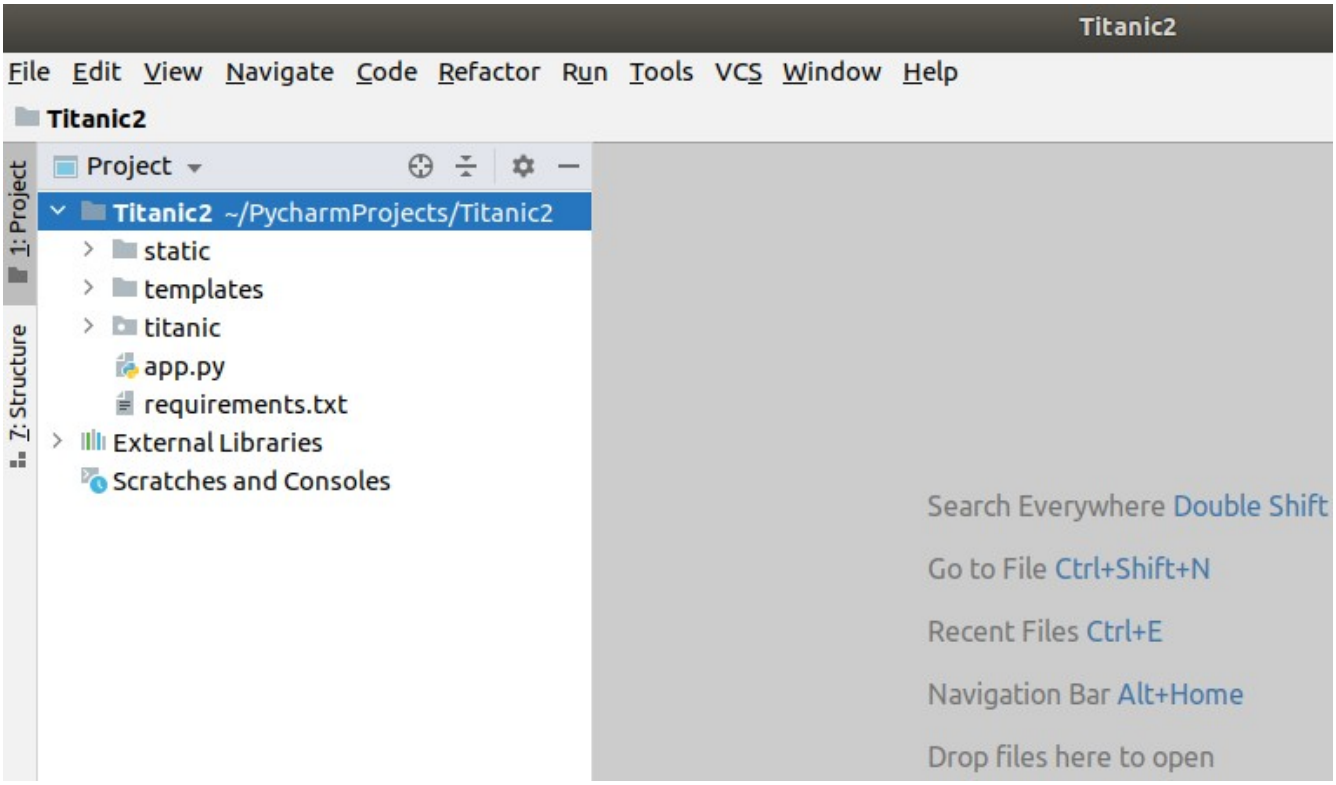
Configure Virtual Environment in PyCharm

This configuration tutorial assumes that you already have Anaconda's Python interpreter Version 3.6 installed. The application, Titanic2 will work with any version of 3.6 or above. The Python interpreter is typically installed in ***/usr/bin***.

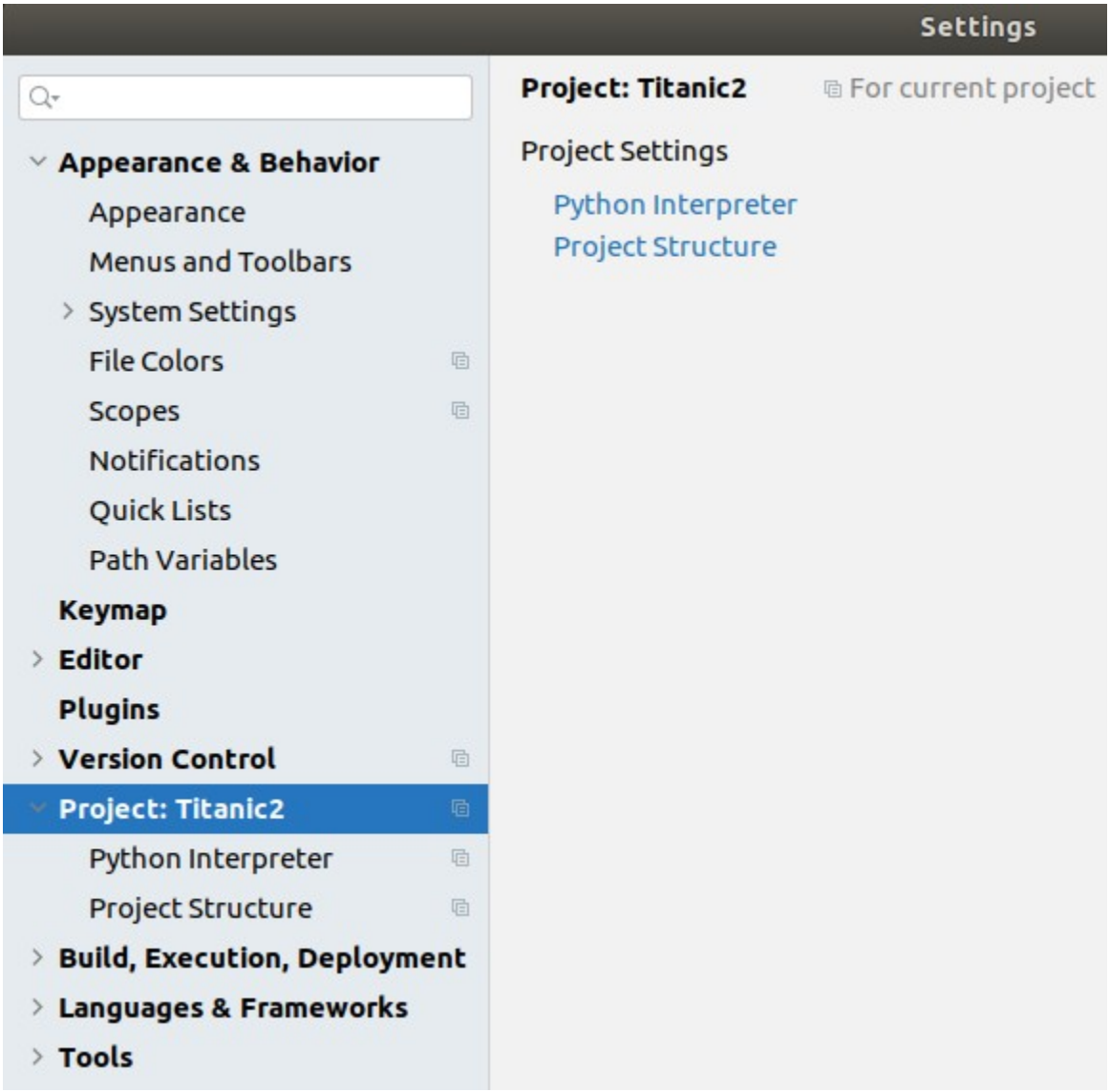
The file, **requirements.txt** contains the version numbers of all the package dependencies for this application. We will next create a Virtual Environment that will contain the dependencies. Start PyCharm and choose to open a project. The Open Project dialog will appear.



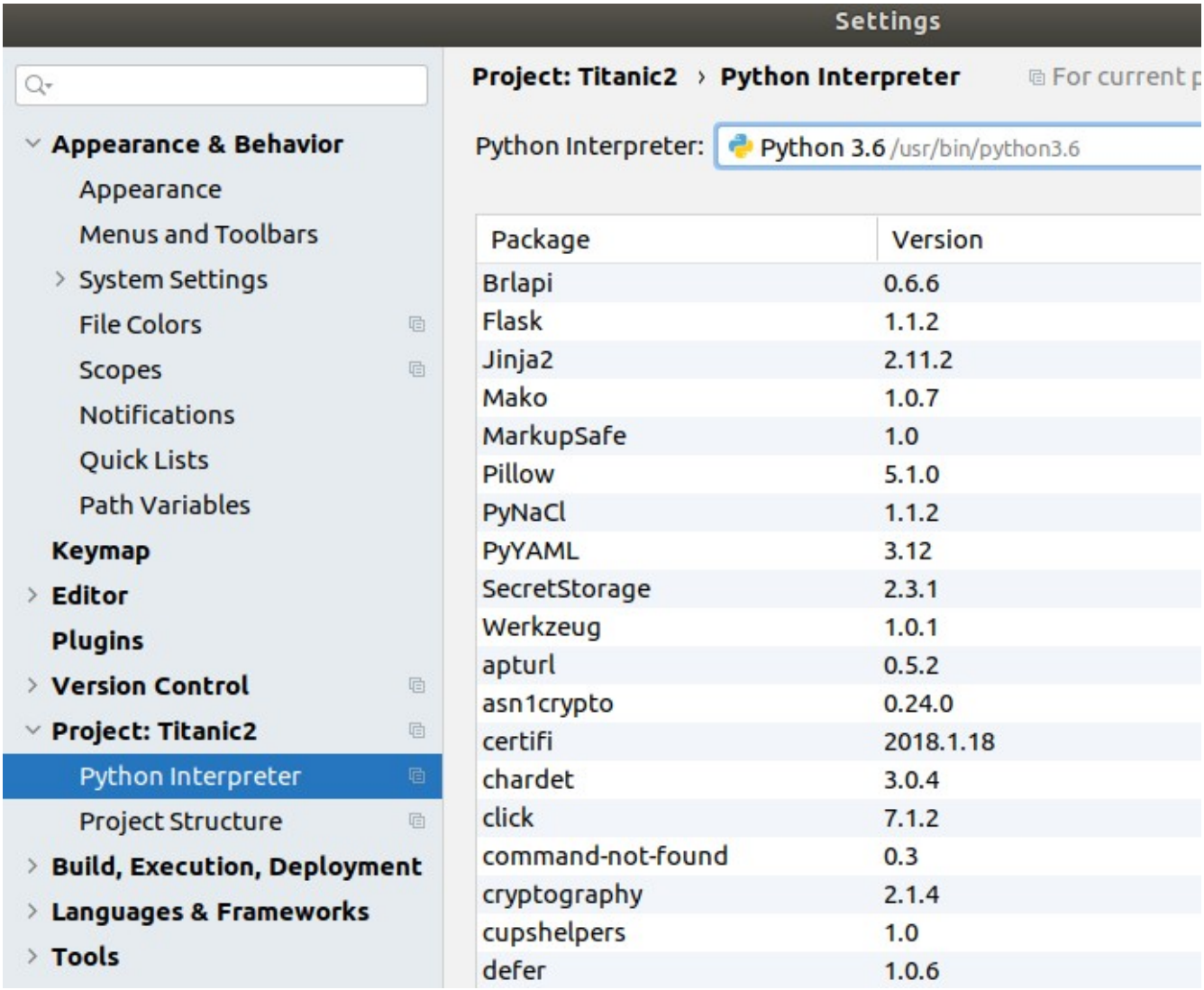
Notice that the name of the project is Titanic2. Once the project is opened in PyCharm, the Project view appears as below:

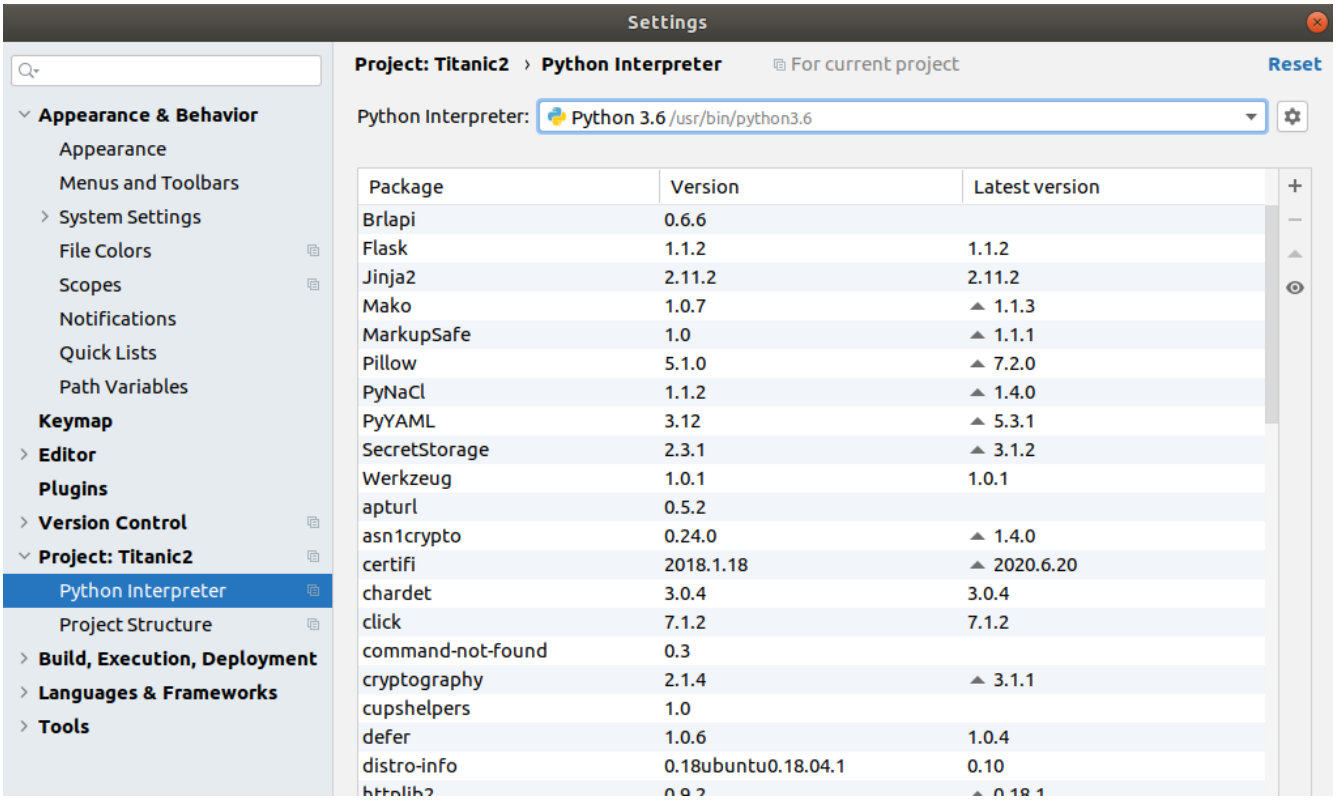


Now that the project is open, we can start the configuration process that will create a Virtual Environment that contains all the dependencies that the project needs. Use the menu selection: "File | Settings". That will open a dialog, and we will see a column on the left side of the dialog with some attributes. Find the attribute labeled **Project: Titanic2**, and expand this attribute.



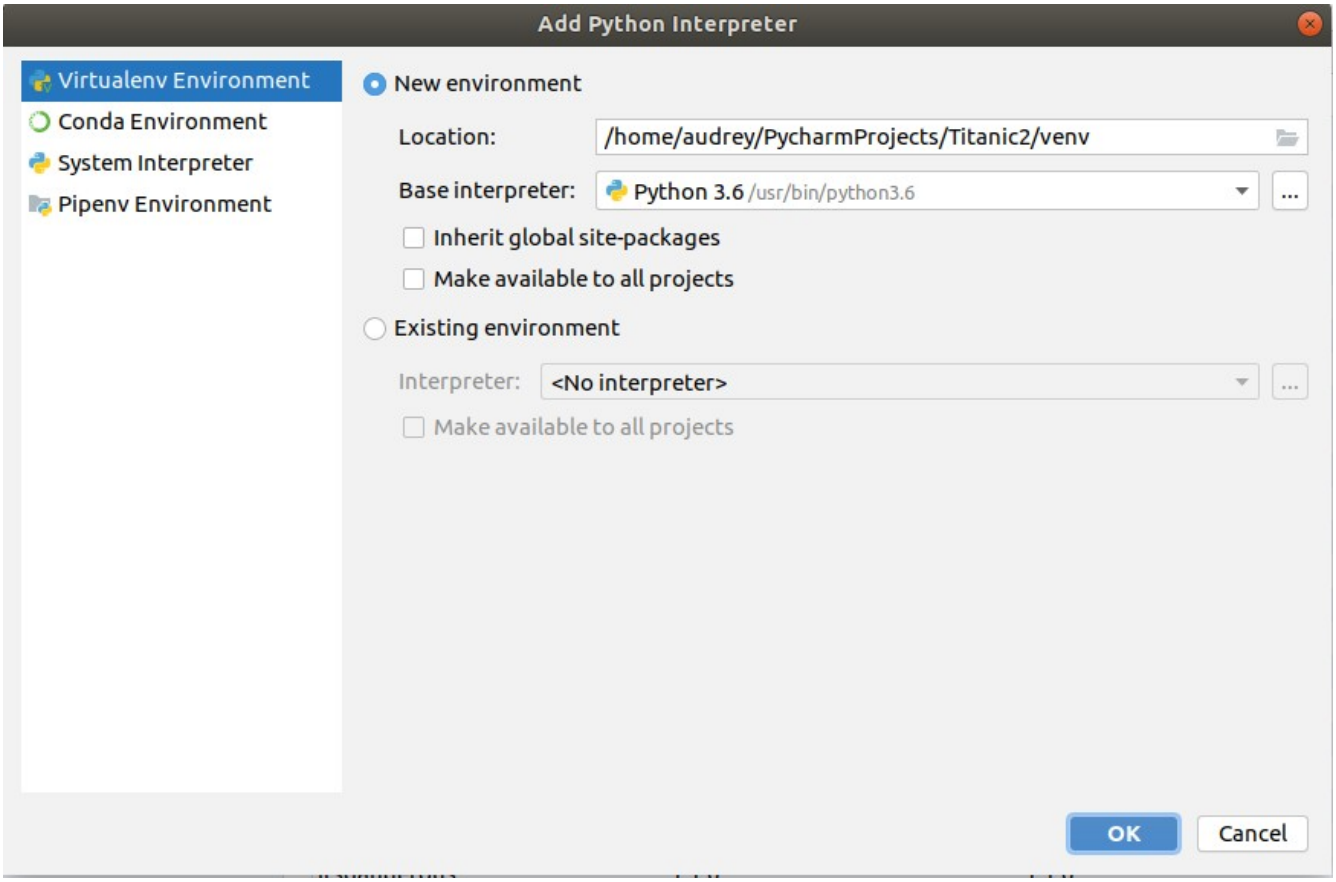
Select the attribute **Python Interpreter** under Project: Titanic2. This will populate the dialog with a pulldown that is labeled **Project interpreter.** If the Project interpreter is not pointing to an installed Python3.6, expand the pulldown and your installed Python3.6 should appear.


Settings
Project: Titanic2 › Python Interpreter
For current p
Python Interpreter: Python 3.6 /usr/bin/python3.6
Appearance & Behavior
Appearance
Menus and Toolbars
System Settings
File Colors
Scopes
Notifications
Quick Lists
Path Variables
Keymap
Editor
Plugins
Version Control
Project: Titanic2
Python Interpreter
Project Structure
Build, Execution, Deployment
Languages & Frameworks
Tools
Package
Version
Brlapi 0.6.6
Flask 1.1.2
Jinja2 2.11.2
Mako 1.0.7
MarkupSafe 1.0
Pillow 5.1.0
PyNaCl 1.1.2
PyYAML 3.12
SecretStorage 2.3.1
Werkzeug 1.0.1
apturl 0.5.2
asn1crypto 0.24.0
certifi 2018.1.18
chardet 3.0.4
click 7.1.2
command-not-found 0.3
cryptography 2.1.4
cupshelpers 1.0
defer 1.0.6

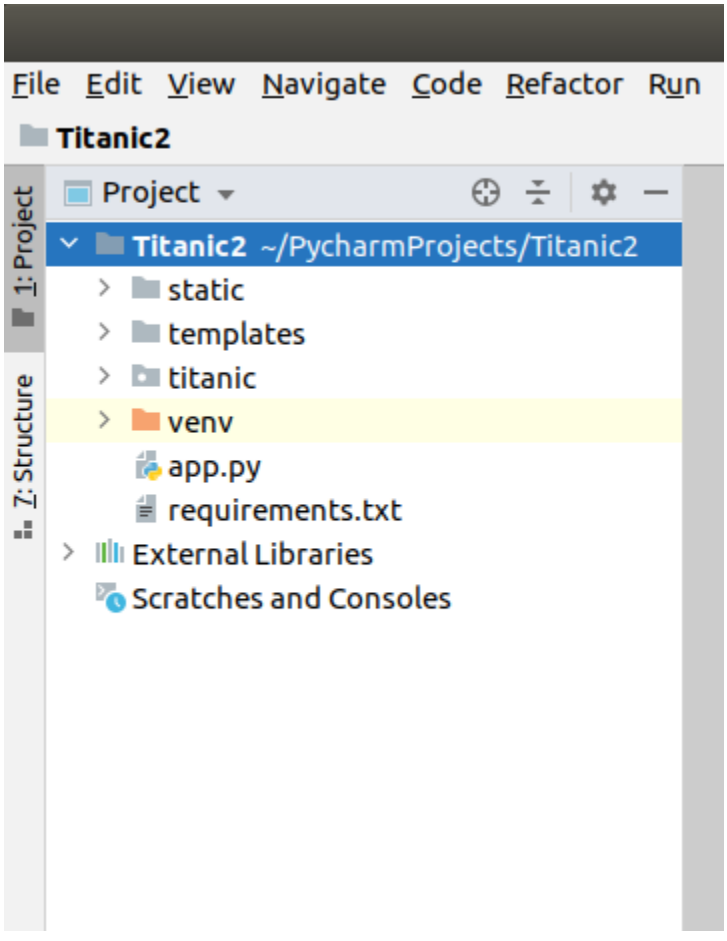


We are now ready to begin configuration of the Virtual Environment. There is a gear icon to the right of the Python interpreter pulldown. Click on the gear and select “Add”.

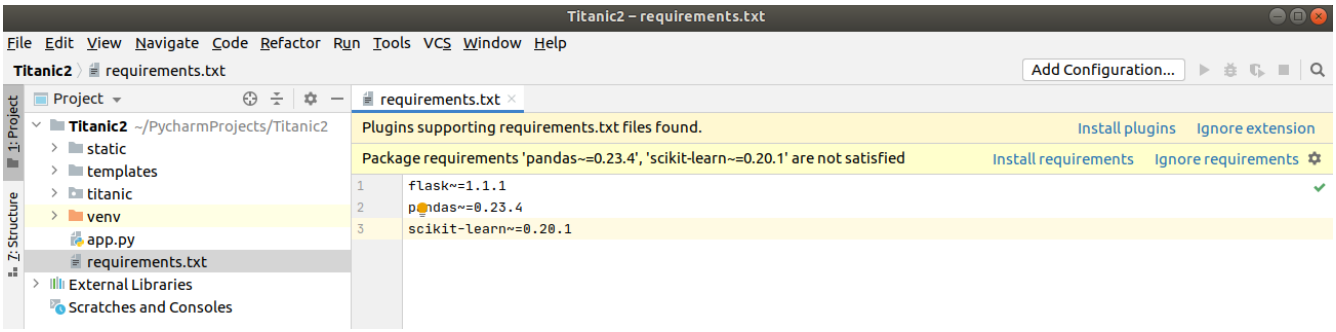
This brings up a dialog that will add a **venv** folder to the project that will contain all the Virtual Environment packages. Select the radio button labeled “New Environment”. Highlight the selection “Virtualenv Environment”.



Click on the OK button to add the **venv** folder. And then click on the OK button to dismiss the Settings dialog. Once the Settings dialog is dismissed, look at the Project explorer on the left side of PyCharm:

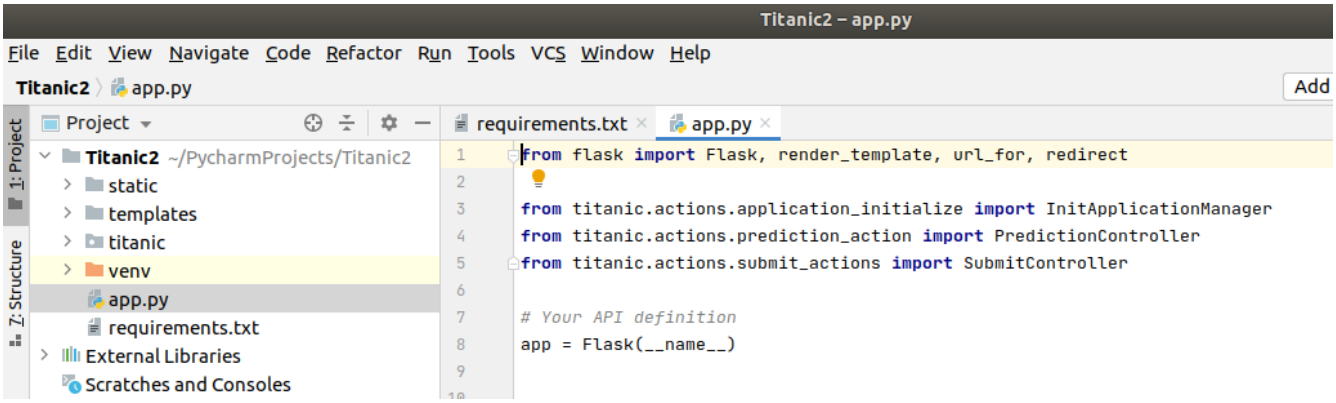


Notice that a new folder, **venv** has been created in the project tree. Now we will add the packages that the project depends on. Do this by opening the file, **requirements.txt** by double clicking on it. This will open the requirements file in the editor screen to the right and above the editor will appear a link to Install requirements.



There are two action links above the text contained in requirements.txt. For now, ignore the first one that is labeled "Install Plugins", but do click on "Install requirements". The package installation will initiate. This installation takes a few seconds.

If all the required packages were installed successfully, then if you open the Python file, app.py, you should see no errors in the **import flask** line.



You can also confirm that pandas and sklearn were installed by opening the Python file, /titanic/*datamanagement*/data_manager.py and confirm no errors.