**Documentation**

**How to create an environment in which any code will run**:

The venv module provides support for creating lightweight virtual environments with their own site directories, optionally isolated from system site directories. Each virtual environment has its own Python binary (which matches the version of the binary that was used to create this environment) and can have its own independent set of installed Python packages in its site directories.

## **Creating virtual environments:**

python3 -m venv /path/to/new/virtual/environment

Running this command creates the target directory.When a virtual environment is active, the VIRTUAL\_ENV environment variable is set to the path of the virtual environment. This can be used to check if one is running inside a virtual environment.

**How to prepare the code for running**

First step is to create jupter notebook with python version then we have create a scikit-learn notebook in which different commands are running for predicting the result with different scikit-learn algorithms. When we are done with that we have to create another notebook in which we have to prepare Deutsch’s algorithm using qiskit. Here we have made end result that came by using the above algorithm.

After that we have to download the file in the .ipynb format so that it can directly run from the terminal and for the user it is easy to open.

**How to run the code including settings:**

We have made readme.md file in which we have mentioned the purpose of the repoistory and why are using the repoistory and how to run the notebooks.

After we have made the .ipynb file we have made a docker file for these notebook and we have made a separate folder so that when we run the code it should run with minimal configuration using docker file.

**Expectations upon running the code:**

When we are running these notebooks:

1.In first notebook we have predicted petrol consumption using scikit-learn algorithms.

2.We have find the output that came after using these three algorithms of scikit-learn.

3.In the second notebook we have used quantum computing.

4.Here comparison between quantum computing and classical computing done.

5.Then we have performed Deutsch’s algorithm using qiskit.

6.Then a graph is plotted for the Deutsch’s algorithm final result.

**Concise descriptions of README.**

# **Why Repository exists?**

In this repository we have discussed about two jupyter notebooks and these two jupyter notebook contains scikit-learn notebook and quantum notebook.

In the first notebook we have performed some operations on dataset using scikit-learn library.In the second notebook we have performed Deutsch’s algorithm using qiskit

**How to run the notebook?**

Open terminal

Add command(Jupyter notebook)

Then choose python from top right menu

Then jupyter notebook will be opened.

Jupyter files have been provided in the above folder.

**How to run docker file?**

**1. Write docker file:**

FROM jupyter/scipy-notebook

RUN pip install scikit-learn

RUN pip3 install qiskit

RUN pip3 install pylatexenc

RUN pip3 install plotly

WORKDIR /app

ADD . /app

**2. Create docker image from docke file:**

$sudo docker build -t project -f Dockerfile

**3.To run docker image:**

**sudo docker run project python3 Scikit-learn.py**

$sudo docker run project python3 quantum-deutsch.py