Here's a step-by-step guide to create and deploy a machine learning project using the breast cancer dataset from sklearn:

1. **Create a folder structure for the project**:

```
ml_project/ |-- src/ | |-- __init__.py | |-- logger.py | |-- exception.py | |-- components/ | |-- __init__.py | |-- data_ingestion.py | |-- data_transformation.py | |-- model_trainer.py | |-- pipelines/ | |-- __init__.py | |-- predict_pipeline.py | |-- train_pipeline.py |-- import_data.py |-- setup.py |-- notebooks/ |-- requirements.txt |-- README.md |-- .gitignore
```

2. **Initialize a git repository and add necessary files**:

```
cd ml_project git init touch README.md touch .gitignore
```

3. **Create a separate environment**:

```
# Create and activate a new virtual environment python3 -m venv env source env/bin/activate
```

4. **Install necessary dependencies**:

```
# Install required packages pip install numpy pandas scikit-learn
```

5. **Write the program to import data from sklearn**:

```
# import_data.py from sklearn.datasets import load_breast_cancer def load_dataset(): return load_breast_cancer() if __name__ == "__main__": data = load_dataset() print("Data loaded successfully!")
```

6. **Write the setup.py for package installation**:

```
# setup.py from setuptools import setup, find_packages setup( name="ml_project", version="0.1", packages=find_packages(), )
```

7. **Write the logging function in logger.py**:

```
# logger.py import logging def setup_logger(): logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s') def get_logger(): return logging.getLogger(__name__)
```

8. **Write the exception function in exception.py**:

```
# exception.py class MLException(Exception): pass
```

9. **Generate egg-info folder**:

```
python setup.py egg_info
```

10.**Update git repository and add necessary files**:

```
git add . git commit -m "Initial commit" git remote add origin
<repository-url> git push -u origin master
```

11.**Add additional files from GitHub**:

- RADM.md - LICENSE

12.**Pull changes from GitHub to local repository**:

```
git pull origin master
```

Now, your machine learning project structure is set up with necessary files, dependencies, and version control using Git. You can continue building your project by implementing data ingestion, data transformation, model training, and prediction pipelines in the respective files and folders.