MLOps Readme - Understanding credit_defaults_model_ Folder

When you serve a model with MLflow, you need to point to a folder like:

```
/home/ubuntu/mlruns/1/<run_id>/artifacts/credit_defaults_model_
```

Inside this folder are key files that MLflow uses to correctly load and serve your model.

Contents of credit_defaults_model_ Folder

File	Purpose
MLmodel	Metadata that describes how to load and serve the model
model.pkl	The actual saved trained model (Pickle file for scikit-learn)
requirements.txt	List of Python packages needed to run the model
conda.yaml	Full environment specification (Python version, pip/conda dependencies)

Detailed File Explanation

1. MLmodel

- Metadata file.
- Tells MLflow what type of model (e.g., sklearn, xgboost, pytorch).
- Defines input/output signature (columns, data types).
- Points to the model artifact (e.g., model.pkl).
- Defines runtime environment (requirements or conda.yaml).

2. model.pkl

- The trained machine learning model itself.
- Saved using Pickle (via `joblib` or `pickle`).
- Contains the full model object (e.g., GradientBoostingClassifier).

3. requirements.txt

- Minimal list of Python packages needed for prediction.
- Used when deploying models with basic `pip install` workflows.

Example:

```
mlflow>=2.0.1
scikit-learn==1.0.2
numpy>=1.20.0
```

4. conda.yaml

- Full Conda environment definition.
- Used for complete environment isolation.
- MLflow can create this environment automatically before serving.

Example:

```
name: mlflow-env
channels:
  - defaults
dependencies:
  - python=3.11
  - pip
  - pip:
    - mlflow>=2.0.1
    - scikit-learn==1.0.2
    - numpy>=1.20.0
```

Serving Flow Overview

```
mlflow models serve -m /path/to/credit_defaults_model_ -p 5001
      ↓
Reads MLmodel file
      ↓
Sets up environment (using requirements.txt or conda.yaml)
      ↓
Loads model.pkl
      ↓
Starts REST API server (ready for predictions)
```

Important Notes

- **Always serve the folder**, not just the model file.
- **Make sure MLmodel is present** — it's the control file.
- **Check conda.yaml or requirements.txt** to ensure packages match.
- **Use "inputs" field in prediction JSON** when sending requests to the REST API.

Congratulations!

You now understand how MLflow models are structured and how the serving flow works. Happy MLOps! 🚀