Model Storage Information

📍 Model Storage Locations

1. **Production Models** (Used by API)

Location: `models/`

```
models/
├── best_model.pkl           # 506 KB - Random Forest model (88.52%
accuracy)
└── preprocessor.pkl         # 2.1 KB - Data preprocessing pipeline
```

Purpose: These are the models currently loaded and used by the FastAPI application for predictions.

Cloud Upload: ✅ These files should be uploaded to cloud storage (e.g., AWS S3, Azure Blob, GCS) for:

- Model versioning
- Disaster recovery
- Model serving in production
- Sharing across teams

---

2. **MLflow Tracked Models** (Experiment History)

Location: `mlruns/1/models/`

```
mlruns/1/models/
├── m-583ec23aaa2c4be38a0611eabe452f0e/   # Random Forest Model
│   └── artifacts/
│       └── model.pkl                      # 506 KB
└── m-8de5d87accce4a5197e29172038f1108/   # Logistic Regression Model
    └── artifacts/
        └── model.pkl                      # 996 B
```

Purpose: MLflow experiment tracking stores:

- All trained models with their parameters
- Metrics (accuracy, precision, recall, F1)
- Hyperparameters
- Training artifacts

Cloud Upload: ✅ The entire `mlruns/` directory can be synced to:

- MLflow Tracking Server (hosted)
- Cloud storage with MLflow backend

📦 What Should Be Uploaded to Cloud?

Option 1: Upload Production Models Only

Files to upload:

```
models/best_model.pkl
models/preprocessor.pkl
```

Use case: Quick deployment, minimal storage

Example commands:

```
# AWS S3
aws s3 cp models/best_model.pkl s3://your-bucket/heart-disease-models/v1/
aws s3 cp models/preprocessor.pkl s3://your-bucket/heart-disease-models/v1/

# Azure Blob Storage
az storage blob upload --account-name youraccount --container-name models \
  --name heart-disease/v1/best_model.pkl --file models/best_model.pkl

# Google Cloud Storage
gsutil cp models/best_model.pkl gs://your-bucket/heart-disease-models/v1/
```

Option 2: Upload MLflow Tracking Directory (Recommended)

Files to upload:

```
mlruns/
```

Use case: Full experiment history, reproducibility, model comparison

Setup MLflow with Cloud Backend:

```
# Configure MLflow to use cloud storage
import mlflow

# AWS S3
mlflow.set_tracking_uri("s3://your-bucket/mlruns")

# Azure Blob
mlflow.set_tracking_uri("wasbs://mlruns@youraccount.blob.core.windows.net/
")

# Google Cloud Storage
mlflow.set_tracking_uri("gs://your-bucket/mlruns")
```

🚀 Docker Image Considerations

Current Dockerfile Approach

The `Dockerfile` currently copies models into the container:

```
COPY models/ /app/models/
```

For production deployment:

1. **Option A: Download from Cloud at Runtime**

   ```
   # Install cloud CLI
   RUN pip install boto3  # for AWS

   # Download models at container startup
   CMD ["sh", "-c", "python download_models.py && python src/app.py"]
   ```

2. **Option B: Build-time Download**

   ```
   # Download during build
   RUN python -c "import boto3; s3.download_file('bucket', 'model.pkl',
   'models/')"
   ```

3. **Option C: Mount from Cloud Storage**

   - Use volume mounts in Kubernetes
   - Cloud-native solutions (AWS EFS, Azure Files)

📊 Model Versioning Strategy

Recommended Structure in Cloud:

```
s3://your-mlops-bucket/
├── heart-disease-models/
│   ├── v1.0.0/
│   │   ├── best_model.pkl
│   │   ├── preprocessor.pkl
```

```
│   │   └── metadata.json
│   ├── v1.1.0/
│   │   ├── best_model.pkl
│   │   ├── preprocessor.pkl
│   │   └── metadata.json
│   └── latest/  (symlink to current version)
├── mlruns/  (MLflow tracking data)
└── datasets/
    └── heart-disease/
        ├── raw/
        └── processed/
```

🔧 Quick Upload Script

Create `upload_models.sh`:

```bash
#!/bin/bash
# Upload models to cloud storage

VERSION="v1.0.0"
BUCKET="your-mlops-bucket"

# Create versioned directory
aws s3 cp models/best_model.pkl s3://$BUCKET/heart-disease-models/$VERSION/
aws s3 cp models/preprocessor.pkl s3://$BUCKET/heart-disease-models/$VERSION/

# Update 'latest' pointer
aws s3 cp models/best_model.pkl s3://$BUCKET/heart-disease-models/latest/
aws s3 cp models/preprocessor.pkl s3://$BUCKET/heart-disease-models/latest/

echo "✅ Models uploaded to s3://$BUCKET/heart-disease-models/$VERSION/"
```

📝 Metadata to Track

Create `models/metadata.json`:

```json
{
  "model_name": "heart_disease_random_forest",
  "version": "1.0.0",
  "trained_date": "2025-12-30",
  "accuracy": 0.8852,
  "model_file": "best_model.pkl",
  "preprocessor_file": "preprocessor.pkl",
  "framework": "scikit-learn",
  "python_version": "3.13.5",
  "dependencies": "requirements.txt",
  "training_data": "UCI Heart Disease Dataset"
}
```

✅ Current Status

- ✅ Models trained and saved locally
- ✅ MLflow tracking configured
- ✅ Best model selected (Random Forest: 88.52%)
- ✅ Preprocessor saved
- ✅ API serving models successfully
- ⏳ Cloud storage integration pending
- ⏳ Model registry setup pending
- ⏳ CI/CD for model deployment pending

🎯 Next Steps for Cloud Integration

1. **Choose cloud provider:** AWS, Azure, or GCP
2. **Set up storage bucket:** Create dedicated bucket for ML models
3. **Configure access:** Set up IAM roles and credentials
4. **Implement download script:** Create `src/download_models.py`
5. **Update Dockerfile:** Modify to download models from cloud
6. **Set up MLflow remote tracking:** Point to cloud-hosted MLflow server
7. **Implement model registry:** Use MLflow Model Registry for versioning
8. **Add CI/CD:** Automate model upload on successful training

---

📚 Additional Resources

- MLflow Model Registry: https://mlflow.org/docs/latest/model-registry.html
- AWS S3 for ML: https://aws.amazon.com/s3/machine-learning/
- Azure ML Model Management: https://docs.microsoft.com/azure/machine-learning/
- GCP Vertex AI: https://cloud.google.com/vertex-ai