# FastAPI Server Sast Report

## Introduction

This report is a brief description of what SonarQube scan found on FastAPI ML model server.

## Security hotpots (review required)

The only hotspot found was the way the model was loaded:

```python
params = torch.load(path, map_location=device, weights_only=False)
```

The problem with this function is that it uses pickle library to load the model file. If the file had malicious payload injected it could be executed during deserialization.

It can be fixed by loading the model using a safe alternative, which is safetensors library - it does not use pickle deserialization:

```python
import torch
import safetensors

model = FinetunedModel()
safetensors.torch.load_model(model, path)
```

The file with models is created and changed by the developer who has access to the VM on the server. It is not downloaded from external sources, nor can itbe changed by a user. The API itself is accessible only by the Django (Gunicorn) server. The adversary would have to have access to the user on VM, that governs the app. Adversary would have a lot more options to disrupt the system and replacing the model would be the smallest issue, if it came to that. Considering this, the conclusion is that it is not a low priority issue.

## Issues (minor)

All the issues regarded code maintainability. While it is important to have a clean code, because it reduces risk of potential bugs in the future, it is not a direct security issue.

Issues:

- 2 variables used camelCase
- 2 variables were unused
- one function had too much complexity

## Conclusion

No high-risk security issues were found. One deserialization hotspot requires awareness but is low-impact given the environment. Maintainability improvements are recommended.