

Documentação do Projecto de Aprendizado de Máquina

Implantação

1. Visão geral

Após o treinamento do modelo decidimos salvar ele usando a biblioteca joblib, essa biblioteca foi projectada especificamente para ser mais eficiente na serialização de objectos que contêm grandes arrays de dados Numpy. Como a maioria dos modelos de Machine Learning , especificamente os criados com o scikit-learn, depende muito desses arrays(para armazenar pesos, coeficientes, etc.), o joblib se torna a ferramenta preferencial de serialização nesse ecossistema.

2. Serialização de modelo

Foi feito um dump(salvar) do modelo com a extensão .joblib garantindo uma maior velocidade, assim temos um salvamento e carregamento mais rápidos para modelos grandes, com essa extensão ela faz o mapeamento de memória(memory mapping) permitindo que você carregue apenas as partes necessárias do modelo, economizando memória.

3. Modelo de Serviço

Optamos por usar o serviço em nuvem streamlit community, pós ele oferece uma hospedagem gratuita, nos permite criar interface com a própria biblioteca e tornar as predições do modelo mais interativa. Com o modelo carregado no ambiente de produção do streamlit community o modelo ainda guarda consigo os seus parâmetros de predição, então após a inserção das features treinadas pelo modelo através da interface da aplicação o modelo recebe esses dados e é chamado a função .predict() para fazer as novas previsões.

4. Integração de API

O modelo pode ser disponibilizado em um serviço de API flask ou fastapi, o que não foi o nosso caso. Em projectos futuros podemos debater mais a cerca deste tema.

5. Considerações de Segurança

Na fase da implantação não foi aplicada quais quer medida de segurança.

6. Monitoramento e registro

O monitoramento do modelo é uma etapa crucial para projectos de aprendizagem de máquina, mas para o nosso caso não aplicamos nenhum parâmetro de monitoramento de registros do desempenho do modelo em produção, em uma outra oportunidade será crucial explorar algumas técnicas de monitoramento como o grafana e o prometheus.