TABLE OF CONTENTS

Model Deployment and API

Serialize the trained claim prediction model.

Practical Applications

A Serialised model can be shared over the internet or as files and can be used much faster and more accessibly without the use of computation power everytime to train it again

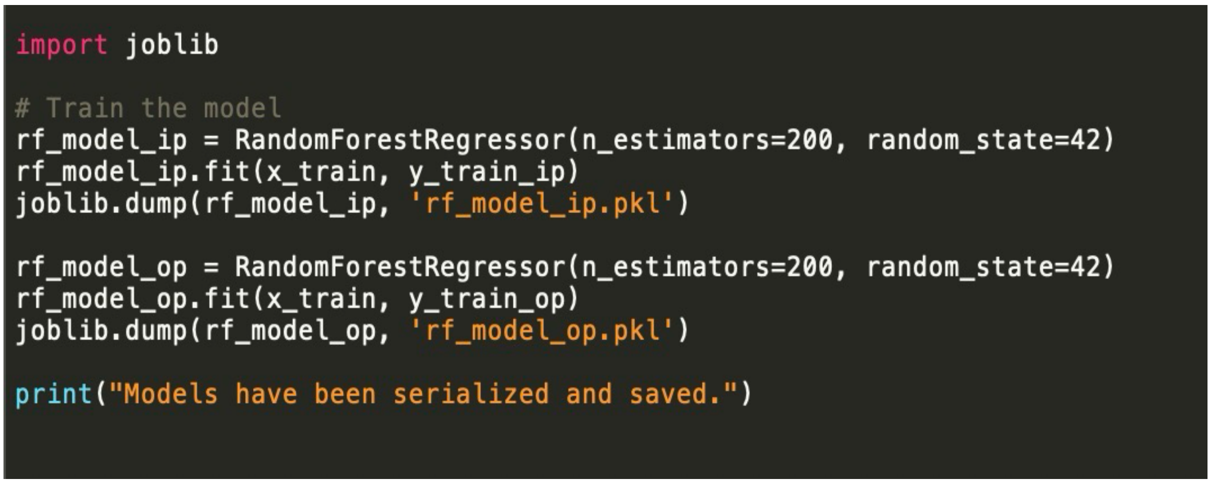
Understanding

Serializing a model refers to the process of converting a machine learning model (or any data structure) into a format that can be easily saved to disk, transferred over a network, or stored in a database, and then later deserialized (converted back) to its original state. This is crucial for saving the model after training so that it can be used for predictions without retraining.

Some Common Libraries used to serialise a model are

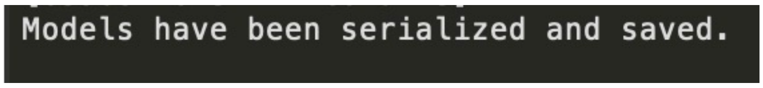
- Pickle
- Joblib
- TensorFlow
- ONNX

Script



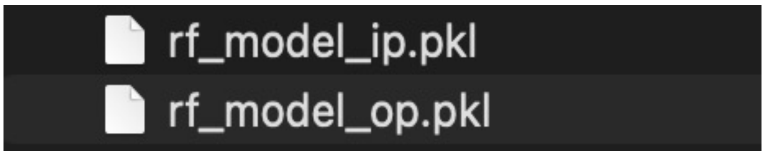
```
import joblib

# Train the model
rf_model_ip = RandomForestRegressor(n_estimators=200, random_state=42)
rf_model_ip.fit(x_train, y_train_ip)
joblib.dump(rf_model_ip, 'rf_model_ip.pkl')

rf_model_op = RandomForestRegressor(n_estimators=200, random_state=42)
rf_model_op.fit(x_train, y_train_op)
joblib.dump(rf_model_op, 'rf_model_op.pkl')

print("Models have been serialized and saved.")
```

Output



```
Models have been serialized and saved.
```

The model Files



```
rf_model_ip.pkl
rf_model_op.pkl
```

Deploy the model as a service with an API for integration.

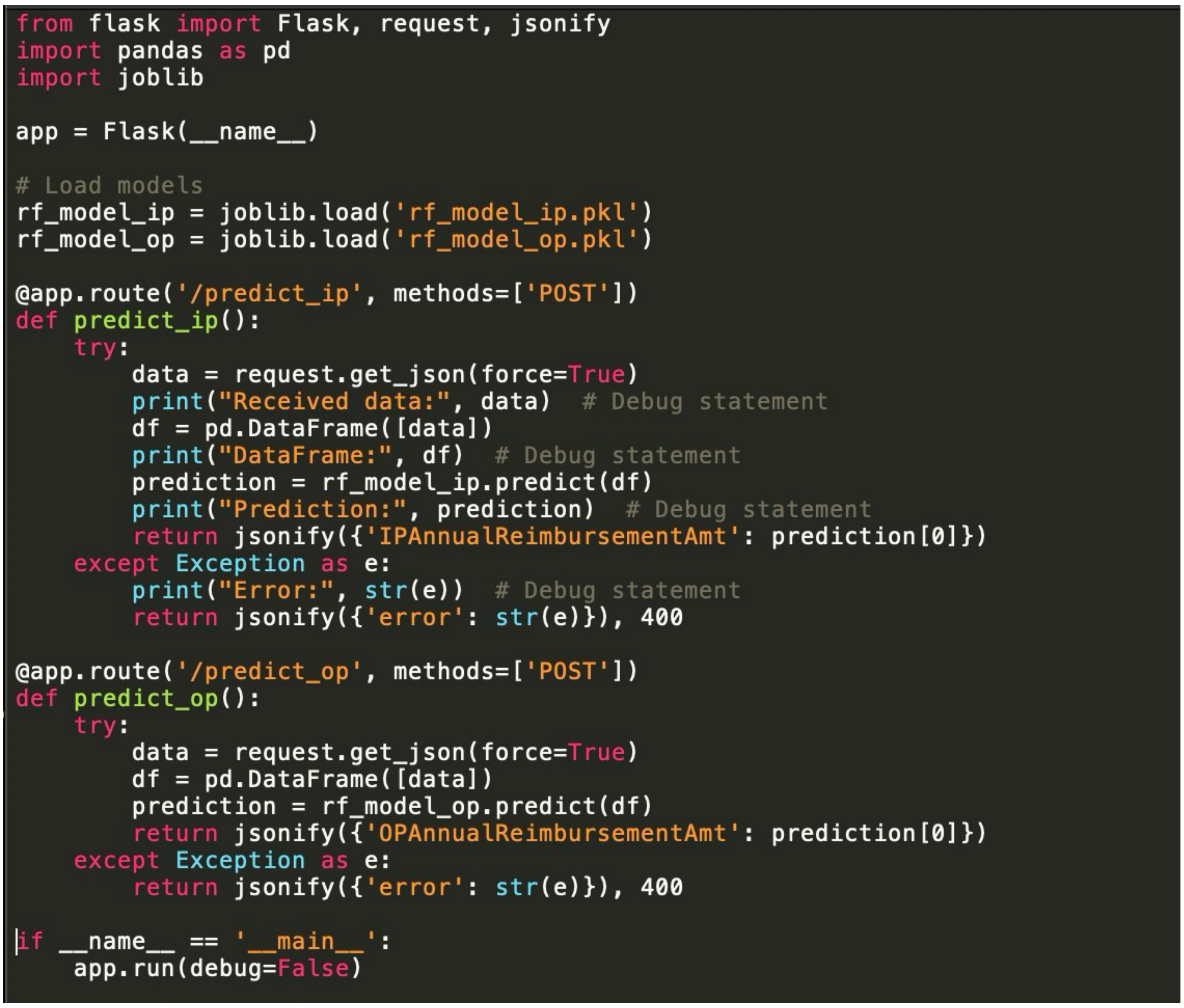
Practical Applications

Financial Websites may deploy stock prediction models as service on their websites which can be used by API for easier and more globally accessible use of their models for the customers

Understanding

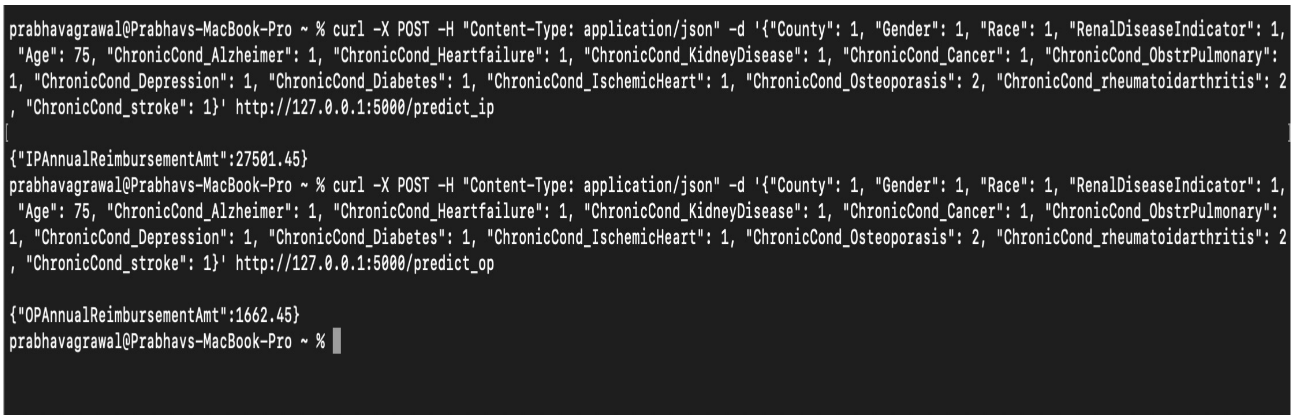
We can deploy a model as a service using various ways one of them can be to host them on a port in local host which can be then accessed via HTTP HTTPS requests

Script



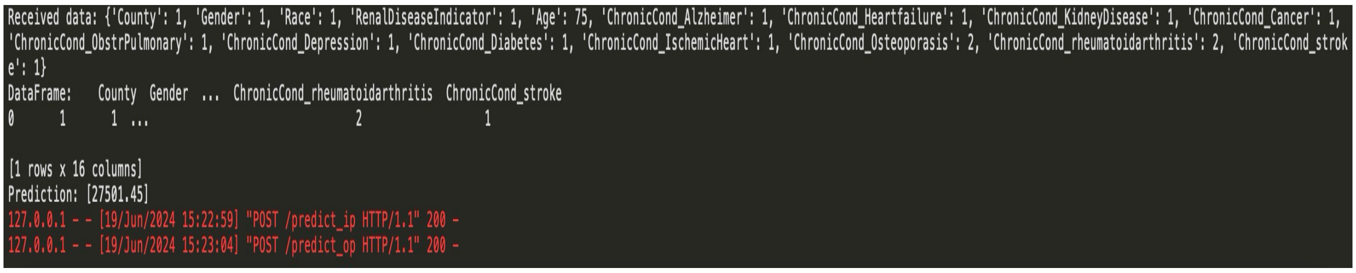
```python
from flask import Flask, request, jsonify
import pandas as pd
import joblib

app = Flask(__name__)

# Load models
rf_model_ip = joblib.load('rf_model_ip.pkl')
rf_model_op = joblib.load('rf_model_op.pkl')

@app.route('/predict_ip', methods=['POST'])
def predict_ip():
    try:
        data = request.get_json(force=True)
        print("Received data:", data)  # Debug statement
        df = pd.DataFrame([data])
        print("DataFrame:", df)  # Debug statement
        prediction = rf_model_ip.predict(df)
        print("Prediction:", prediction)  # Debug statement
        return jsonify({'IPAnnualReimbursementAmt': prediction[0]})
    except Exception as e:
        print("Error:", str(e))  # Debug statement
        return jsonify({'error': str(e)}), 400

@app.route('/predict_op', methods=['POST'])
def predict_op():
    try:
        data = request.get_json(force=True)
        df = pd.DataFrame([data])
        prediction = rf_model_op.predict(df)
        return jsonify({'OPAnnualReimbursementAmt': prediction[0]})
    except Exception as e:
        return jsonify({'error': str(e)}), 400

if __name__ == '__main__':
    app.run(debug=False)
```

<u>API Query And OUTPUT</u>



```
prabhavagrawal@Prabhavs-MacBook-Pro ~ % curl -X POST -H "Content-Type: application/json" -d '{"County": 1, "Gender": 1, "Race": 1, "RenalDiseaseIndicator": 1,
 "Age": 75, "ChronicCond_Alzheimer": 1, "ChronicCond_Heartfailure": 1, "ChronicCond_KidneyDisease": 1, "ChronicCond_Cancer": 1, "ChronicCond_ObstrPulmonary":
1, "ChronicCond_Depression": 1, "ChronicCond_Diabetes": 1, "ChronicCond_IschemicHeart": 1, "ChronicCond_Osteoporasis": 2, "ChronicCond_rheumatoidarthritis": 2
, "ChronicCond_stroke": 1}' http://127.0.0.1:5000/predict_ip

{"IPAnnualReimbursementAmt":27501.45}
prabhavagrawal@Prabhavs-MacBook-Pro ~ % curl -X POST -H "Content-Type: application/json" -d '{"County": 1, "Gender": 1, "Race": 1, "RenalDiseaseIndicator": 1,
 "Age": 75, "ChronicCond_Alzheimer": 1, "ChronicCond_Heartfailure": 1, "ChronicCond_KidneyDisease": 1, "ChronicCond_Cancer": 1, "ChronicCond_ObstrPulmonary":
1, "ChronicCond_Depression": 1, "ChronicCond_Diabetes": 1, "ChronicCond_IschemicHeart": 1, "ChronicCond_Osteoporasis": 2, "ChronicCond_rheumatoidarthritis": 2
, "ChronicCond_stroke": 1}' http://127.0.0.1:5000/predict_op

{"OPAnnualReimbursementAmt":1662.45}
prabhavagrawal@Prabhavs-MacBook-Pro ~ %
```

On 127.0.0.1:5000/predict_ip – The inpatient prediction model has been deployed
On 127.0.0.1:5000/predict_op – The outpatient prediction model has been deployed



```
Received data: {'County': 1, 'Gender': 1, 'Race': 1, 'RenalDiseaseIndicator': 1, 'Age': 75, 'ChronicCond_Alzheimer': 1, 'ChronicCond_Heartfailure': 1, 'ChronicCond_KidneyDisease': 1, 'ChronicCond_Cancer': 1,
'ChronicCond_ObstrPulmonary': 1, 'ChronicCond_Depression': 1, 'ChronicCond_Diabetes': 1, 'ChronicCond_IschemicHeart': 1, 'ChronicCond_Osteoporasis': 2, 'ChronicCond_rheumatoidarthritis': 2, 'ChronicCond_strok
e': 1}
DataFrame:    County  Gender  ...  ChronicCond_rheumatoidarthritis  ChronicCond_stroke
0      1      1  ...                                2                   1

[1 rows x 16 columns]
Prediction: [27501.45]
127.0.0.1 - - [19/Jun/2024 15:22:59] "POST /predict_ip HTTP/1.1" 200 -
127.0.0.1 - - [19/Jun/2024 15:23:04] "POST /predict_op HTTP/1.1" 200 -
```