

# Incident management process enriched event log



ÉCOLE  
**D'INGÉNIEURS**  
PARIS-LA DÉFENSE

Shuyu CHEN

# 1 Introduction

This event log was extracted from data gathered from the audit system of an instance of the ServiceNow platform used by an IT company and enriched with data loaded from a relational database.

It contains the entire process of incident management, including key points in time, Id of responsible persons, etc.

The dataset contains 141712 events of 24918 incidents.

## 2 Type of variables

### **36 variables in total:**

5 time variables

15 different categories of ids

4 Boolean variables

3 int variables

9 categorical variables

# 3 Target

- Change all the time variable to timestamp
- Target variable is generated by subtracting “sys\_updated\_at” variable from “closed\_at” variable

# 4 Label encoder

- Regression would be used, so the data type should be numeric, so I used label encoder to change category and bool type data to numeric.

```
from sklearn.preprocessing import LabelEncoder
```

```
def convert(data, column):  
    number = LabelEncoder()  
    data[column] = number.fit_transform(data[column])  
    data = data.fillna(-999)  
    return data
```

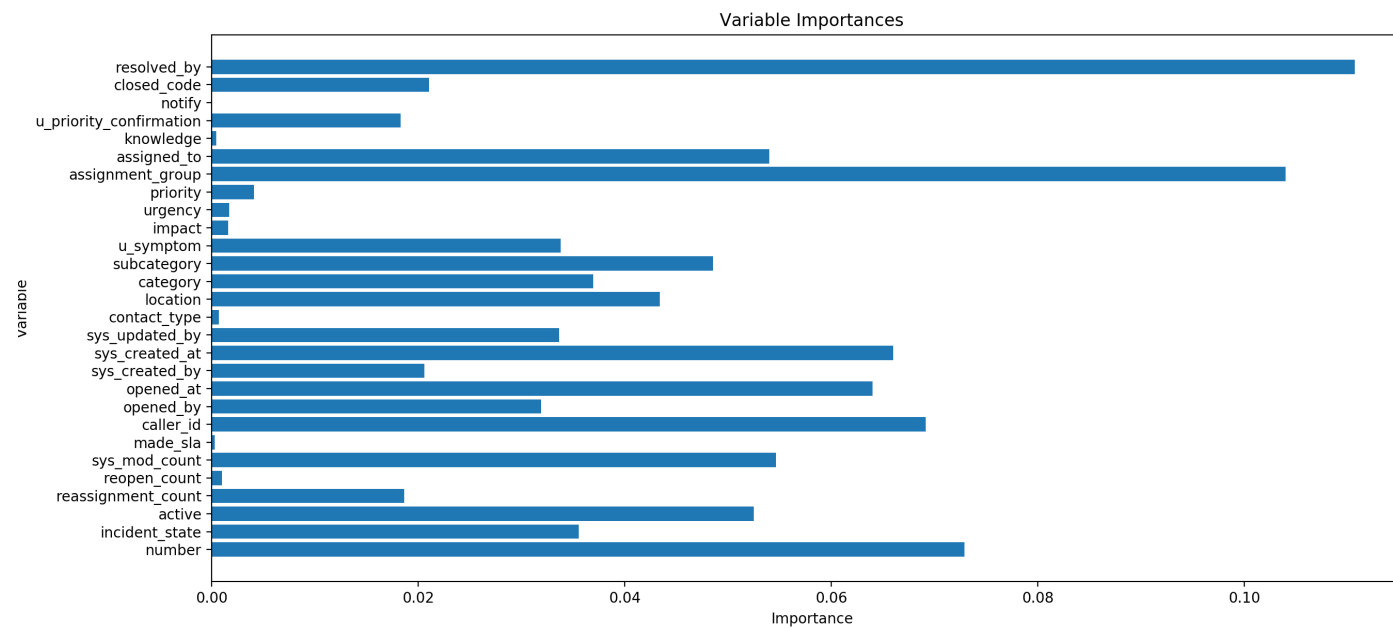
# 5 Methods

- decision tree regression (score $\approx$ 0.92)
- random forest (score $\approx$ 0.96)
- grid search to find the best hyperparameters.

# 6 feature engineering

- First, I used the dataset with “resolved\_at” column. The score is about 0.96 but the feature importance showed that “resolved\_at” variable is the most important.
- So, I deleted “resolved\_at” column, and build a new random forest model. But the score was about 0.78.
- Then I deleted all columns whose importance equal 0, and built a new model, then the score increased to 0.96.

# 7 feature importance



Final chosen variables:

number  
active  
sys\_mod\_count  
opened\_by  
sys\_created\_by  
sys\_updated\_by  
category  
u\_symptom  
assigned\_to  
closed\_code  
resolved\_at



# 8 result

Result without  
feature engineering

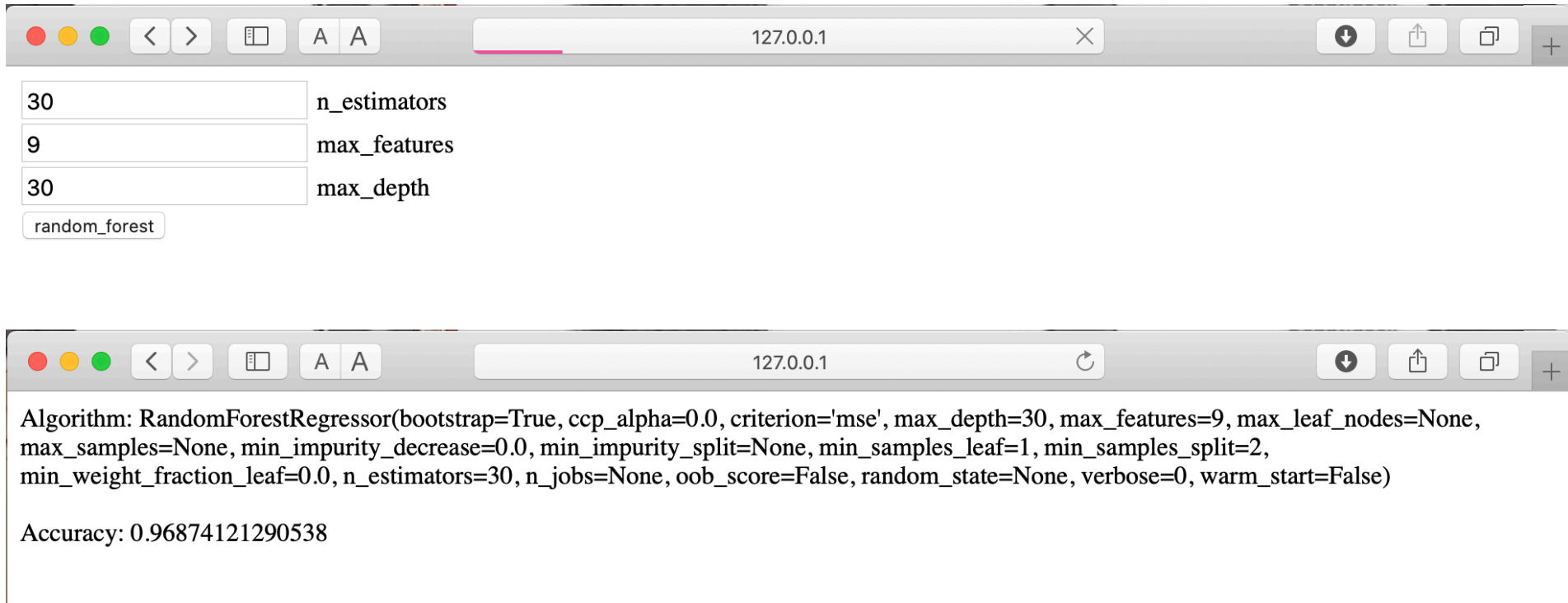
```
(0.7711257408991313,  
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=50,  
                        max_features=28, max_leaf_nodes=None,  
                        min_impurity_decrease=0.0, min_impurity_split=None,  
                        min_samples_leaf=1, min_samples_split=2,  
                        min_weight_fraction_leaf=0.0, n_estimators=50,  
                        n_jobs=None, oob_score=False, random_state=None,  
                        verbose=0, warm_start=False))
```

Result with feature  
engineering

```
(0.9690632635369949,  
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=50,  
                        max_features=20, max_leaf_nodes=None,  
                        min_impurity_decrease=0.0, min_impurity_split=None,  
                        min_samples_leaf=1, min_samples_split=2,  
                        min_weight_fraction_leaf=0.0, n_estimators=30,  
                        n_jobs=None, oob_score=False, random_state=None,  
                        verbose=0, warm_start=False))
```

# 9 Flask API

Choose hyperparameters to build model



The screenshot shows a web browser window with the address bar displaying '127.0.0.1'. The browser has standard navigation buttons (back, forward, home, search) and window controls. The main content area contains a form for configuring a Random Forest model. The form has three input fields for hyperparameters: 'n\_estimators' with the value '30', 'max\_features' with the value '9', and 'max\_depth' with the value '30'. Below these fields is a button labeled 'random\_forest'. The browser window is titled '127.0.0.1' and has a refresh button. The form is styled with a light gray background and rounded corners.

30	n_estimators
9	max_features
30	max_depth

random\_forest

Algorithm: RandomForestRegressor(bootstrap=True, ccp\_alpha=0.0, criterion='mse', max\_depth=30, max\_features=9, max\_leaf\_nodes=None, max\_samples=None, min\_impurity\_decrease=0.0, min\_impurity\_split=None, min\_samples\_leaf=1, min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0, n\_estimators=30, n\_jobs=None, oob\_score=False, random\_state=None, verbose=0, warm\_start=False)

Accuracy: 0.96874121290538



Thank you for your evaluating