ML Applications on IT Incidents Management

Claudio Obregón Noriega

LinkedIn: linkedin.com/in/obregoncr
Correo: cobregon@stevens.edu

1

ML Applications on IT Incidents Management

Abstract

Effective IT incident management is crucial for minimizing downtime and ensuring the reliability of IT services in today's fast-paced technological environment. This project explores the application of machine learning techniques to enhance incident management processes. Using convolutional neural networks (CNNs), incidents are categorized with improved accuracy, enabling faster and more efficient resolution. Support Vector Machines (SVMs) are employed to predict incident priority, ensuring critical issues are addressed promptly. Additionally, time series analysis is utilized to forecast incident trends, allowing organizations to allocate resources proactively and mitigate potential disruptions. The results demonstrate that integrating machine learning into IT incident management can streamline operations, optimize response times, and improve service quality. This project underscores the transformative potential of AI-driven solutions in IT service management, paving the way for more resilient and efficient IT operations.

2

Agenda

01 Project Description

02 ML Application on IT Incidents Management

03 Literature Review: References

04 Findings & Conclussions

3

ML and IA Applications on ITSM

Incidents Forecast

Machine Learning models can analyze historical data trends to predict the occurrence of incidents son that IT teams could be prepare

Resources Management

Machine Learning models can easily identify improvements areas and therefore optimize resource assignment for incidents handling.

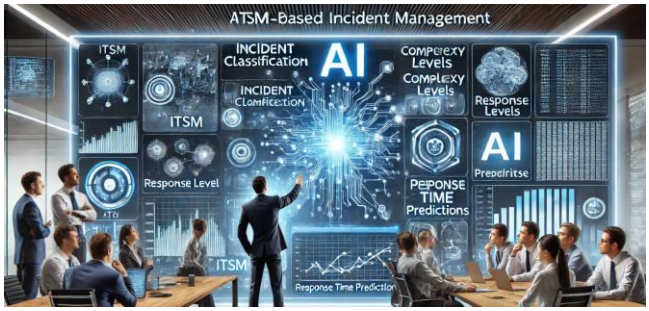
Tasks Automatization

Machine Learning can automatize repetitive tasks such as incident resolution allowing IT team to focus in other activities which adds more value for the business.

4

Project Objectives

- To explain the methodology and benefits of using convolutional neural networks (CNNs) for accurate incident categorization in IT incident management.
- To showcase how Support Vector Machines (SVM) can be utilized to predict and prioritize IT incidents based on severity and urgency.
- To explore the application of time series analysis for forecasting future incidents, enabling proactive decision-making and resource allocation.



5

5

Scope of Work

- **Literature Review**: Conduct a comprehensive review of existing research on Machine Learning applications in IT Incidents management. This will involve collecting and analyzing previous studies and outcomes to understand current methodologies, success rates, and limitations.
- **Data Acquisition:** Source and featured dataset of IT Incidents that are relevant for this project. These datasets include both labeled data to train and test the model and new data to make the predictions. Data has been obtained from Kaggle research platform and from my own elaboration dataset.
- **Coding and Training the Model**: Develop a machine learning algorithm-based model for:
  - **Incident Prediction:** Predict future incidents based on historical data patterns to help prepare resources and reduce unexpected system impacts.
  - **Incident Classification:** Use ML models to automatically classify incidents into categories or types, assisting teams in routing issues to appropriate departments or teams more swiftly.
  - **Incident Prioritization:** Predict incident severity or urgency to determine which incidents need immediate attention, ensuring that critical issues are addressed promptly to minimize system disruption.
- **Summarizing Findings:** Analyze and document the model's performance based on the testing and validation results. Summarize key findings, including the model's accuracy and performance. Discuss any challenges encountered during the project and propose recommendations for further improvements or additional research. Prepare a comprehensive report and presentation that encapsulates all phases of the project from literature review to practical outcomes.

6

6

Case1: Incidents Classification with ML

7

Case1: Incidents Classification with ML

Description: Incidents Classification with ML algorithms using Python.

Input: ITSM Incidents database

Output: Incidents classifications based on key features such as descripcion “Titulo”, Tecnician “asignado a”, user “usuario”, and priority: “prioridad”

Tools and Procedures:

- Python
- Neural Networks Algorithms - CNN

Challenges

- Classification issues for IT Teams leading to extended resolution time for incidents
- Resource planning for each IT Teams
- Predict demand for IT Services

8

Procedure:

Preprocessing:

- Convert text fields into numerical features using NLP techniques such as TF-IDF if the description/title is used.
- Encode categorical variables like priority, type, and status using one-hot encoding or label encoding.
- Split the data into training and testing sets.

Modeling:

- Use a Neural Network model

Evaluation:

- Evaluate the model using precision, accuracy, recall, and F1-score.
- Use a Confusion Matrix to understand how well the model performs across different classes.

Features Codification:

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Assuming df_incidente_cleaned is already defined, create an explicit copy to avoid warnings.
df_incidente_cleaned = df_incidente_cleaned.copy()

# Define LabelEncoder instances
le_titulo = LabelEncoder()
le_reporter = LabelEncoder()
le_priority = LabelEncoder()
le_technician = LabelEncoder()

# Apply encoding to the fields and store them in new columns
df_incidente_cleaned['Categoría_Titulo_Enc'] = le_titulo.fit_transform(df_incidente_cleaned['Titulo'])
df_incidente_cleaned['Solicitante_Enc'] = le_reporter.fit_transform(df_incidente_cleaned['Solicitante'])
df_incidente_cleaned['Prioridad_Enc'] = le_priority.fit_transform(df_incidente_cleaned['Prioridad'])
df_incidente_cleaned['Tecnico_Enc'] = le_technician.fit_transform(df_incidente_cleaned['Asignado a Técnico'])

# Verify that the columns were created successfully
print(df_incidente_cleaned[['Categoría_Titulo_Enc', 'Solicitante_Enc', 'Prioridad_Enc', 'Tecnico_Enc']].head())
```

	Categoría_Titulo_Enc	Solicitante_Enc	Prioridad_Enc	Tecnico_Enc
1	7268	878	2	64
2	1496	954	2	64
3	3063	105	2	64
4	4894	938	2	64
5	22757	880	2	28

```
print(df_incidente.shape)
print(df_incidente_cleaned.shape)
```

```
(59949, 51)
(59929, 55)
```

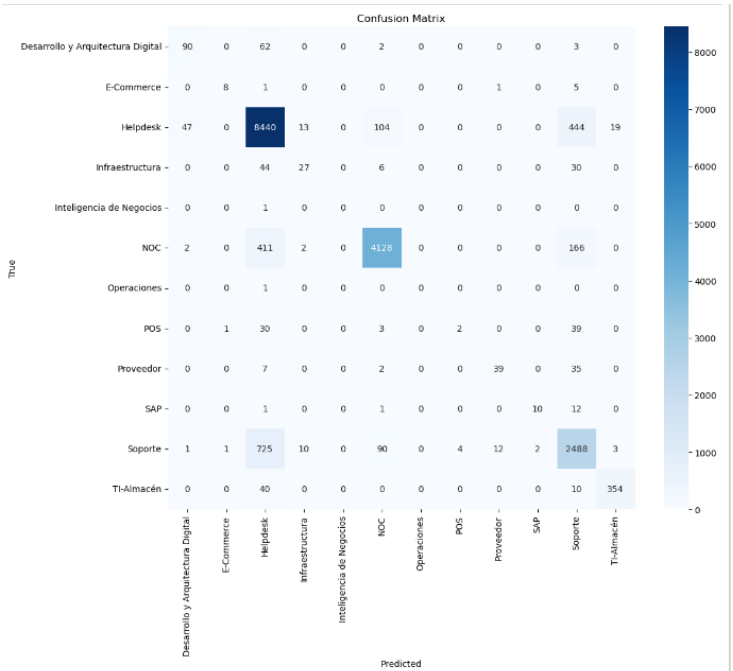
ML – NN Algorithm:

```python
from sklearn.model_selection import train_test_split

# Split the data into 70% training and 30% testing
train_data, test_data = train_test_split(
    df_incidente_cleaned[features + [target]],
    test_size=0.3,
    random_state=42
)

# Separate features and target for training and testing
X_train = train_data[features]
y_train = train_data[target]
X_test = test_data[features]
y_test = test_data[target]

# Verify the shapes of the training and testing data
print(f"Training data shape: {X_train.shape}, Training target shape: {y_train.shape}")
print(f"Test data shape: {X_test.shape}, Test target shape: {y_test.shape}")
```

```
Training data shape: (41950, 4), Training target shape: (41950,)
Test data shape: (17979, 4), Test target shape: (17979,)
```

	precision	recall	f1-score	support
Desarrollo y Arquitectura Digital	0.64	0.57	0.61	157
E-Commerce	0.80	0.53	0.64	15
Helpdesk	0.86	0.93	0.90	9067
Infraestructura	0.52	0.25	0.34	107
Inteligencia de Negocios	1.00	0.00	0.00	1
NOC	0.95	0.88	0.91	4709
Operaciones	1.00	0.00	0.00	1
POS	0.33	0.03	0.05	75
Proveedor	0.75	0.47	0.58	83
SAP	0.83	0.42	0.56	24
Soporte	0.77	0.75	0.76	3336
TI-Almacén	0.94	0.88	0.91	404
accuracy			0.87	17979
macro avg	0.78	0.48	0.52	17979
weighted avg	0.86	0.87	0.86	17979

Results:



Case2: Incidents Prioritization with ML

13

Case2: Incidents Prioritization with ML

Description: Incidents Prioritization with ML algorithms using Python.

Input: ITSM Incidents database

Output: Incidents classifications based on key features such as descripcion “Titulo”, Tecnician “asignado a”, user “usuario”, and priority: “prioridad”

Tools and Procedures:

- Python
- Support Vector Machines SVM

Challenges

- Classification issues for IT Teams leading to extended resolution time for incidents
- Resource planning for each IT Teams
- Predict demand for IT Services

14

Procedure:

Preprocessing:

- Select independent variables (features) and the dependent variable (label).
- Encode the values of these variables using equivalent numerical values.
- Split the data into training and testing sets.

Modeling:

- Use of SVM model

Evaluation:

- Evaluate the model applied to the test data using metrics such as precision, accuracy, recall, and F1-score.
- Use a Confusion Matrix to understand how well the model is performing across different classes.

Data Pre-Processing:

	CI_Cat	CI_Subcat	WBS	Category
0	subapplication	Web Based Application	WBS000162	incident
1	application	Web Based Application	WBS000088	incident
2	application	Desktop Application	WBS000092	request for information
3	application	Web Based Application	WBS000088	incident
4	application	Web Based Application	WBS000088	incident
5	application	Web Based Application	WBS000088	incident
6	application	Web Based Application	WBS000055	incident
7	application	Web Based Application	WBS000088	incident
8	application	Web Based Application	WBS000088	incident
9	application	Web Based Application	WBS000055	incident

```
# Label Encoding
enc= LabelEncoder()
for i in (0,1,2,3):
    X.iloc[:,i] = enc.fit_transform(X.iloc[:,i])
```

```
print("DataFrame X")
print(X.head(10))
print("Data y")
print(y.head(10))
```

```
DataFrame X
  CI_Cat CI_Subcat  WBS Category
0      5        16  137        1
1      0        16   70        1
2      0         4   74        3
3      0        16   70        1
4      0        16   70        1
5      0        16   70        1
6      0        16   43        1
7      0        16   70        1
8      0        16   70        1
9      0        16   43        1
```

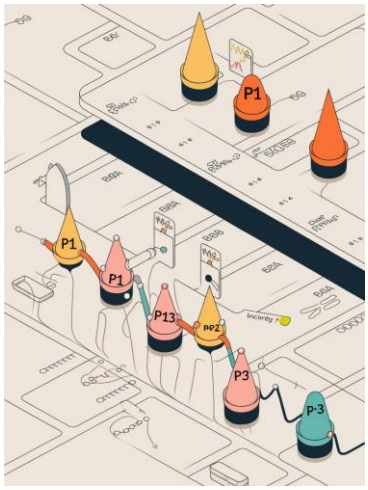
SVM Application:

1. Support Vector Machine

```
# Training the model
from sklearn.svm import SVC
rbf_svc = SVC(kernel='rbf',C=10,gamma=0.1).fit(X_train,y_train)
```

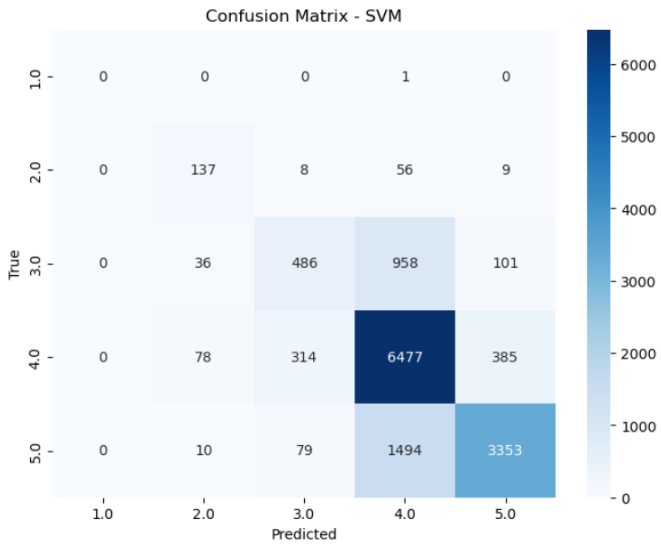
```
# Predicting the model
y_predict_svm = rbf_svc.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Print the classification report
print("Classification Report:")
print(classification_report(y_test, y_predict_svm, zero_division=1))

# Define unique labels based on the unique values in y_test
unique_labels = np.unique(y_test)

# Compute and plot the confusion matrix
cm = confusion_matrix(y_test, y_predict_svm)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=unique_labels, yticklabels=unique_labels)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix - SVM')
plt.show()
```



Results:

```
Classification Report:
              precision    recall  f1-score   support

         1.0       1.00      0.00      0.00         1
         2.0       0.52      0.65      0.58       210
         3.0       0.55      0.31      0.39      1581
         4.0       0.72      0.89      0.80      7254
         5.0       0.87      0.68      0.76      4936

    accuracy                           0.75     13982
   macro avg       0.73      0.51      0.51     13982
weighted avg       0.75      0.75      0.74     13982
```



Case3: Incidents Forecast with ML

19

Case3: Incidents Forecast with ML

Description: Incidents Forecast with ML algorithms using Python.

Input: ITSM Incidents database

Output: Incidents forecast based on key features

Tools and Procedures:

- Python
- Time series SARIMA

Challenges

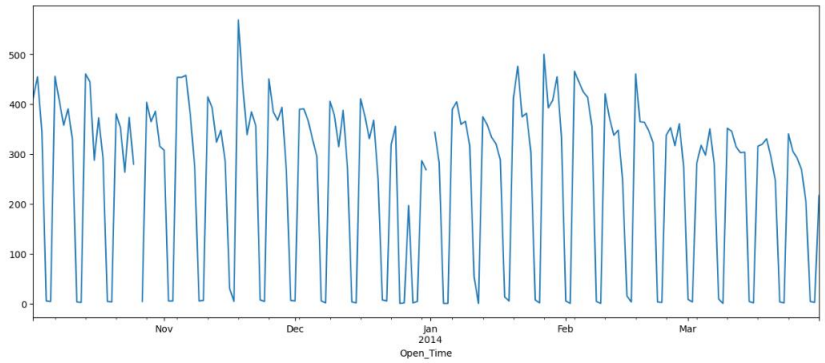
- Classification issues for IT Teams leading to extended resolution time for incidents
- Resource planning for each IT Teams
- Predict demand for IT Services

20

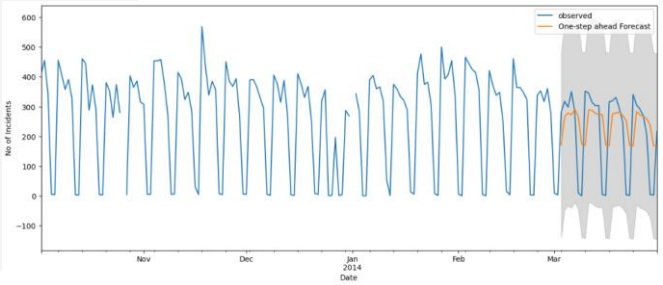
ITSM Incidents Database

Open_Time	No_Incidents
2013-10-02	412
2013-10-03	455
2013-10-04	345
2013-10-07	456
2013-10-05	6

```
# Plotting number of tickets per day after October 2013
data2.plot(figsize=(15,6))
plt.show()
```



Forecast using Time Series SARIMA Model:




```
# Making a list of values for p,d & q
p = d = q = range(0,2)
pdq = list(itertools.product(p,d,q))
```

```
# Checking the AIC values per pairs
for param in pdq:
    mod = sm.tsa.statespace.SARIMAX(data2,order=param,enforce_stationarity=False,enforce_invertibility=False)
    results = mod.fit()
    print('ARIMA{} - AIC:{}'.format(param, results.aic))
```

```
ARIMA(0, 0, 0) - AIC:2539.6180293605685
ARIMA(0, 0, 1) - AIC:2373.7853824722524
ARIMA(0, 1, 0) - AIC:2371.128960804689
ARIMA(0, 1, 1) - AIC:2313.1363347365786
ARIMA(1, 0, 0) - AIC:2365.291646936565
ARIMA(1, 0, 1) - AIC:2337.312508693354
ARIMA(1, 1, 0) - AIC:2373.128068065154
ARIMA(1, 1, 1) - AIC:2294.4315812436525
```

```
# Choosing the model with minimum AIC and the ARIMA Model for Time Series Forecasting
mod = sm.tsa.statespace.SARIMAX(data2,order=(1,1,1))
results = mod.fit()
print(results.summary().tables[1])
```

```
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.3386      0.090      3.771      0.000       0.163       0.515
ma.L1         -0.9989      0.428     -2.332      0.020      -1.839      -0.159
sigma2      2.52e+04   9781.775      2.576      0.010    6023.428    4.44e+04
==============================================================================
```

```
# Predicting the future values and the confidence interval
pred = results.get_prediction(start=pd.to_datetime('2014-3-3'),end=pd.to_datetime('2014-3-31'),dynamic=False)
pred_ci = pred.conf_int()
pred.predicted_mean.round()
```

References

- S. Silva, R. Pereira and R. Ribeiro, "Machine learning in incident categorization automation," 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), Caceres, Spain, 2018, pp. 1-6
- Ain, A. A. Z., & Safitri, C. (2023). Enhancing ITIL Incident Management: Innovative Machine Learning Approaches for Efficient Incident Prioritization and Resolution. Jurnal Teknik Informatika, 16(2)Gestión de Proyectos
- Boonprapapan, T., Seresangtakul, P., & Horata, P. (2024). Service priority classification using machine learning. Science, Engineering and Health Studies, 18, Article 24020002
- ITSM Incident Management (ABC Tech): https://www.kaggle.com/datasets/ahanwadi/itsm-data

Conclussions

The evaluation of Machine Learning (ML) applications in ITSM focused on three critical cases: **Incidents Forecasting**, **Incidents Classification**, and **Incidents Prioritization**. The findings demonstrate significant potential for ML to enhance IT service management processes:

1.Incidents Forecasting: Predictive models showed high accuracy in forecasting incident occurrence patterns, enabling proactive resource allocation and system maintenance. This reduces downtime and mitigates potential risks.
2.Incidents Classification: Supervised learning algorithms effectively categorized incidents, streamlining the triaging process. This ensures incidents are routed to the appropriate teams faster, reducing response times and improving operational efficiency.
3.Incidents Prioritization: ML models successfully determined incident urgency and impact, aiding in prioritization. This ensures critical issues are resolved first, enhancing service reliability and customer satisfaction.