PySpark ML models for audio samples classification

August 2025

Abstract

This work explores urgency detection in spoken Italian by evaluating the performance of three supervised classifiers implemented using the pyspark.ml library: Random Forest, Logistic Regression, and Gradient-Boosted Trees. A custom dataset was created containing short utterances labeled as either urgent (e.g., warnings or alerts) or non-urgent. For each audio file, both acoustic features (via Librosa and Parselmouth) and textual features (via TF-IDF on Italian transcriptions) were extracted. The classifiers were trained and tested on three input sets: (1) audio + text features, (2) audio-only features, and (3) text-only features. Results show that combining acoustic and textual features significantly improves classification accuracy. Among the tested models, Gradient-Boosted Trees achieved the best performance overall. These findings suggest that multimodal approaches are effective for urgency detection even in low-resource, multilingual contexts.

Keywords

audio classification, urgency detection, TF-IDF, Spark MLlib, supervised learning

1. Introduction

The objective of this project is to evaluate the performance of selected machine learning models from the **PySpark ML library** for classifying Italian audio samples as either "**normal**" or "**urgent**." Urgent speech includes utterances conveying a sense of alert or emergency, such as "Watch out! A car is coming!"

The models considered in this study are:

- **Random Forest**
- **Logistic Regression**
- **Gradient-Boosted Trees**

To enable model training and evaluation, a dedicated **dataset** was created consisting of Italian audio recordings labeled by urgency. Each recording was paired with its Italian transcription and corresponding translations in English, French, and German.

The dataset was used to construct three different types of input **samples**:

- Samples combining features extracted from both the audio (via **Librosa** and **Parselmouth**) and the transcriptions (via **TF-IDF**).
- Samples containing only audio-based features.
- Samples containing only text-based features.

All three classifiers were trained and tested on these sample sets. Their performance was then compared to assess the impact of different feature combinations on classification **accuracy**.

Notebooks and samples available at:
https://github.com/Matti02co/AudioUrgencyClassifierPySpark

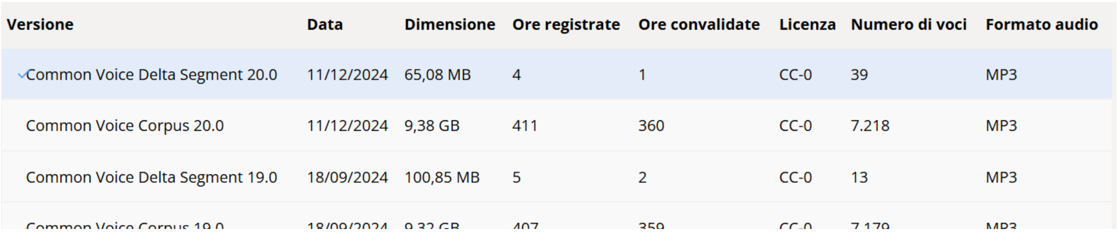
2. Dataset creation and setup

The first step in this project involved the construction of a custom audio dataset in Italian, designed to support the task of urgency classification. A total of **300** short utterances were collected, equally balanced between urgent and non-urgent speech. The target duration for each audio file was between 10 and 20 seconds, and all clips featured a single speaker. The final format chosen for the recordings was **MP3**.

2.1 Non-Urgent Audio

To obtain the 150 non-urgent samples, an initial search was conducted to identify existing publicly available datasets containing spoken Italian that met the above criteria. Several candidates were evaluated:

- **EmoFilm**: contains movie-derived clips, but most are shorter than 10 seconds.
- **DEMoS** (Database of Elicited Mood in Speech): includes emotionally expressive speech but is not openly accessible.
- **RAVDESS**: provides emotion-laden speech, though only in English.
- **Freesound**: did not contain suitable material for this task.
- **Mozilla Common Voice**: an open-source collection of read speech in various languages, including Italian.



Versione	Data	Dimensione	Ore registrate	Ore convalidate	Licenza	Numero di voci	Formato audio
Common Voice Delta Segment 20.0	11/12/2024	65,08 MB	4	1	CC-0	39	MP3
Common Voice Corpus 20.0	11/12/2024	9,38 GB	411	360	CC-0	7.218	MP3
Common Voice Delta Segment 19.0	18/09/2024	100,85 MB	5	2	CC-0	13	MP3
Common Voice Corpus 19.0	18/09/2024	9,32 GB	407	359	CC-0	7.179	MP3

Figure 1: Mozilla Common Voice snapshot.

Using Mozilla Common Voice, 68 non-urgent samples were extracted. However, due to their neutral prosody and lack of expressiveness, an additional 82 recordings were manually extracted from YouTube videos using **Audacity**, ensuring they conformed to the required format. This brought the total to 150 non-urgent clips.

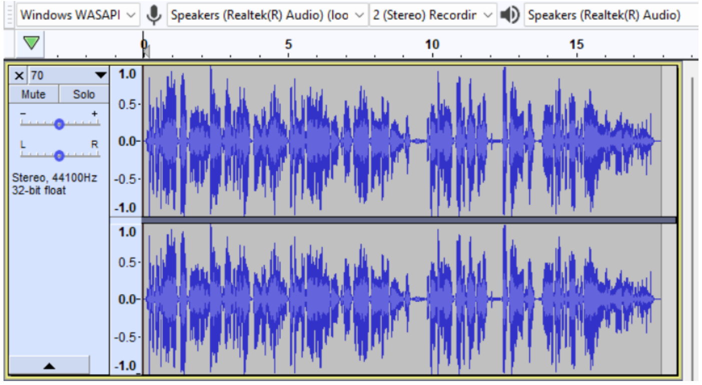


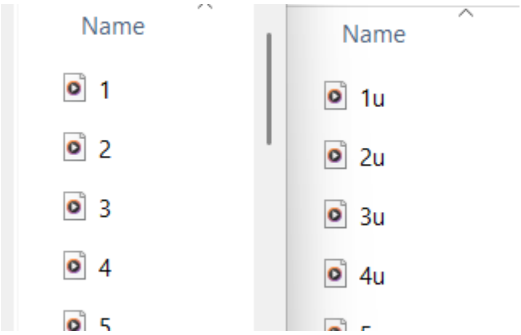
Figure 2: Audacity snapshot.

2.2 Urgent Audio

Initially, urgent audio samples were also sought on YouTube. Despite targeting specific scenarios (e.g., accidents, emergencies, livestreams), only 4 usable clips were collected due to frequent background noise, overlapping speech, or suboptimal recording quality.

To overcome this limitation, the remaining 146 urgent samples were **recorded ad hoc** by three different speakers. The phrases were generated using ChatGPT and reflected emergency or high-tension scenarios, including those involving the use of AR/VR headsets in shared environments.

To maintain consistent file naming and facilitate later processing, all 300 audio files were renamed sequentially (1.mp3, 2.mp3, ..., 150.mp3). Urgent files were suffixed with the letter u (e.g., 23u.mp3), enabling easy identification and ensuring filename uniqueness within a single directory.



Name
1
2
3
4
5

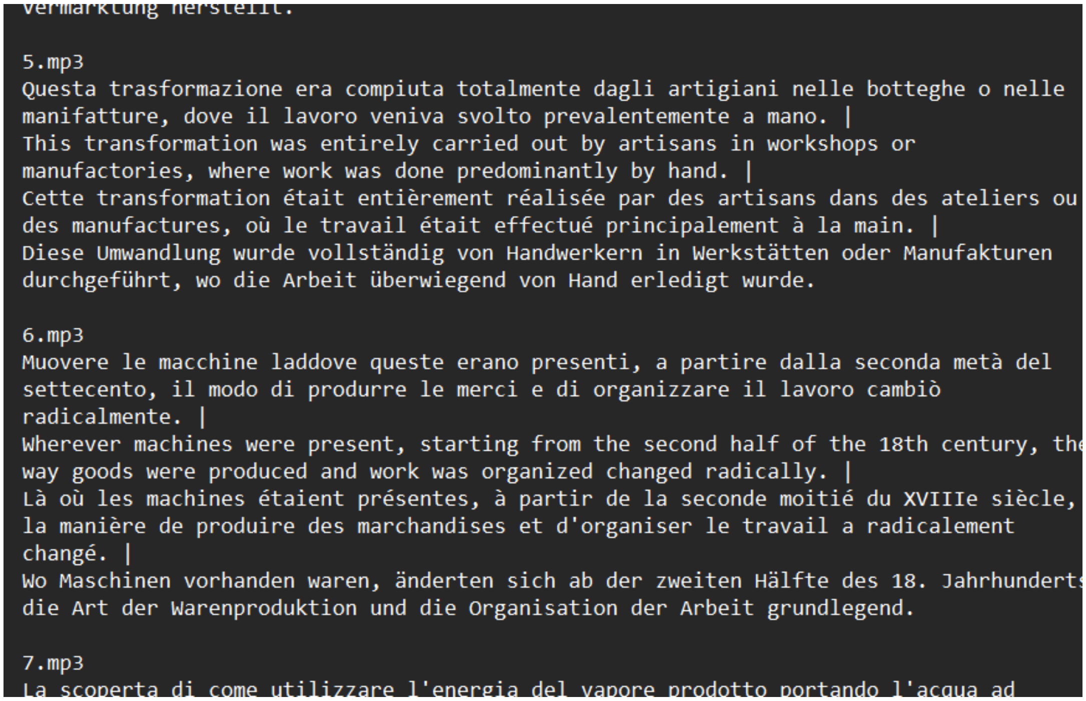
Name
1u
2u
3u
4u

Figure 3: File naming for easy identification.

2.3 Transcriptions

After organizing the audio files, transcriptions were prepared for each sample. For recordings generated via ChatGPT, the transcript was already available. To streamline the data loading process during model training, a single text file named “Trascrizioni.txt” was created. Each entry in the file includes:

- The filename
- The transcription in Italian
- The corresponding translations in English, French, and German
- Entries are separated by blank lines for clarity



```
Vermarktung herstellt.

5.mp3
Questa trasformazione era compiuta totalmente dagli artigiani nelle botteghe o nelle
manifatture, dove il lavoro veniva svolto prevalentemente a mano. |
This transformation was entirely carried out by artisans in workshops or
manufactories, where work was done predominantly by hand. |
Cette transformation était entièrement réalisée par des artisans dans des ateliers ou
des manufactures, où le travail était effectué principalement à la main. |
Diese Umwandlung wurde vollständig von Handwerkern in Werkstätten oder Manufakturen
durchgeführt, wo die Arbeit überwiegend von Hand erledigt wurde.

6.mp3
Muovere le macchine laddove queste erano presenti, a partire dalla seconda metà del
settecento, il modo di produrre le merci e di organizzare il lavoro cambiò
radicalmente. |
Wherever machines were present, starting from the second half of the 18th century, the
way goods were produced and work was organized changed radically. |
Là où les machines étaient présentes, à partir de la seconde moitié du XVIIIe siècle,
la manière de produire des marchandises et d'organiser le travail a radicalement
changé. |
Wo Maschinen vorhanden waren, änderten sich ab der zweiten Hälfte des 18. Jahrhunderts
die Art der Warenproduktion und die Organisation der Arbeit grundlegend.

7.mp3
La scoperta di come utilizzare l'energia del vapore prodotto portando l'acqua ad
```

Figure 4: Trascrizioni.txt snapshot.

This structure ensured that each sample’s textual representation could be quickly accessed and aligned with its audio counterpart during feature extraction and sample creation.

2.4 Environment Setup

Before training the models, in order to access the data easily from **Google Colab**, both the audio files and the transcription file Trascrizioni.txt were uploaded to **Google Drive**, specifically under the directory /content/drive/MyDrive/audiozzi.

Several preliminary steps were then performed in the Colab notebook, including:

- Installation of **Apache Spark**
- Mounting of Google Drive to enable data access
- Installation or import of additional libraries and classes as needed (e.g., Parselmouth)

3. Sample Generation

At this stage, the process of generating the training samples was initiated. The samples were created by **concatenating features** extracted from the audio recordings with features derived from their corresponding transcriptions. To facilitate this process and enable clearer comparisons, three dedicated functions were implemented:

- **extract_audio_features**: extracts MFCC and RMSE features from the audio using the Librosa library, aiming to capture both the timbre and the temporal energy of the voice.
- **extract_pitch**: computes the average pitch of the audio using Parselmouth.
- **extract_text_features**: represents the transcription using a TF-IDF vectorization.

The decision to extract these particular features was based on their effectiveness in characterizing urgent speech. **MFCC** and **RMSE** are suitable for distinguishing between calm and agitated vocal tones (e.g., shouting), while **pitch** typically increases in urgent speech. **TF-IDF**, on the other hand, provides a numerical representation of the text by analyzing the frequency of each term in relation to the entire corpus, helping identify both rare and common terms typically associated with different contexts.

An additional function was implemented to parse the transcription file and build a **dictionary** that maps each audio file name to its corresponding transcription, enabling fast and reliable access during sample creation.

A loop was then used to process all audio files stored in the Google Drive directory:

- Only files with the .mp3 extension were considered.
- A label was assigned to each sample to indicate whether the audio was urgent or not, based on the filename (urgent files end with "u.mp3").
- The transcription was retrieved from the dictionary.
- MFCC, RMSE, pitch, and text features were extracted.
- All extracted features were concatenated, and a sample was created containing the filename, feature vector, transcription, and label.

All generated samples were serialized and saved into a .pkl **binary file** in the same Drive directory. This approach allows for efficient reuse of pre-processed datasets and avoids repeating the time-consuming feature extraction phase (which takes several minutes for 300 audio files). This strategy is particularly beneficial when working with larger datasets.

4. Model Training and Evaluation

Once the samples were generated, it was possible to proceed with training and evaluating the performance of the classifiers.

The samples were loaded from the binary file and converted into a Spark DataFrame, which was then randomly split into a **training set** (80%) and a **test set** (20%).

Three classifiers were selected for evaluation: **Random Forest**, **Logistic Regression**, and **Gradient Boosted Trees** (GBT). These models were chosen due to their differing strengths and internal mechanisms: Random Forest is known for its robustness when handling complex numerical features, Logistic Regression offers simplicity and efficiency for binary classification problems, and GBT provides powerful predictive capabilities as a more sophisticated ensemble method.

Each model was trained on the training set and then evaluated on the test set using **accuracy** as the primary performance metric. With samples including all available features (MFCC, RMSE, pitch, and TF-IDF), the average accuracy scores obtained were:

- Random Forest: **98.08%**
- Logistic Regression: **100%**
- Gradient Boosted Trees: **96.15%**

Repeated tests using different randomly generated training-test splits confirmed the stability of these results, with accuracy consistently **above 94%.**

Among the models, GBT tended to yield the lowest accuracy, while both Random Forest and Logistic Regression consistently approached perfect classification.

4.1. Librosa & Parselmouth vs TF-IDF

To better understand which features contributed most significantly to the classification task, the same models were tested on two alternative sets of samples:

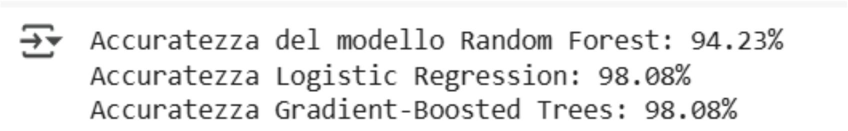
- Samples containing **only audio-derived features**: MFCC, RMSE, and pitch (extracted using Librosa and Parselmouth).
- Samples containing **only textual features**, represented using TF-IDF.

This was achieved by slightly modifying the sample generation pipeline:

- For the audio-only dataset, the text feature extraction step was removed.
- For the text-only dataset, all audio feature extraction steps (MFCC, RMSE, and pitch) were skipped.

Each of the two sample sets was saved as a separate binary file for convenience. Model training and testing were carried out following the same methodology: random 80-20 split and evaluation using accuracy.

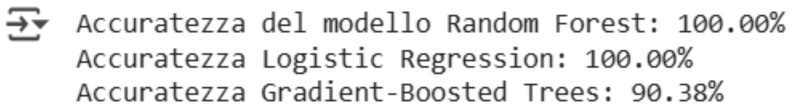
Results showed that the average accuracy across models remained relatively high for both types of features, with a mean accuracy of approximately **96.7%** in both cases. However, a closer inspection revealed significant differences in the performance of each model depending on the feature set:



```
Accuratezza del modello Random Forest: 94.23%
Accuratezza Logistic Regression: 98.08%
Accuratezza Gradient-Boosted Trees: 98.08%
```

Figure 5: audio-based model accuracy.

With audio-only features, Random Forest underperformed compared to Logistic Regression and GBT.



```
Accuratezza del modello Random Forest: 100.00%
Accuratezza Logistic Regression: 100.00%
Accuratezza Gradient-Boosted Trees: 90.38%
```

Figure 6: text-based model accuracy.

With text-only features, both Random Forest and Logistic Regression significantly outperformed Gradient Boosted Trees.

These findings suggest that textual features may be more linearly separable, making them particularly well-suited for simpler models such as Logistic Regression, while the nature of audio features may require models better tuned for complex numeric patterns.

Investigating the cause of the perfect performance of Random Forest and Logistic Regression on text-only features, 10+ strongly urgency-related words were found in transcriptions and their removal led to a significant performance drop. This explains the effectiveness of TF-IDF in urgency classification.

5. Conclusions

In conclusion, the classifiers tested from Spark's machine learning library demonstrated a **strong ability** to distinguish between urgent and non-urgent audio samples across all three configurations tested, consistently achieving accuracy scores above 90%.

Following the experiments with separated audio and text features, results suggest that the slightly lower performance of the Gradient Boosted Trees model may be attributed to its relative difficulty in handling text-based features, especially when compared to Random Forest and Logistic Regression.

Conversely, the marginally lower performance of Random Forest compared to the other two models when using only audio features could indicate a small disadvantage in capturing the relevant acoustic patterns.

A particularly noteworthy finding was the effectiveness of TF-IDF in classifying transcriptions with **semantically distinct vocabularies**. The excellent performance on the text-only samples can be largely attributed to the presence of urgency-related keywords, which TF-IDF captured effectively due to its emphasis on rare but informative terms within the dataset.

Overall, the results are promising for the task of urgency classification in audio data using models available in pyspark.ml. Based on the findings, the best-performing configurations for practical use could be summarized as follows:

- **Logistic Regression** using both audio and textual features.
- **Random Forest** or **Logistic Regression** when only TF-IDF-based textual features are available.

Future work may include expanding the dataset for both training and testing, exploring feature selection techniques to reduce dimensionality, and testing the models on multilingual datasets to assess generalization across languages.

5. References

[1] Original university project repository in Italian: https://github.com/Matti02co/BigData

Developed during the Big Data course at University of Cagliari, 2025.

6. License