



```
MovieNet003.ipynb ☆
File Modifica Visualizza Inserisci Runtime Strumenti Guida Data ultima modifica: 9 giugno

+ Codice + Testo

# https://kgptalkie.com/multi-label-image-classification-on-movies-poster-using-cnn/
!pip3 install tensorflowjs

import tensorflow as tf

from tensorflow.keras import Sequential
from tensorflow.keras.layers import Flatten, Dense, Dropout, BatchNormalization, Conv2D, MaxPool2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing import image
from tensorflow.keras.callbacks import ModelCheckpoint, TensorBoard
from tensorflow import keras
import tensorflowjs as tfjs
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from tqdm import tqdm
import os
import datetime
from google.colab import drive
import zipfile
import gc
from IPython import get_ipython
from skimage.util import random_noise
import cv2
from operator import itemgetter
```

Movie Poster Classifier

22/06/2021
(rev. 2)

Ugo Marchesini

Matricola : 0000977872

Università di Bologna

Dipartimento di Informatica - Scienza e Ingegneria

Corso di Laurea Magistrale in Ingegneria Informatica

Esame di Sistemi Digitali



Panoramica

Il poster cinematografico è il biglietto da visita di un film.

Sintetizzano in una immagine una storia e delle emozioni.

Devono comunicare al potenziale spettatore quale tipo di spettacolo verrà offerto ovvero a quale genere appartiene : *«A tutti i livelli del processo di produzione e fruizione dei film, quindi, i generi aiutano ad assicurare che la maggior parte dei membri di una società condivida almeno delle nozioni generali sui molti film che si contendono la nostra attenzione»* (Bordwell e Thompson).

Nella pagina di Wikipedia di Generi Cinematografici è possibile trovare questa affermazione : *"In Saper vedere il cinema, Antonio Costa spiega che la classificazione del film in generi è uno degli aspetti fondamentali dell'istituzione cinematografica. L'etichetta del genere orienta il semplice spettatore su che cosa aspettarsi da un certo film."*

Nonostante il poster sia un'opera d'arte associata ad un'opera d'arte (il film stesso) , e quindi soggetto alla variabilità della sensibilità artistica, è possibile per un osservatore umano riconoscere il/i generi del film a partire dal poster.

Perchè non farlo fare ad una macchina?

Un **rete neurale convoluzionaria supervisionata** permette di classificare delle immagini con le relative label.

Essendo arte si pone dei problemi di tipo storico , stilistico e di convenzioni nella valutazione dei generi.

I poster sono classificati con molti generi e quindi entriamo nel caso di una **classificazione multi-label**.

I poster degli anni '20 del secolo scorso sono principalmente dei disegni , ma negli ultimi decenni prevalgono delle foto , le animazioni classiche della Disney hanno uno stile bidimensionale ma i film di animazioni più recenti hanno stili che prediligono il 3D , etc.

Questo implica che un classificatore che sia addestrato su un range temporale molto esteso potrebbe non riconoscere il genere , per questo il progetto è indirizzato a classificare per anni.

La valutazioni dei generi è una operazione che risente una certa soggettività che diventa evidente per il genere drammatico . La Treccani riporta questa valutazione : *"Dunque una definizione del 'drammatico' come genere appare molto difficile per il cinema e possibile, prima di tutto, in negativo. I toni devono essere seri, problematici, non lievi e rasserenanti come accade nella commedia; i sentimenti possono essere in primo piano, ma non nei termini esasperatamente romantici del melodramma; l'azione non deve prevalere come nell'avventuroso (con le sue declinazioni di western e di bellico); né canzoni e musiche devono essere più importanti delle parti dialogate, come nei musical; i misteri svelati dall'intreccio nella parte finale non devono essere lo scopo della narrazione, come nei vari tipi di poliziesco, con le declinazioni di noir, film di spionaggio, o*



*film di gangster; le immagini dell'ambiente non devono essere più importanti delle vicende dei personaggi di finzione, perché altrimenti ci si colloca in un ambito affine al documentario; e d'altra parte le ambientazioni non possono essere troppo irrealistiche e, soprattutto, non vi possono essere miracoli o soluzioni trascendenti, a opera di esseri soprannaturali, celesti o infernali che siano, perché altrimenti ci si colloca nel fantastico. Sembrerebbe dunque che, procedendo così per esclusione, lo spazio del drammatico filmico finisca per essere estremamente ridotto, fino quasi ad annullarsi. **E invece, se si consultano i dizionari del cinema che contengono nelle schede filmografiche anche la classificazione per generi, la voce 'drammatico' si rivela frequentissima. Proprio perché tale viene definito tutto quanto non può rientrare nelle altre categorie.***

Questa valutazione sulla classificazione dei generi, in particolare per il genere drammatico, pone un limite sulla accuracy di una rete predittiva. Dato che un addestramento supervisionato richiede una label associata al poster e tutto quello che non può essere inquadrato in un genere rientra nel genere drammatico il risultato che qualsiasi predizione avrà una componente di drammatico.

Per esperienza personale posso aggiungere che le classificazioni offerte da manuali cinematografici o siti più o meno professionali risentono comprensibilmente della valutazione soggettiva del critico.

Obiettivi

L'obiettivo di questo progetto è la creazione di una web application che permetta di riconoscere il genere cinematografico a partire dal poster.

Vengono elencati i sotto obiettivi per completare il progetto :

1. Identificazione ed ottimizzazione della rete
2. Sperimentare ambienti di esecuzione differenti
3. Raccolta delle metriche per la valutazione dell'efficienza della rete



Esempio di un progetto esistente , limiti e considerazioni generali

Ci sono vari esempi di progetti di cnn per la classificazione dei poster cinematografici ho preso come esempio iniziale quello a questo [link](#) , che d'ora in poi sarà chiamato KgpNet .

Dopo alcuni tentativi sul mio desktop (ASUSPC) sono emersi dei limiti :

1. la ram di 8GB è un collo di bottiglia e mi sono orientato verso Colab
2. L'accuracy di KgpNet ,con il suo dataset, non è stata possibile replicarla , da qui la necessità di individuare una rete con migliore accuracy.
3. la dimensione delle immagini è risultata critica nella velocità di elaborazione
4. il range temporale del dataset KgpNet è riferito agli anni '80 quindi troppo ristretto per lo scopo della rete
5. Alcuni generi sono troppo rappresentati o troppo sotto rappresentati
6. Un dataset esteso nell'ordine di 1k immagini per anno implica che gli stili grafici spaziano tra quello statunitense , che è il più diffuso , indiano , cinese ed altri. L'ipotesi che si può fare inizialmente è che stili differenti potrebbero influire sulla accuracy.
7. Un qualsiasi dataset potrebbe contenere immagini che non sono poster ma : foto di scena , scannerizzazioni delle copertine delle VHS o scannerizzazioni dei poster cartacei. Anche in questo caso l'accuracy ne potrebbe risentire.



Tappe intermedie

Vengono elencate le tappe , non necessariamente in sequenza, per completare il progetto :

I. Raccolta delle immagini e delle labels detto dataset

Al fine di raccogliere le immagini e le labels ho sviluppato un'applicazione spider (in C#) che ha scaricato le immagini e le labels da un famoso sito di cinema. Il sito presenta le informazioni in formato json ed è stato necessario trasformarle in un file csv più gestibile per le librerie python/Tensorflow .

II. Google Drive come file system per Colab

Per una più efficiente gestione da parte di Colab le immagini sono compresse per anno in file zip e le relative labels in file csv e caricate su Google Drive (GD). Colab permette di accedere velocemente a GD in particolare è molto performante la gestione di file compressi.

GD è anche usato come spazio per salvare i sorgenti (.ipynb) , i modelli creati (.h5 e Tfls) , i log e i file di history delle metriche.

III. Ricerca della/delle cnn più adatte alla classificazione

Un articolo che affronta il problema della classificazione di poster cinematografici suggerisce l'utilizzo di ResNet50.

La tabella comparativa sul sito di Keras.io offre una panoramica alle reti rese disponibili da Keras , e ResNet50 ha un buon compromesso tra accuracy e spazio occupato dal modello.

IV. Scrittura del codice per la creazione del modello (cnn addestrata)

Lo script principale per l'addestramento della rete è salvato su un account Google ([MovieNet](#)).

Le funzionalità dello script prevedono : la creazione delle directory in GD ,caricamento del dataset, creazione del model , addestramento del modello , conversione da h5 a Tfls e salvataggio delle metriche.



V. Sperimentare ambienti per l'elaborazione (on line , pc)

Dopo aver scritto i primi gli script python in Visual Studio 2019 fatto alcuni test su PC , sia su CPU che GPU, e contemporaneamente su Colab. E' risultato più efficiente passare a Colab con il runtime GPU , la performance usando la CPU Colab è paragonabile a quella del PC.

Il limite della RAM di Colab è 12.7 GB nella versione gratuita.

VI. Ottimizzazione della rete

La ricerca sui parametri ottimali ha portato a queste conclusioni :

Immagine a 256x256 per rimanere nei limiti di memoria di Colab.

Epoch = 100 , l'addestramento termina prima per early stopping a causa dell'aumento della loss.

Funzione di loss = `binary_crossentropy`.

Learning rate = 0.001 , valore di default.

Batch_size = 32 , valore di default.

Model checkpoint metric = `val_binary_accuracy (max)`

Early Stopping metric = `val_loss (min)`

Early stopping patience = 20 .

Test_size = 0.15 , 85% immagini per il training e 15% per la validazione.

La scelta di usare la `val_binary_accuracy` come metrica per il model checkpoint è derivata dal fatto che essendo la somma dei truepositive + truenegative / count è una misura migliore per salvare il best model durante l'addestramento.

L'accuracy di default è la `categorical_accuracy` che invece misura la frequenza con cui l'indice del valore predetto più alto è uguale all'indice del valore true più alto , ma dato che ci possono essere più generi non è garantito che coincidano , mostrando , come è ben visibile nel capitolo metrics , una sottovalutazione della "bontà" del modello.



VII. Imbalance Data

La presenza di imbalance data , ovvero alcuni generi sono sovra rappresentati in particolare il drammatico , ha portato alla ridefinizione del dataset con immagini oversampled e undersampled.

La ridefinizione non è avvenuta sulla base dei generi singoli ma di generi composti , ovvero sono stati definiti dei generi “compose” che sono la composizione dei generi singoli identificandoli con degli interi ottenuti con un’operazione in base 2 del vettore dei generi.

Al dataset finale sono stati imposti dei limiti : è stato imposto un limite dell'1% del dataset originale per ogni genere compose su cui fare oversampled ed è stato imposto un limite del 4% del dataset originale per ogni genere compose su cui fare undersampled.

Ovvero nessun genere compose è fuori dal range 1-4 % rispetto al dataset originale.

Questa ridefinizione ha portato ad una triplicazione del dataset originale.

VIII. Porting del model da python (.h5) a Tensorflow.js

La conversione del model da h5 a Tsjs è necessaria per la fase di deploy sulla web app.

IX. Hosting

La web app è ospitata su [Azure](#).

Il dataset è ospitato su [GitHub](#).

I model e le metriche sono ospitati su [GitHub](#)

I sorgenti sono ospitati su [GitHub](#)

I file per il training sono ospitati su GD.

X. Sviluppo della web application

Il ciclo di vita della web application segue questi passi :

1. Creazione della table delle predizione
2. Caricamento dei modelli Tsjs da GitHub
3. L'utente effettua l'upload del poster
4. Predizione del poster per gli anni dal 2000 al 2019
5. Scrittura delle predizioni nella table dei primi 6 generi di ogni anno predetto

La web applicazione è stata sviluppata sullo stack Asp.Net core\C# per il lato server e Html\ Javascript per il lato client.



Le metriche

In questo capitolo sono presenti i grafici che riguardano le metriche divise per anno.

Dal 2009 sono state aggiunte le top_k_categorical_accuracy per k=1,3,5.

La metrics accuracy per la loss binary_crossentropy è categorical_accuracy , che coincide con la top_1_categorical_accuracy come è visibile dai grafici.

I tempi di esecuzione sono 39 secondi per epoch.

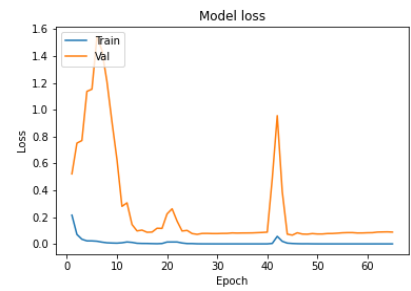
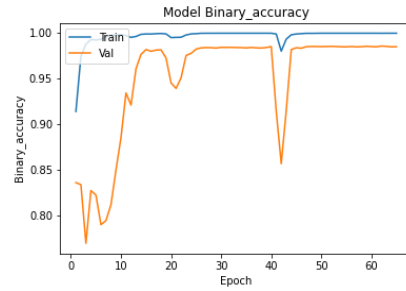
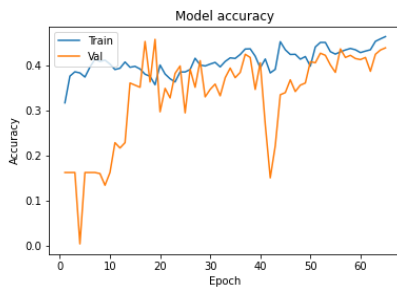
Dataset

Year	N° Image Dataset Orig.	N° Image dopo OU	N° Epoch
2000	992	2967	65
2001	991	2936	40
2002	990	2985	47
2003	988	2957	62
2004	999	3075	34
2005	998	2997	61
2006	999	3043	45
2007	999	3147	49
2008	1000	3446	58
2009	999	3131	57
2010	999	3096	46
2011	997	3037	42
2012	1000	3349	38
2013	999	2972	48

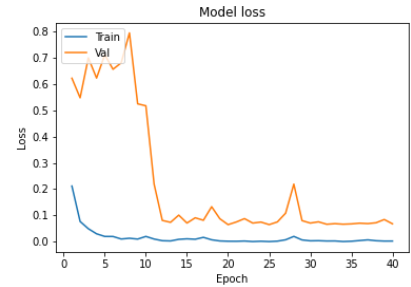
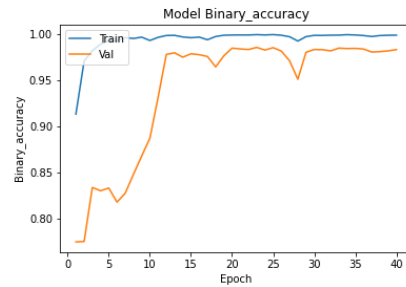
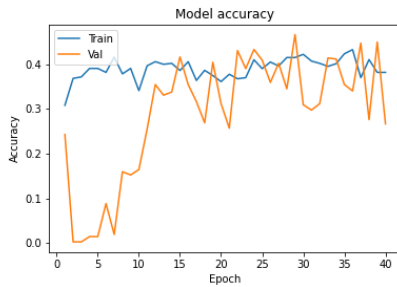


2014	1000	3505	54
2015	1000	3590	34
2016	1000	3623	46
2017	1000	3731	40
2018	1000	3547	34
2019	1000	3611	43

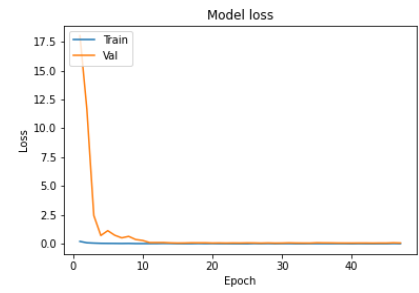
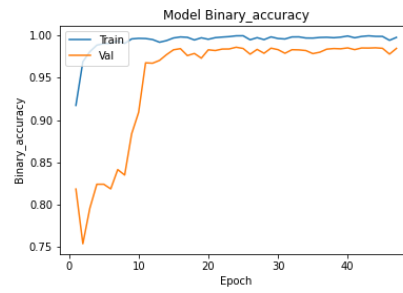
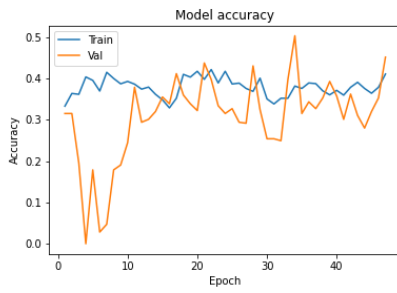
Year 2000



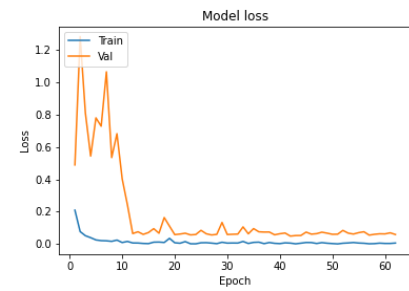
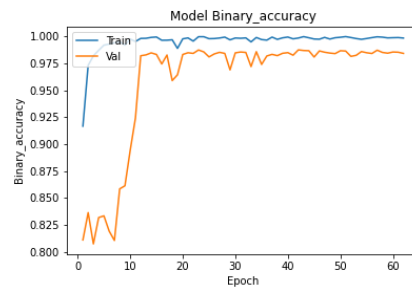
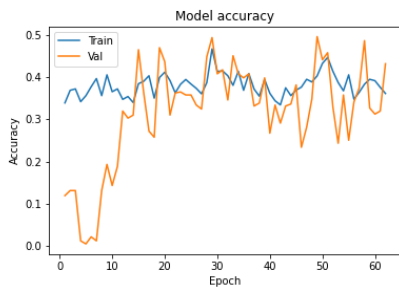
Year 2001



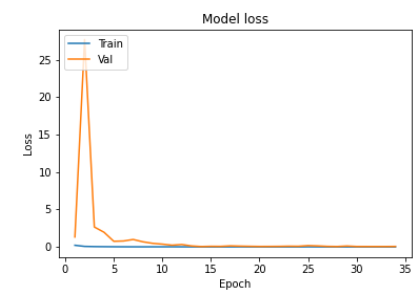
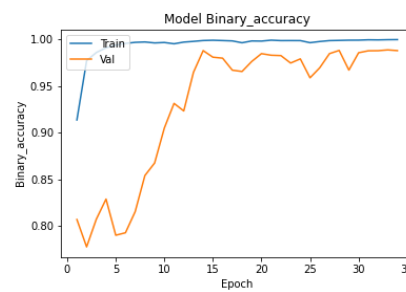
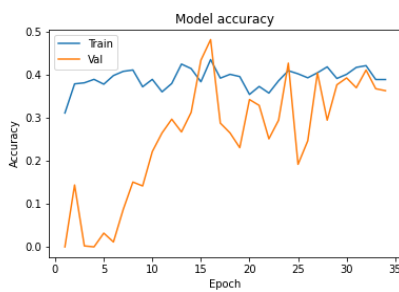
Year 2002



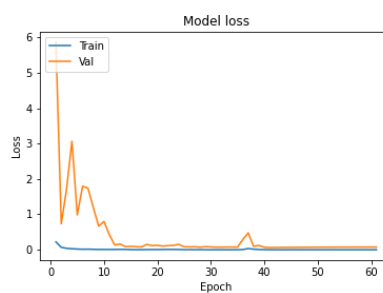
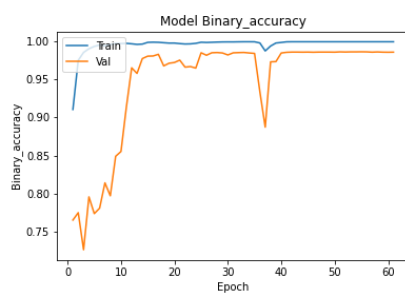
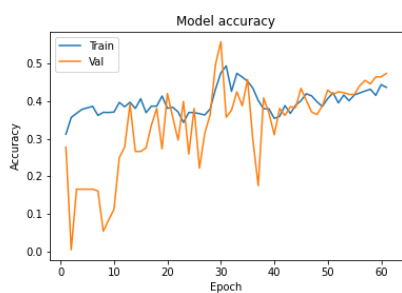
Year 2003



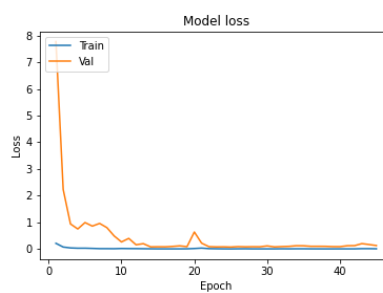
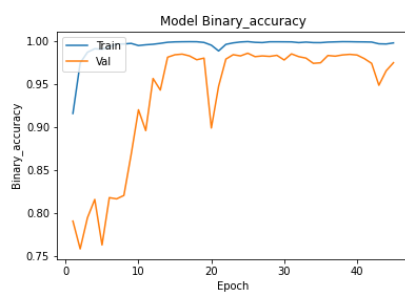
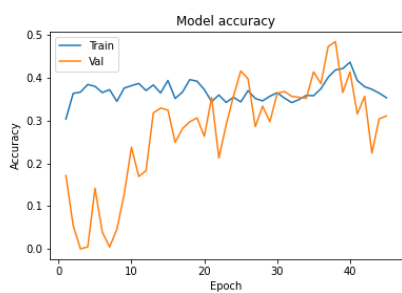
Year 2004



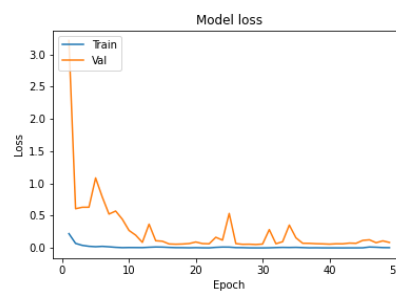
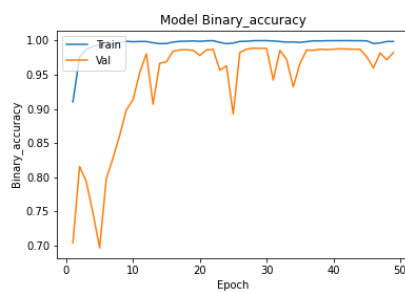
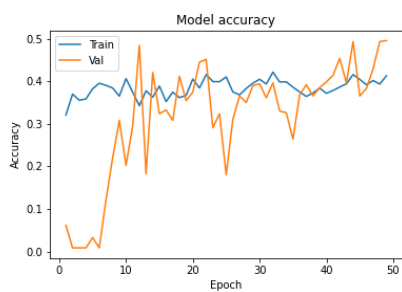
Year 2005



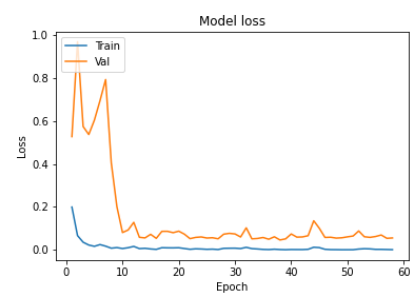
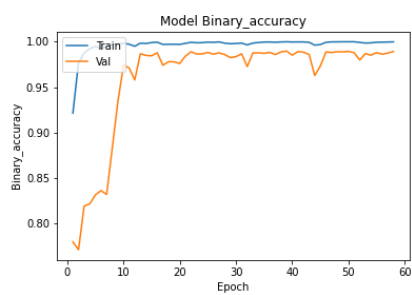
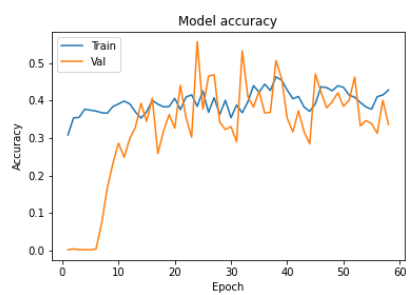
Year 2006



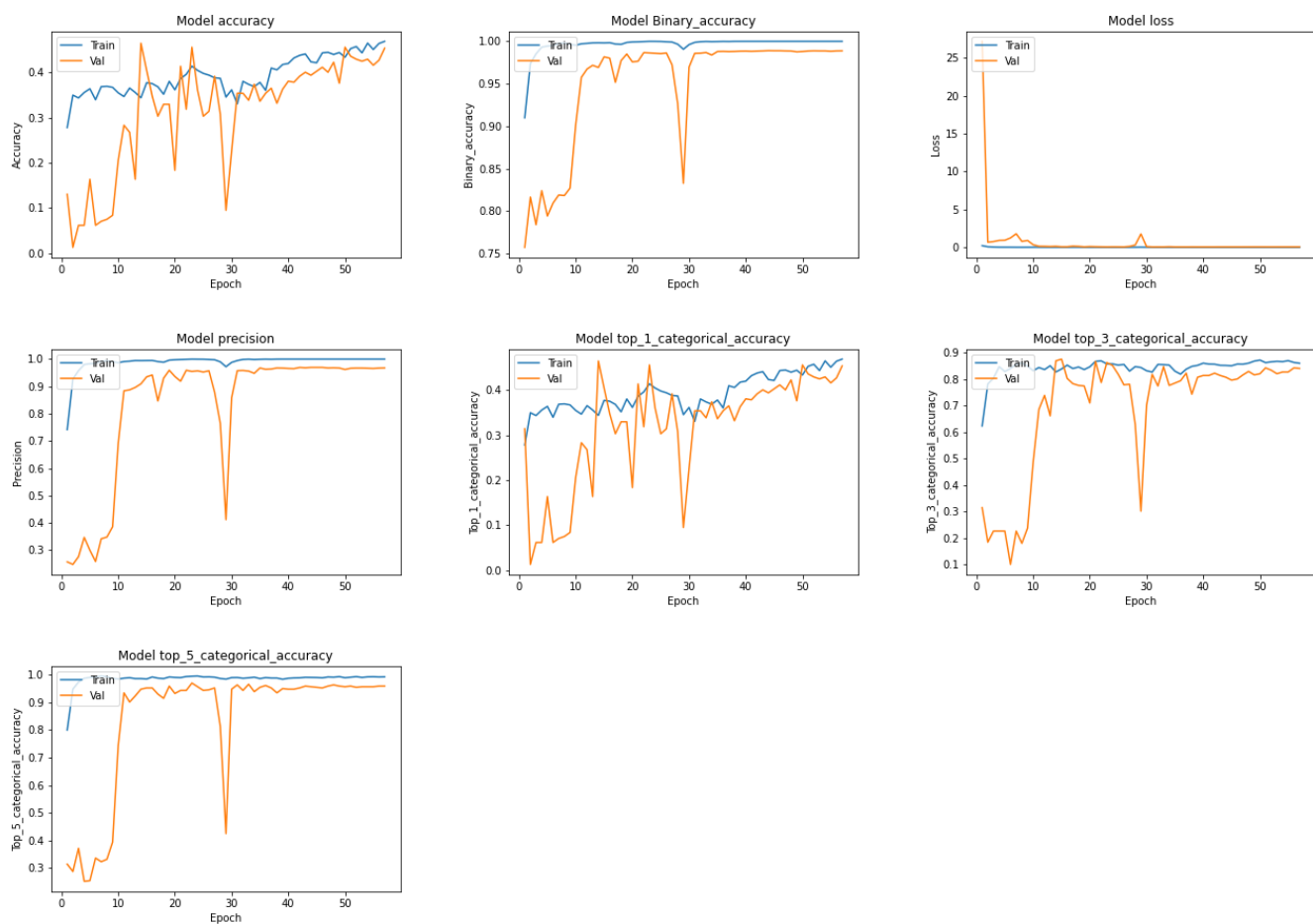
Year 2007

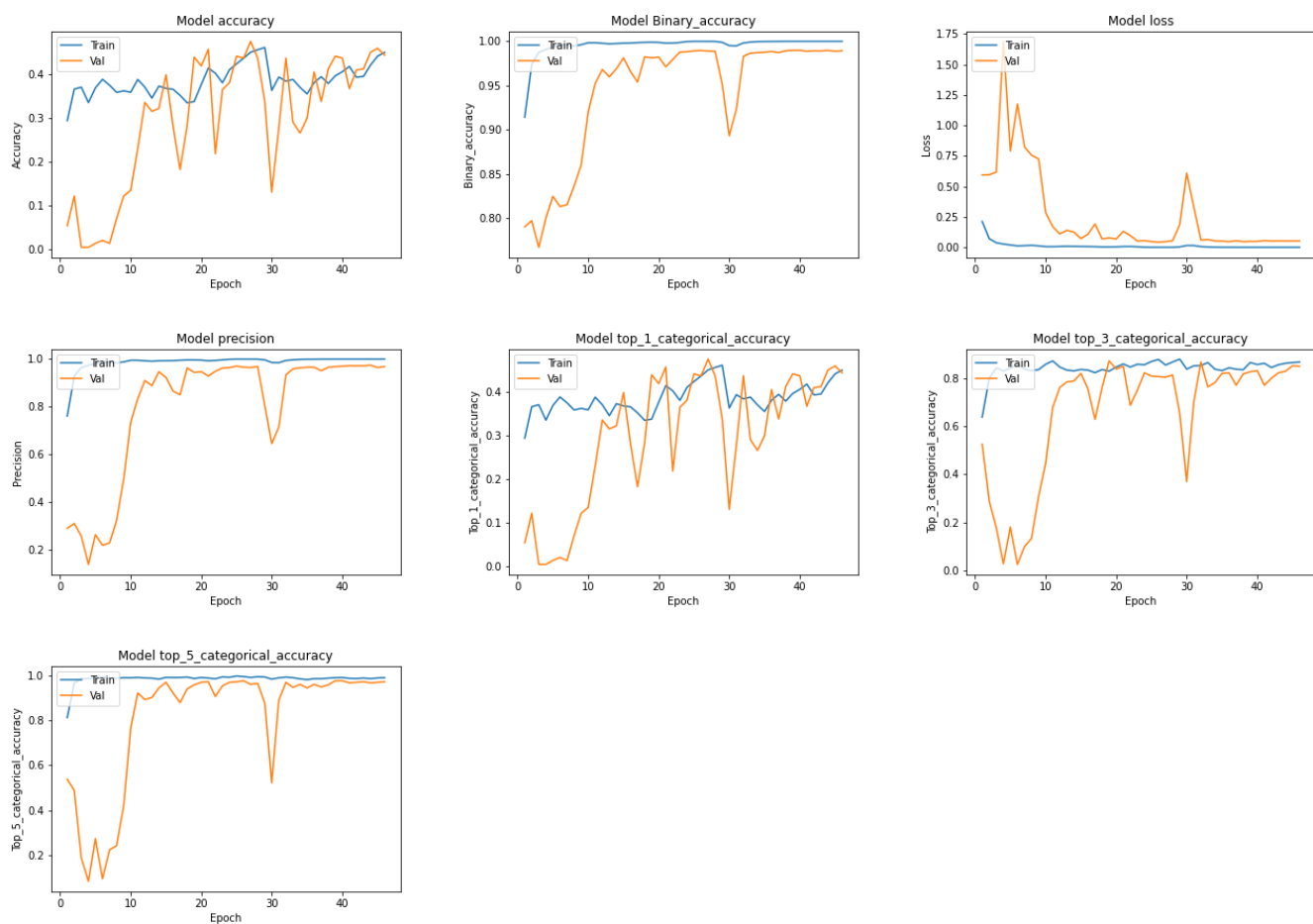


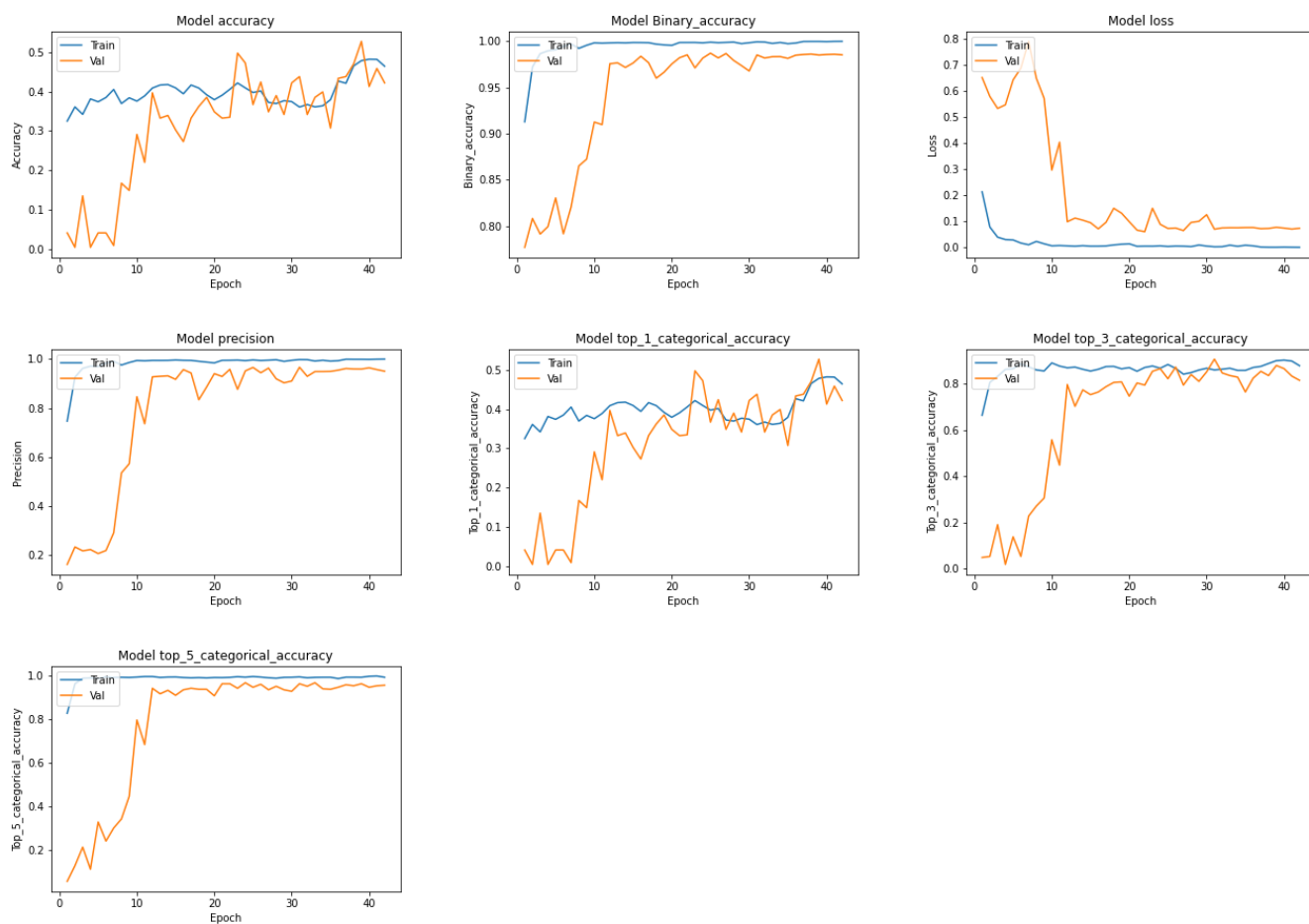
Year 2008

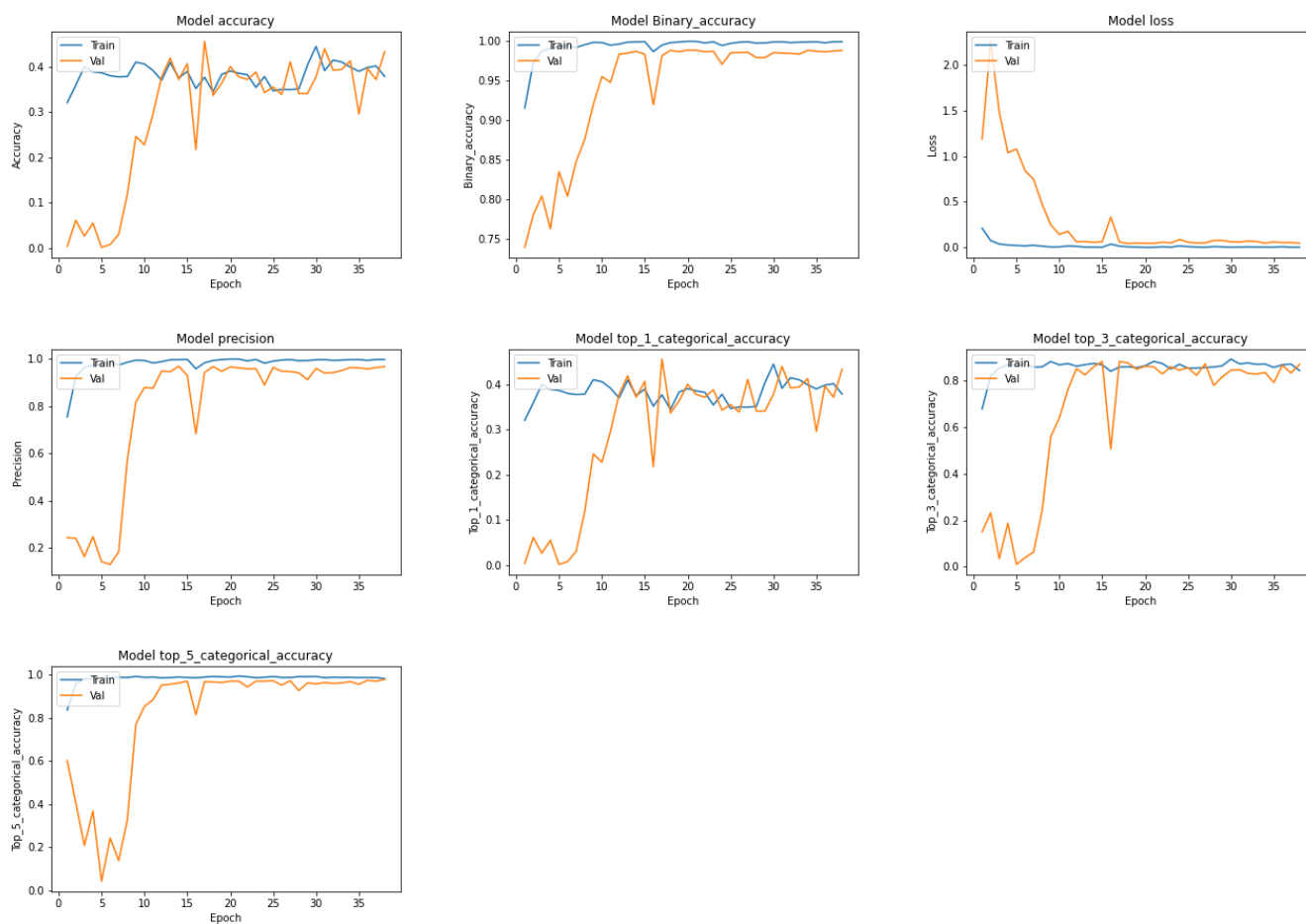


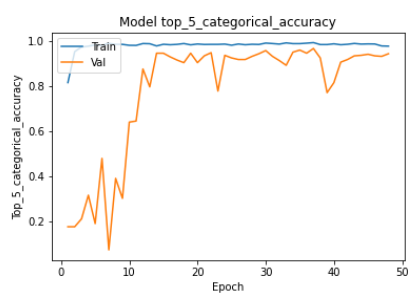
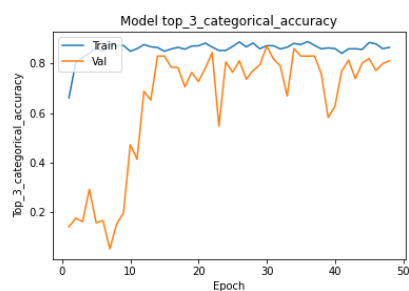
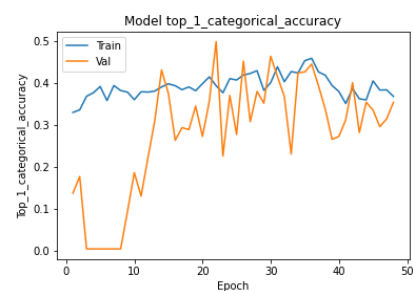
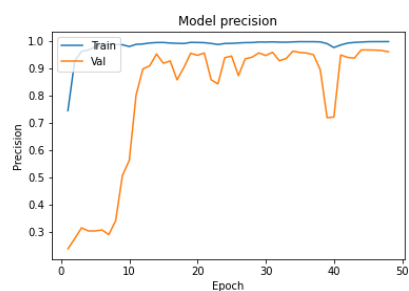
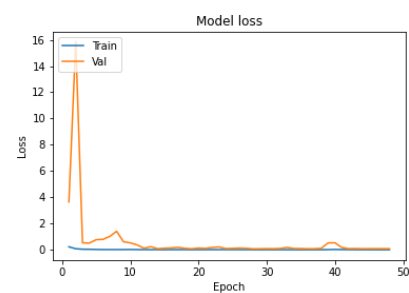
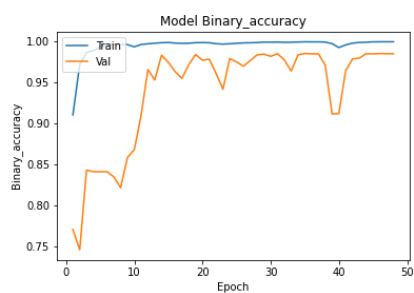
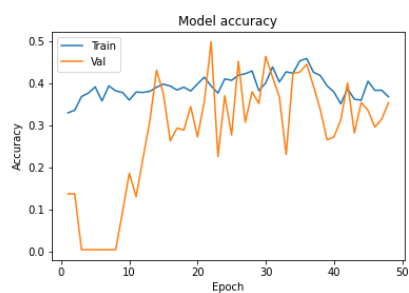
Year 2009

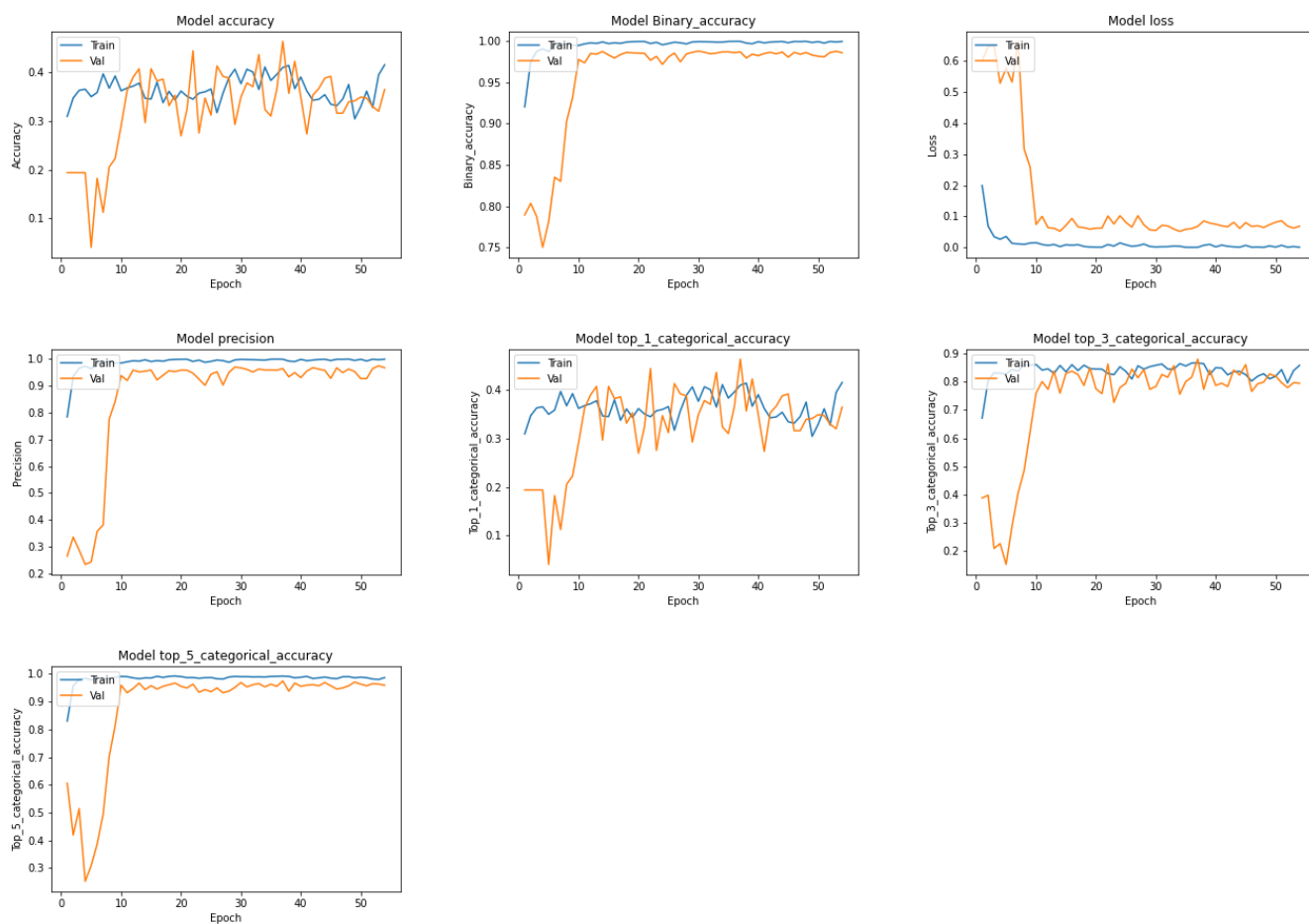


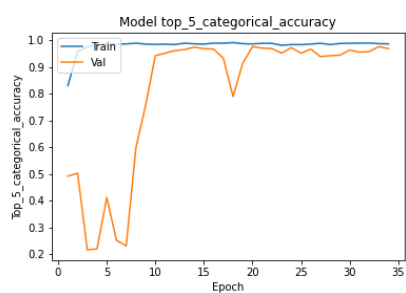
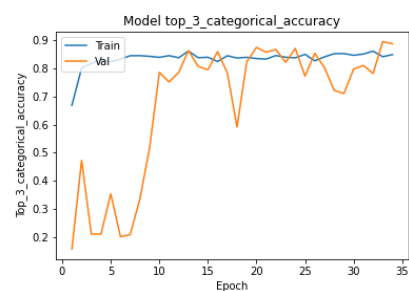
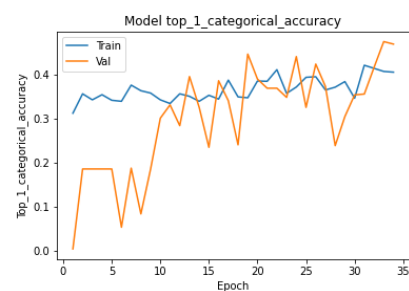
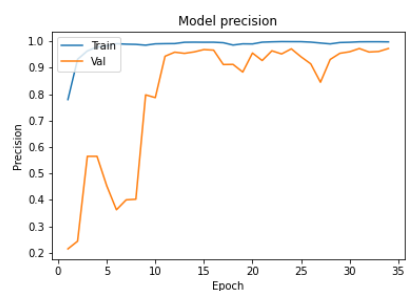
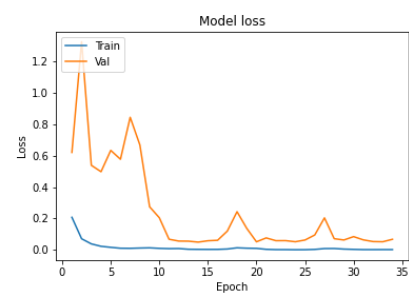
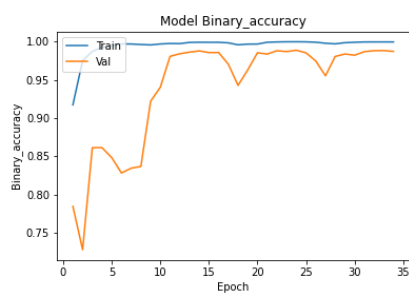
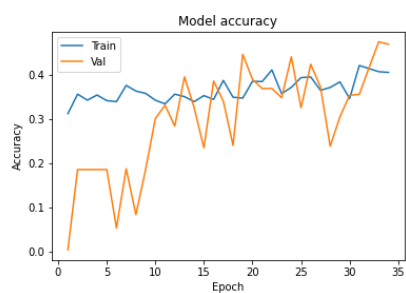


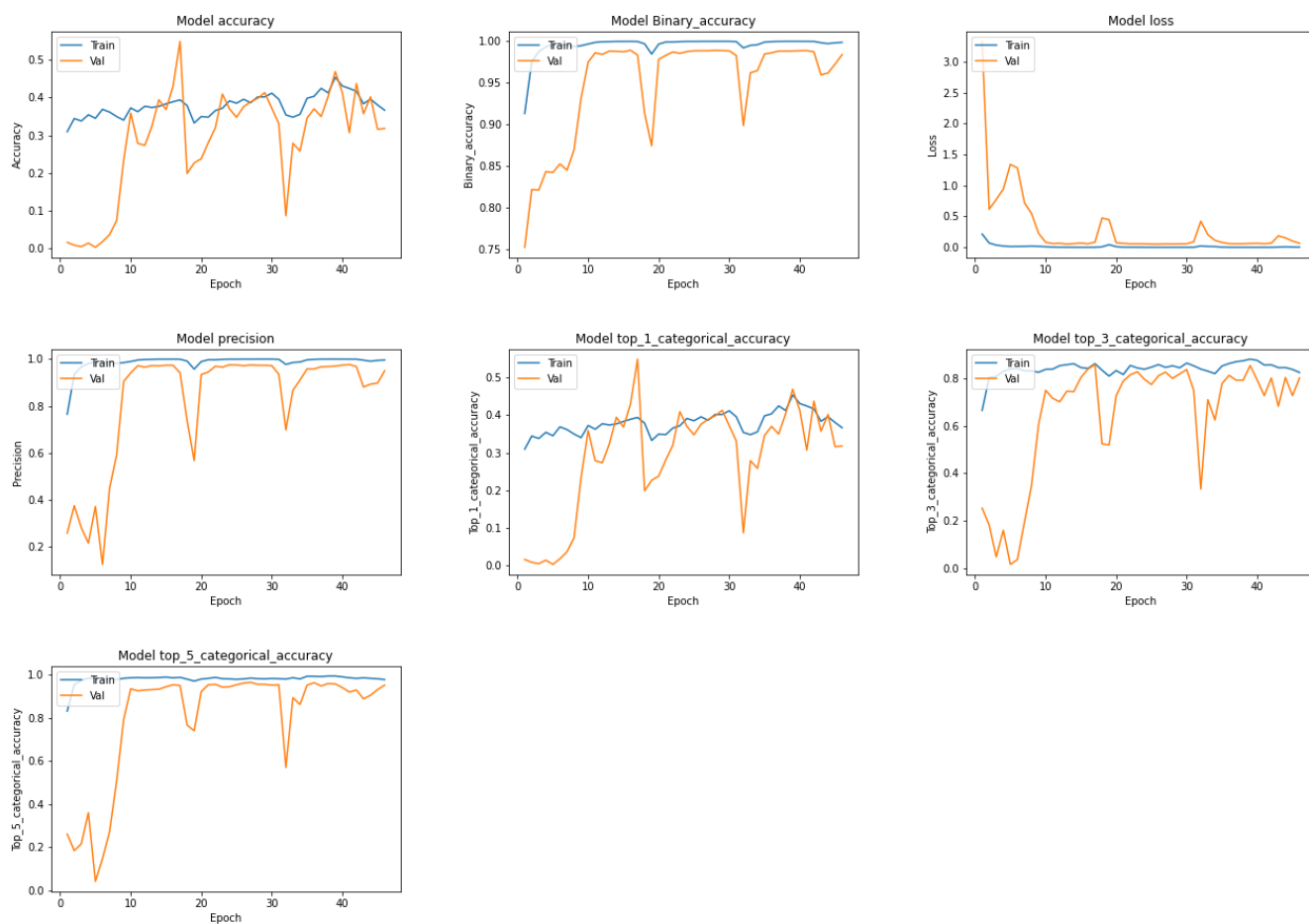


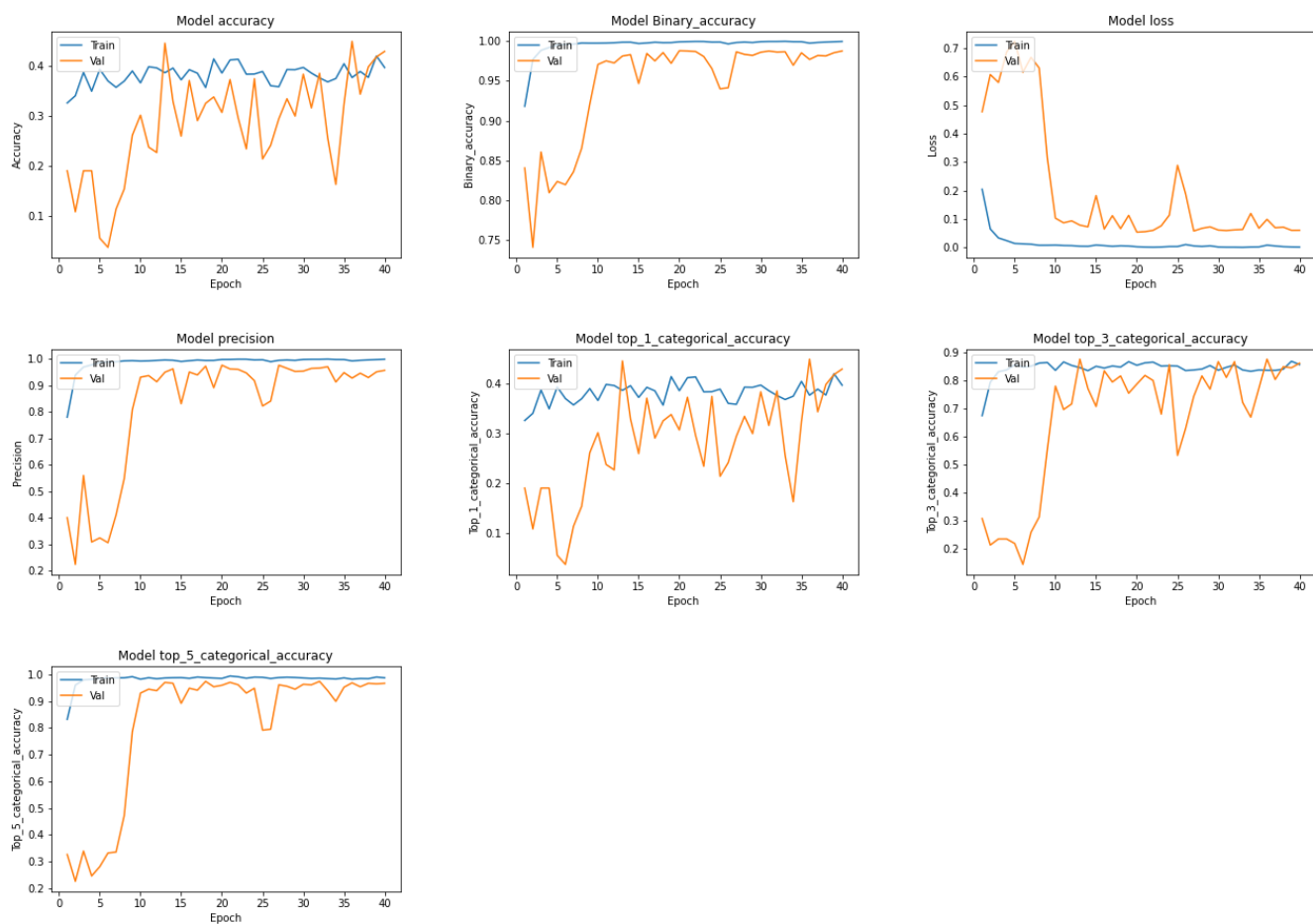


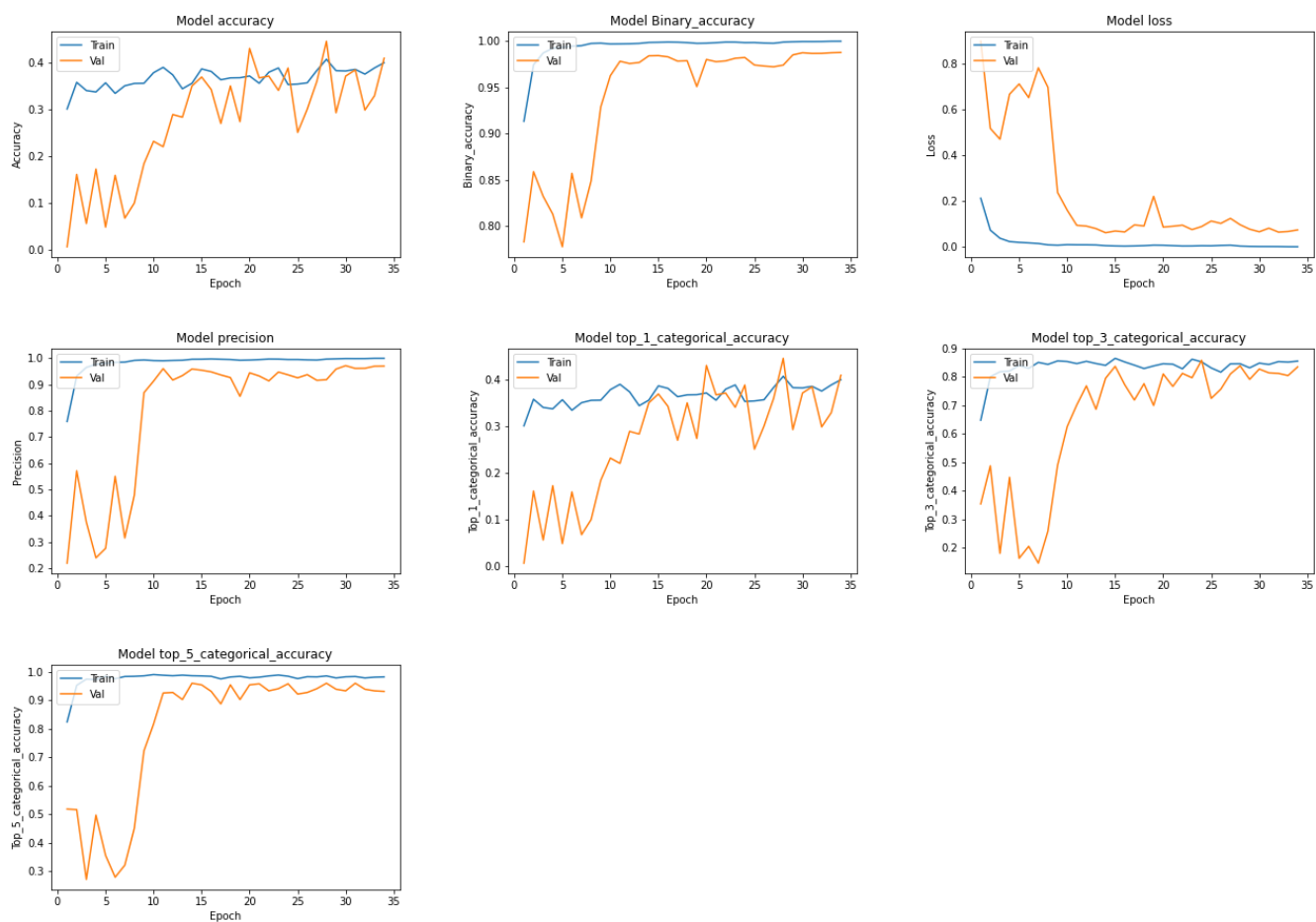


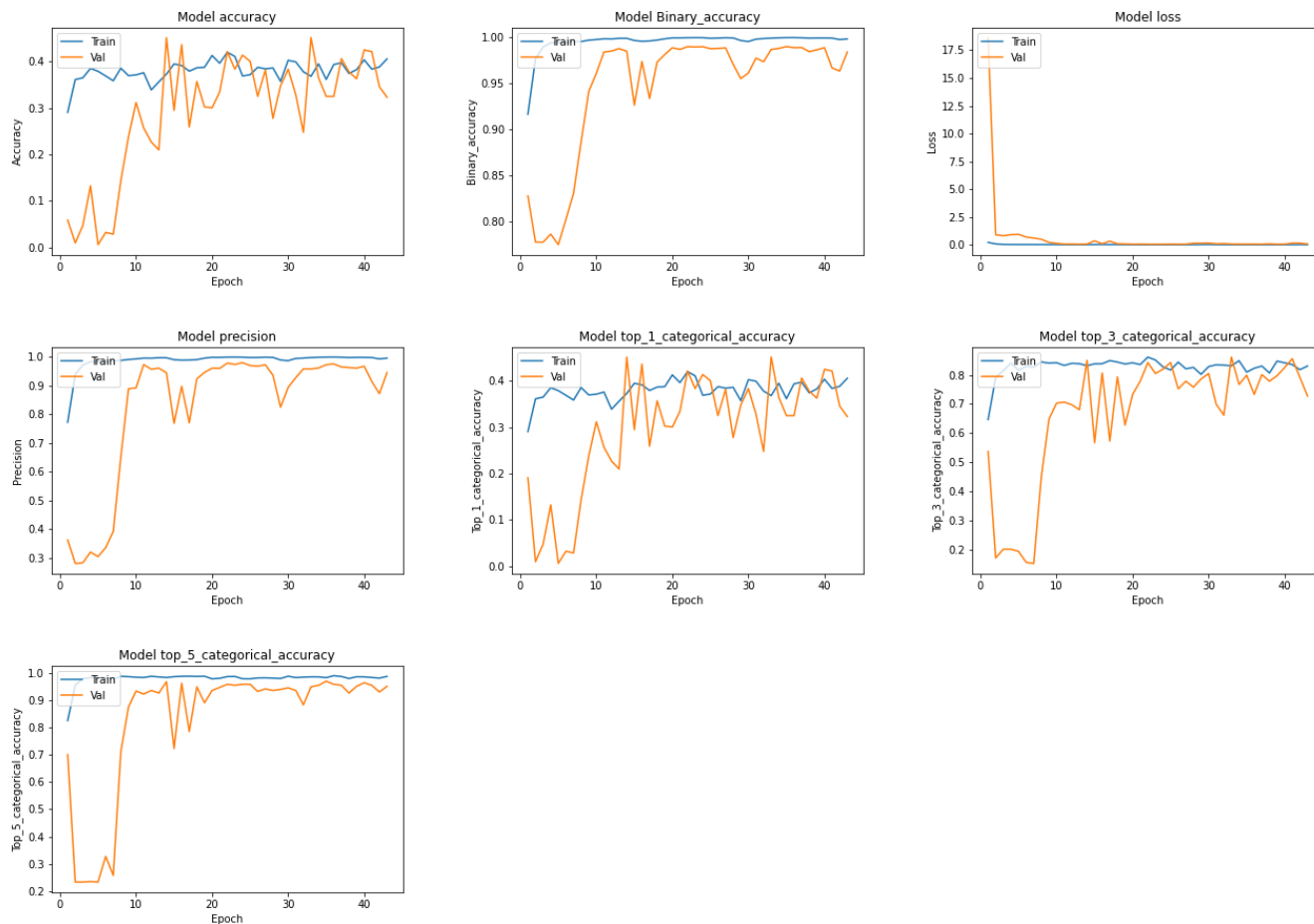












Roadmap per sviluppi futuri

I possibili miglioramenti di questa applicazione possono includere :

1. Definizione di una metrica più adatta , una specializzazione della categorical accuracy che conti la frequenza degli indici ad 1 della true rispetto agli indici dei valori più alti della predicted.
2. Estensione dei modelli agli anni 1920-1999
3. Previsione basata su decenni e/o periodi mobili , per esempio un range decennale [year-5,year+4]
4. Test con altri [modelli](#)
5. Deploy lato server con altri framework per il machine learning come ML.NET