Università degli studi di
Milano-Bicocca

Advanced Machine Learning

Final Project

NYC - Taxi Trip Duration

Authors:
Mattia Pennati - 793375 - m.pennati7@campus.unimib.it
Mirko Rima - 793435 - m.rima@campus.unimib.it
Francesco Rovera - 794194 - f.rovera1@campus.unimib.it

February 12, 2019

Contents

List of Figures

List of Tables

Abstract

In questo lavoro saranno presentati alcuni modelli predittivi in grado stimare il tempo di percorrenza di una corsa in taxi nella città di New York con la maggiore precisione possibile. Per questo scopo verranno effettuate diverse analisi, si utilizzeranno reti neurali e si sfrutteranno i vantaggi ottenibili dall'utilizzo di alcune tecniche di ottimizzazione. Si illustrerà infine il modello finale, ovvero il miglior predittore ottenuto per il problema di riferimento.

1 Introduction

New York si estende su un'area di circa 780 km^2, conta più di 8,5 milioni di abitanti e, considerando l'agglomerato metropolitano, risulta uno dei centri economici più importanti al mondo. Tali motivi rendono i taxi uno dei mezzi di trasporto preferiti da turisti e cittadini. Chi non vorrebbe conoscere in anticipo il tempo necessario a giungere a destinazione?
Si è ritenuto di grande interesse approfondire il problema della predizione del tempo di percorrenza di una corsa prendendo in considerazione le condizioni al momento del viaggio (clima, orario, giorno, ecc. . .).
L'obiettivo principale di questo elaborato sarà quindi l'analisi e l'ottimizzazione di diversi modelli con il fine di trovare il miglior predittore della variabile di interesse (la durata del viaggio). Tale variabile può essere influenzata da un elevatissimo numero di elementi (incidenti, condizioni atmosferiche avverse, traffico, ecc...). Dopo una fase di analisi si stimerà non solo un modello continuo che preveda l'esatto tempo di percorrenza, ma anche un modello discreto poiché ritenuto di maggiore utilità sia per il fornitore del servizio sia per chi ne usufruisce. Viene infatti considerato di maggiore interesse conoscere un range indicativo di tempo con un'alta affidabilità, piuttosto che il tempo espresso in secondi, ma con una probabile inaccuratezza dovuta ai molteplici fattori esposti precedentemente.

2 Datasets

Il dataset utilizzato è il file **train.csv** della seguente repository su Kaggle: nyc-taxi-trip-duration/data.
Il set di dati principale è stato pubblicato dalla NYC Taxi and Limousine

Commission e contiene i dati relativi a 1458644 corse in taxi nell'anno 2016 a Ney York City. Esso include i seguenti attributi:

Nome Attributo	Descrizione	Tipo
id	identificativo univoco per ogni viaggio	character
vendor_id	codice indicante il fornitore associato al record del singolo viaggio	integer
pickup_datetime	data e ora di attivazione del tassametro	double
dropoff_datetime	data e ora di disattivazione del tassametro	double
passenger_count	numero di passeggeri nel veicolo	integer
pickup_longitude	longitudine di attivazione del tassametro	double
pickup_latitude	latitudine di attivazione del tassametro	double
dropoff_longitude	longitudine di disattivazione del tassametro	double
dropoff_latitude	latitudine di disattivazione del tassametro	double
store_and_fwd_flag	flag indicante se il record di viaggio è stato salvato nella memoria del veicolo prima di inviarlo al fornitore. (Y/N)	character
trip_duration	durata del viaggio in secondi	integer

Table 1: Descrizione delle features del dataset train.csv

Dataset aggiuntivi

Avendo a disposizione un numero limitato di attributi si è pensato ad alcuni fattori che potessero influenzare il viaggio di un veicolo. Motivo per il quale si sono presi in considerazione il meteo, causa di successive complicazioni (diminuzione della velocità, traffico, ecc.) e le differenti zone della città dalle quali parte e alle quali giunge una determinata corsa in taxi.
Per questa ragione sono stati integrati altri due dataset, ovvero:

- **Weather:** dataset delle condizioni meteoreologiche nelle diverse ore nella città di NY nell'anno 2016 (nyc-hourly-weather-data/data).
- **Zone:** dataset sulla suddivisione in zone della città di New York (1:Manhattan, 2:Brooklyn, 3:Queens, 4:Bronx, 5:Staten Island) a partire dalle coordinate geografiche (NewYorkZonesCentroids).

Pre-processing dei dataset e Data Cleaning

Si elenca la procedura riassuntiva che mostra le fasi di preparazione del dataset finale e di data cleaning.

1. Rimozione delle features ritenute irrilevanti dai dataset e cambiamento del formato di alcune colonne da caratteri a valori discreti.
2. Suddivisione della colonna "data", espressa in data e ora, in due attributi separati (data e ora), scartando minuti e secondi.
3. Mantenimento di una sola istanza per ogni ora di ogni giorno nel dataset relativo alle condizioni climatiche
4. **Inner join** dei due dataset train.csv e Weather.csv, considerando come chiavi le features data e ora.
5. Conversione della data dai singoli giorni in:
   - dayweek: il giorno della settimana
   - holidays: (variabile booleana, 0 = giorno feriale e 1 = festivo)
   - month: il mese
6. Aggiunta della colonna "distance" (ricavata dalle latitudini e longitudini tramite la libreria "geopy.distance"). La stima della distanza risulterà però indicativa poichè calcolata come distanza in linea d'aria e non distanza stradale.
7. Rimozione delle colonne con troppi valori nulli ($> \frac{1}{3}$) e, successivamente, delle righe con valori nulli.
8. Rimozione dei numerosi outlier, probabilmente causati da errori di inserimento del tassista, corruzione dei dati o rumore nelle misure.
9. Fase di normalizzazione dei dati (Z-score normalization) per incrementare le performance delle tecniche di Machine Learning impiegate.

Analisi della correlazione

Successivamente alle modifiche del dataset si passa ad una rapida analisi della correlazione riportata in Fig. 1.
Come si può notare, la variabile target, trip_duration, risulta molto poco correlata con le altre features, fattore che limiterà le potenzialità dei modelli predittivi implicando la necessità di ulteriori dati.
Data la scarsa correlazione si è deciso di rimuovere ogni feature che risulti incorrelata (correlazione inferiore allo 0.025 in valore assoluto), ri-analizzando successivamente la correlazione a scopo illustrativo.
Si osserva che l'esecuzione di un modello predittivo senza la rimozione delle

features è stata provata con scarsi risultati sia a livello di misure di qualità che a livello di tempo di esecuzione.

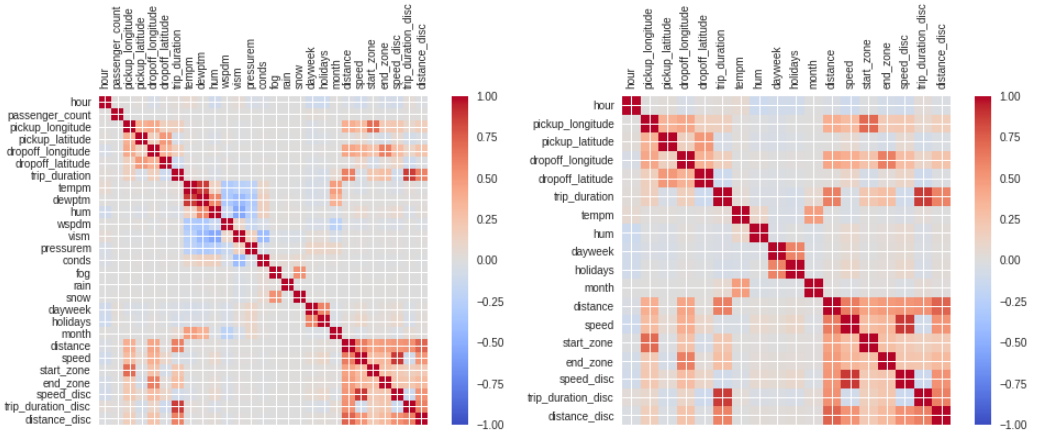

Figure 1: Visualizzazione della correlazione fra le features del dataset prima e dopo la rimozione degli attributi con correlazione irrilevante.

Analisi degli outlier

Alcuni valori del dataset sono stati considerati come outliers, ovvero misure impossibili da registrare, e si è proceduto con la loro rimozione. Ne sono un esempio alcuni tempi di percorrenza eccessivi (es. 40 giorni) o alcune coordinate (Fig. 2) relative a luoghi di partenza in Canada, in altri stati dell'America o, addirittura, nell'oceano (dato che ci riferiamo alla sola New York e alle sue zone limitrofe).

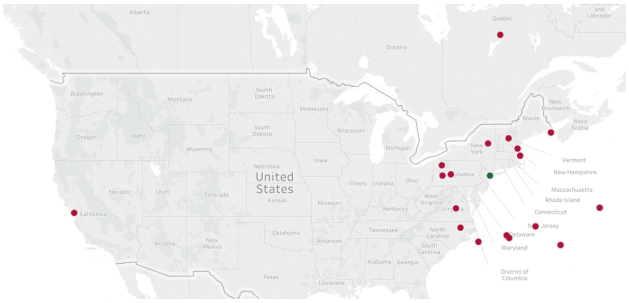

Figure 2: In verde NY, in rosso varie zone di partenza considerabili errate

Successivamente si sfrutterà lo Z-score per scalare e centrare i dati, distribuendoli con media nulla e deviazione standard unitaria, e identificare come ulteriori outliers quei punti troppo distanti da 0.
Si sono creati due modelli, che verrano illustrati dettagliatamente nella sezione seguente (Sezione 3), uno continuo e uno discreto.
Per la creazione di quest'ultimo si suddivideranno i tempi di percorrenza in 5 classi discrete ritenute semanticamente significative, ovvero:

- 0: Viaggio breve (minore di 5 minuti)
- 1: Viaggio medio-breve (tra 5 e 10 minuti)
- 2: Viaggio medio (tra 10 e 25 minuti)
- 3: Viaggio medio-lungo (tra 25 e 40 minuti)
- 4: Viaggio lungo (tra 40 minuti e 2 ore)

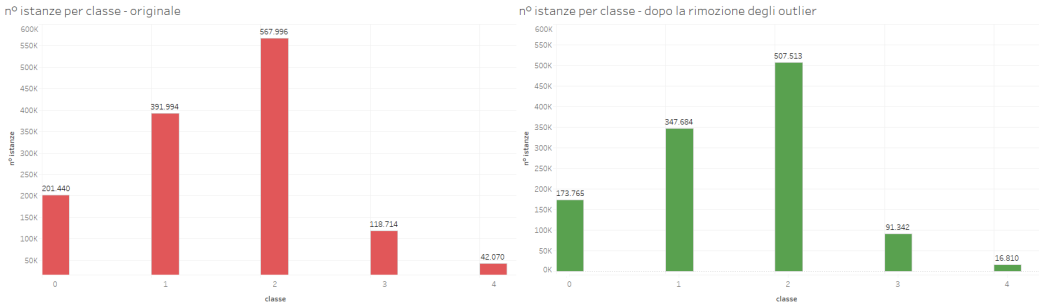

Figure 3: Valori divisi per classi (in ordine da sinistra verso destra , da 0 a 4) prima della rimozione degli outlier (rosso) e dopo la rimozione (verde)

Come si può notare in Fig. 3, la distrubuzione del numero di viaggi presenti nel dataset per ogni classe creata rimane pressochè invariata prima e dopo la rimozione deli outlier. Questo mostra chiaramente che i viaggi più frequenti sono quelli per distanze medie. Nonostante le classi ottenute non risultino bilanciate le si mantiene invariate per il valore informativo che posseggono.
Al termine della fase di rimozione degli outlier il train è passato da avere 1.458.644 istanze a 1.137.114, vedendosi quindi rimuovere, tramite il metodo z-score e analisi manuali, circa 320.000 istanze. Nonostante risultino una parte consistente del dataset si è ritenuto di avere comunque a disposizione un numero più che sufficiente di dati, preferendo quindi ignorare ogni istanza che potesse risultare ambigua.

3 The Methodological Approach

In accordo alle richieste del progetto si cercherà di sviluppare un regressore mantenendo la variabile target come continua (trip_duration). Osservando alcuni risultati della challenge su kaggle (nyc-taxi-trip-duration/kernels) si può però appurare come i risultati siano poco accurati (circa 0,30 come migliore RMSLE registrato). Per questo motivo viene ritenuta utile la creazione di un secondo modello, la cui variabile target rimarrà la stessa del precedente, ma verrà discretizzata. Tale scelta è motivata dal fatto che per un utente potrebbe risultare più utile conoscere con un'alta probabilità l'intervallo di tempo necessario a completare la corsa piuttosto che ottenere il minutaggio esatto, ma con un alto valore di inaccuratezza.
Queste considerazioni e il fatto che l'implementazione di una semplice rete neurale a scopo esplorativo abbia fornito performance deludenti hanno condotto alla decisione di arricchire il dataset originale con nuove informazioni e procedere con una fase di data cleaning. Per questo motivo sono state aggiunte, come illustrato dettagliatamente nella sezione precedente, informazioni relative al clima, alla geografia e alle festività.
Una volta completato il dataset di partenza si creano due semplici modelli che possano fungere come base di partenza per ulteriori future analisi. A seguire, in Tabella 2, viene riportata la struttura dei due modelli. Si osserva che per entrambi i modelli realizzati si hanno 2 livelli nascosti (non considerando il Dropout). Questo è dovuto sia al fatto che vengano ritenuti sufficienti per il numero non elevato di features a disposizione sia al fatto che l'esecuzione a scopo di confronto di una rete neurale più profonda incrementi il tempo di esecuzione senza migliorare le misure di qualità considerate.
Dato che i modelli in questione sono stati creati utilizzando paramentri scelti arbitrariamente si è deciso di procedere con una notevole fase di ottimizzazione degli hyper-parameters delle reti neurali implementate.
Come primo passo di ottimizzazione è stata eseguita una Grid Search sui parametri che non necessitavano di essere campionati o per i quali si sono scelti dei possibili valori alternativi, ovvero:

- loss function
- optimization algorithm
- batch size
- numero di neuroni per i due livelli nascosti

Ottenuti i risultati di questa prima fase di ottimizzazione se ne esegue un'ulteriore che mira però a migliorare l'optimization algorithm che ha portato ad ot-

	Modello discreto	Modello continuo
Act hidden layer	ReLu	ReLu
Act out layer	Softmax	Linear
Loss function	cat_crossentropy	mse
optimizer	adadelta	rmsprop
# neuroni lv 1	32	32
# neuroni lv 2	16	16
Epoche	20	20
Batch size	256	128
Dropout	0.2	-
Early stopping	val_loss	val_rmsle

Table 2: Configurazione modelli pre-ottimizzazione.

tenere lo score migliore e stabilire il numero di neuroni per ogni livello nascosto ricercando i valori in un intervallo ristretto centrato sul valore ottimo trovato tramite Grid Search.
Si impiega quindi un'ottimizzazione bayesiana mediante i Gaussian Process con l'obiettivo di stabilire i migliori valori per i seguenti parametri:

- learning rate
- numero di neuroni per i due livelli nascosti

Si osserva che la fase di ottimizzazione è stata svolta approssimativamente sul 10% del dataset per motivi legati al tempo di esecuzione richiesto. L'utilizzo di un dataset ridotto, le poche configurazioni testate e il tempo di esecuzione hanno rappresentato un grosso limite nella possibilità di ricercare approfonditamente il modello che meglio risolvesse il problema.
In conclusione vengono implementati e trainati, sia nel caso continuo che nel caso discreto, i modelli ottenuti dall'utilizzo dei migliori parametri ricavati in precedenza. Per queste reti neurali, la cui struttura è riportata in Tabella 3, verrà utilizzato un numero di epoche maggiore rispetto ai modelli precedenti e la fase di training sarà effettuata sull'intero dataset in modo da ottenere così i due migliori modelli finali. (tempi di esecuzione in Tab. 4)
Si eseguiranno gli stessi modelli anche sulla versione originale del dataset in modo da poter confrontare i risultati. Per quest'ultima prova si suddividerà comunque la feature data nelle features dayweek, month e hour e si eseguirà comunque la fase di data cleaning.

	Modello discreto	Modello continuo
Act hidden layer	ReLu	ReLu
Act out layer	Softmax	Linear
Loss function	cat_crossentropy	mae
optimizer	Adadelta(lr = 1.050)	Adam (lr = 0.01)
# neuroni lv 1	75	54
# neuroni lv 2	10	20
Epoche	30	30
Batch size	128	128
Dropout	2 livelli da 0.2	1 livello da 0.2
Early stopping	val_loss	val_rmsle

Table 3: Configurazione modelli post-ottimizzazione.

Modello	# epoche	Tempo di Training
discreto(dataset esteso)	23 (early stopping)	13 min e 46 s
continuo(dataset esteso)	14 (early stopping)	16 min e 5 s
discreto(dataset originale)	17 (early stopping)	5 min e 32 s
continuo(dataset originale)	11 (early stopping)	3 m e 50 s

Table 4: Tempi di esecuzione della fase di training per i modelli finali.

4 Results and Evaluation

In questa sezione verranno presentati i principali risultati ottenuti sia per il regressore, il modello con la variabile target continua, sia per il classificatore, ovvero il modello con la variabile target discretizzata. Si illustrera anche il risultato dell'esecuzione degli stessi modelli su una versione del dataset non estesa (sul solo train.csv), al quale si esegue comunque la fase di data cleaning e si suddivide la data in dayweek, hour e month.

Regressore

A seguire saranno riportati i risultati del modello continuo sia per il dataset originale che per il dataset esteso e ripulito (dopo la fase di data cleaning, la rimozione delle features non correlate e l'ottimizzazione).

Dataset originale

Si riportano i risultati dell'esecuzione del modello finale sul dataset originale.

Train		Validation	
Loss (MAE)	RMSLE	Loss (MAE)	RMSLE
194	0.28	181	0.27

Table 5: Loss e RMSLE per il modello finale addestrato sul dataset originale.

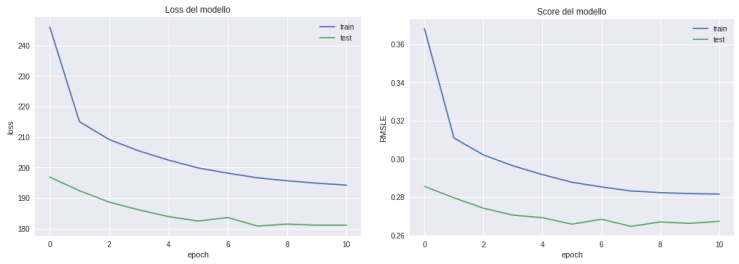

Figure 4: Tendenze per Loss function e RMLSE al crescere del numero di epoche (dataset originale).

Dataset esteso

Si riportano i risultati dell'esecuzione del modello finale sul dataset esteso.

Train		Validation	
Loss (MAE)	RMSLE	Loss (MAE)	RMSLE
189	0.27	177	0.26

Table 6: Loss e RMSLE per il modello finale addestrato sul dataset esteso.

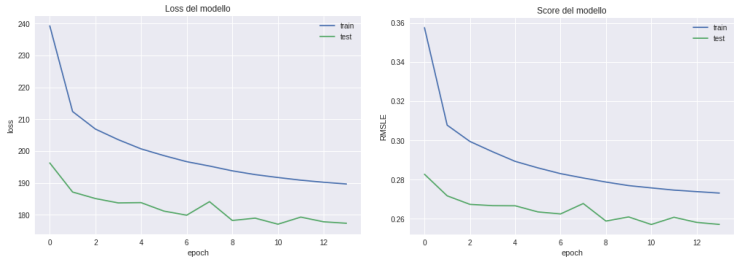

Figure 5: Tendenze per Loss function e RMLSE al crescere del numero di epoche (dataset esteso).

Classificatore

A seguire sono riportati i risultati del modello discreto sia per il dataset originale che per il dataset esteso.

Dataset originale

Si riportano i risultati dell'esecuzione del modello finale sul dataset originale.

Train		Validation	
Loss (cat. crossentropy)	Accuracy	Loss (cat. crossentropy)	Accuracy
0.77	67%	0.72	69%

Table 7: Loss e accuracy per il modello finale addestrato sul dataset originale.

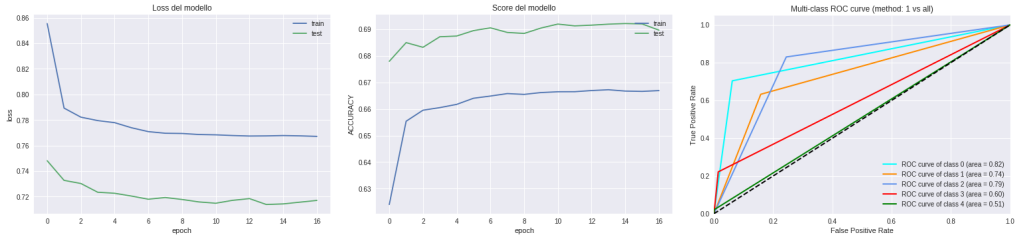

Figure 6: Nelle due immagini a sinistra tendenze per Loss function e Accuracy al crescere del numero di epoche, a destra curva ROC in versione "1 vs all" multiclass (dataset originale).

Dataset esteso

Si riportano i risultati dell'esecuzione del modello finale sul dataset esteso.

Train		Validation	
Loss (cat. crossentropy)	Accuracy	Loss (cat. crossentropy)	Accuracy
0,75	68%	0,70	70%

Table 8: Loss e accuracy per il modello finale addestrato sul dataset esteso.

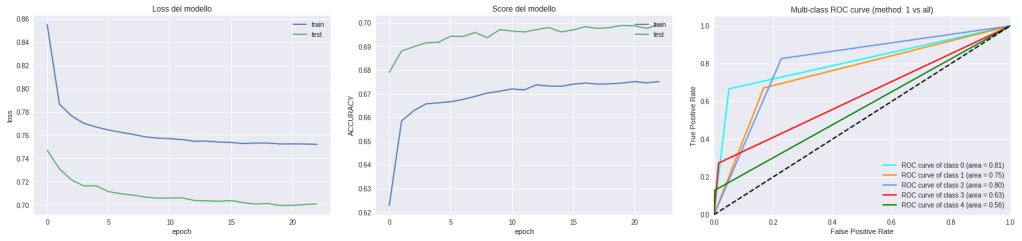

Figure 7: Nelle due immagini a sinistra tendenze per Loss function e Accuracy al crescere del numero di epoche, a destra curva ROC in versione "1 vs all" multiclass (dataset esteso)

5 Discussion

Il dataset di riferimento, come testimoniato dalla competition su Kaggle, presenta molteplici difficoltà che impediscono, momentaneamente, di ottenere un predittore che abbia un'elevata affidabilità. Le motivazioni sono da ricercare nella poche features presenti e nella loro bassa correlazione con la variabile target (la durata di una corsa in taxi).
Conoscendo in anticipo queste problematiche si è deciso di sfruttare una fase di ottimizzazione e di provare ad estendere il dataset originale con alcune informazioni ritenute importanti come la distanza in linea d'aria tra luogo di partenza e di arrivo, le condizioni climatiche e il quartiere di partenza e quello di destinazione. Inoltre si è eseguita una fase di pulizia dei dati, rimuovendo possibili misure errate e ogni valore considerabile outlier.
Nonostante gli sforzi e nonostante l'ottimizzazione porti a una miglioria delle metriche qualitative analizzate non si ottengono i risultati sperati tramite

l'estensione del dataset. L'inserimento di nuove features porta un vantaggio minimo in termini di qualità delle predizioni, ma incrementa notevolmente il tempo di esecuzione, rendendo attualmente sfavorevole il suo utilizzo.
Si sottolinea però come questo lavoro rappresenti solo un'iniziale approfondimento del problema di interesse e fornisca comunque spunti per ulteriori modifiche. Si potrebbe, ad esempio, estendere ulteriormente il dataset ricercando informazioni che possano avere una correlazione maggiore con la variabile target come un indice di traffico per ogni zona, la distanza stradale anzichè la distanza in linea d'aria, il numero di semafori lungo il percorso, ecc...
Durante lo svolgimento di questo lavoro vengono comunque mostrati due modelli finali con misure di qualità migliori dei principali modelli sviluppati durante la challenge su Kaggle. Questo dimostra come la pulizia dei dati, l'utilizzo di reti neurali e una fase di ottimizzazione permettano di incrementare le performance e rappresentino un'ottima base di partenza per approcciarsi alla predizione del tempo di percorrenza di una corsa in taxi nella città di New York. Si considera quindi utile, oltre all'aggiunta di informazioni maggiormente rilevanti, una fase di ottimizzazione più massiccia che però può essere permessa solo da una potenza computazionale decisamente maggiore rispetto quella a disposizione durante lo svolgimento del lavoro.
Si osserva inoltre che, per quanto riguarda il modello discreto, i risultati delle curve ROC mostrino come la classe numero 4 (viaggi dai 40 ai 120 minuti) venga predetta decisamente peggio delle altre classi. Questo avviene per la sua numerosità, troppo bassa rispetto alle altre, e suggerisce una suddivisione delle classi maggiormente bilanciata. Un fattore analogo, anche se meno evidente, può essere osservato anche per la classe numero 3 (dai 25 ai 40 minuti), in quanto anch'essa risulta sbilanciata rispetto alle prime 3 classi (la 0, la 1 e la 2). Una rapida prova accorpando le classi 3 e 4 mostra un leggero miglioramento (prestazioni circa il 2% migliori), ma diminuisce il significato delle singole classi, non motivandone quindi l'implementazione.

6 Conclusions

Alla fine della nostra analisi possiamo concludere che:

- si ottengono due modelli finali in grado di eseguire predizioni con performance migliori delle principali consegne durante la challenge su Kaggle, ma, nonostante ciò, la qualità ottenuta non risulta ancora sufficiente per un reale utilizzo.

- l'estensione del dataset non fornisce, attualmente, miglioramenti notevoli che ne motivino l'utilizzo, limitandosi quindi a incrementare esponenzialmente il tempo di esecuzione della fase di training dei modelli.
- la versione finale del classificatore permette di ottenere predizioni discretamente affidabili, rendendo i risultati più interpretabili e, forse, più utilizzabili.
- la fase di ottimizzazione, che richiede uno sforzo computazionale notevole, è risultata utile soprattutto per quanto riguarda la scelta di di diversi hyper-parameters (ottimizzatore, numero di neuroni, learning rate, ...) ma le poche configurazioni testate rapportate all'elevato costo computazionale e alla bassa capacità computazionale a disposizione rappresentano un notevole limite.
- il lavoro svolto permette comunque di ottenere un'idea iniziale di come approcciarsi a problemi simili, problemi che negli ultimi anni risultano sempre più studiati a causa della quantità sempre maggiore di dati disponibili.

Sviluppi futuri

Riepilongando quanto già accennato nella sezione precedente, al fine di poter ottenere performance migliori, si potrebbe agire nei seguenti modi:

- arricchire il dataset tramite ulteriori informazioni fortemente correlate con la variabile target (trip_duration) come, per esempio:
  - informazioni relative al traffico inerenti al tragitto di ciascuna corsa relazionata all'orario;
  - informazioni riguardanti un indice della quantità di traffico per ogni zona (quartiere) della città;
  - distanze stradali espresse in Km tra punto di partenza e arrivo, anzichè distanze in linea d'aria;
  - numero di semafori (o incroci) lungo il tragitto;
  - dati relativi a ulteriori annate.
- suddividere ulteriormente le zone, non limitandosi solamente alla distinzione sulla base dei cinque quartieri principali;
- migliorare la fase di ottimizzazione mediante l'utilizzo di macchine consone a questo tipo di sforzo computazionale e che permettano di esplorare una maggior quantità di parametri e di configurazioni di parametri.