Università degli Studi di Padova
Dipartimento di Filosofia, Sociologia, Pedagogia e Psicologia Applicata
Corso di Laurea Magistrale in

Psicologia Sociale, del Lavoro e della Comunicazione

Tesi di Laurea

Analisi esplorativa di dati emotivi su short text provenienti dai social media mediante sentiment analysis

Relatore Prof. Antonio Calcagnì
Dipartimento di Scienze Psicologiche dello Sviluppo e della Socializzazione

Laureanda Elisabetta Genovese
Matricola N 1206657

Anno Accademico 2020/2021

Indice

Introduzione

Il presente lavoro nasce dalla curiosità di indagare l'espressione dell'interazione sociale in un momento storico in cui l'unica modalità concessa di interagire socialmente era contactless: i social media durante la pandemia Covid-19, in particolare durante il primo lockdown in Italia tra Marzo e Aprile del 2020, in cui l'isolamento sociale era particolarmente rigoroso.
L'interazione sociale è un buon campo d'azione per esprimere emozioni e, considerato l'uso dei social media, l'idea era quella di esplorare la quantità di emozione espressa e le covariate che riguardassero la variazione di tale contenuto emotivo espresso.

Il lavoro di tesi si presenta dunque come un'analisi esplorativa di un dataset di dati testuali provenienti dai Social Media (Facebook e Instagram) raccolti tra Marzo e Aprile 2020, durante il primo lockdown.
L'elaborato si struttura in una prima presentazione del background teorico, del metodo per estrarre i dati emotivi e della scelta della variabile dipendente da considerare come espressione della quantità dei dati emotivi.In ultimo, l'utilizzo del modello statistico per l'analisi e la sua applicazione su R.

Capitolo 1

Teoria e metodi

Il primo approccio quantitativo all'analisi dei contenuti testuali è stato svolto dall'editore del New York World, John Speed, curioso di sapere su quali temi stesse convergendo il giornalismo nel 1893 (Kennedy et al. 2021). Il presente lavoro si basa sull'utilizzo di un particolare strumento informatico per svolgere sentiment analysis, VADER, per estrarre la quantità di contenuto emotivo espresso sui social media tramite testo scritto, ma prima di parlare specificamente di quest'ultimo è necessario preparare un cappello introduttivo sulla teoria del Natural Language Processing, la Sentiment Analysis e il pacchetto NLTK di Python.

1.1 NLP e Sentiment Analysis

La definizione del Natural Language Processing (NLP) è data da Liddy (Liddy 2001) come "una gamma teoricamente giustificata di tecniche computazionali per analizzare e rappresentare testi naturali a uno o più livelli di analisi linguistica allo scopo di ottenere un'elaborazione del linguaggio simile a quella umana per una serie di compiti o applicazioni", dunque è un processo di trattamento automatico di dati testuali mediante calcolatore elettronico.
Il linguaggio umano, tuttavia, ha delle proprietà intrinseche di ambiguità (es. "essere" inteso come verbo e "essere" come sostantivo, tipo *essere umano*) che rendono particolarmente difficile e complesso il calcolo computazionale dello

stesso, per tale ragione viene suddiviso in fasi diverse: dall'analisi lessicale per la scomposizione in token (es. le parole) a quella semantica (assegnazione di significati), passando l'analisi grammaticale e per l'analisi sintattica fino ad arrivare all'espressione linguistica.

Scomporre il testo, quindi, in dati strutturati è la base per poter compiere operazioni sui dati testuali.

L'operazione di estrazione di significato su cui il presente lavoro è basato è quello di estrazione di contenuto emotivo mediante Sentiment Analysis.

La Sentiment Analysis (SA) è un campo di applicazione utilizzato per estrarre il *sentiment* dal testo, con valenza positiva, negativa ed eventualmente neutra.
Lo stato dell'arte della Sentiment Analysis consta di diverse metodologie di estrazione e catalogazione del sentiment nei testi, dal *Machine Learning, LIWC, ANEW, the General Inquirer, SentiWordNet, il Support Vector Machine (SVM) e VADER* (Hutto e Gilbert 2014).
Solitamente la SA - anche detta *opinion mining* - è utilizzata per estrarre opinioni, emozioni e atteggiamenti nei confronti di un argomento o un prodotto in un determinato arco temporale. In questo caso non è presente uno specifico argomento (ad esempio un hashtag o un prodotto da recensire), ma i dati raccolti si distribuiscono nell'arco temporale del primo lockdown di Marzo e Aprile 2020 con osservazioni ripetute per gli stessi soggetti.

Per il presente elaborato, dopo una serie di prove esplorative, è stato scelto l'utilizzo del framework VADER (dal pacchetto `nltk`) in quanto è di facile applicazione, ha una buona efficacia confrontato sia con gli altri strumenti sopra citati che con confronti con classificazioni fatte da esseri umani (Hutto e Gilbert 2014), inoltre come output dà una variabile in grado di distinguere il sentiment rilevato in negativo, neutro e positivo, aspetto cruciale che ci ha permesso di comprendere la presenza o l'assenza di contenuto emotivo.

1.2 Suite NLTK e framework VADER

NLTK è una suite di programmi scritti in linguaggio di programmazione Python basata sul NLP. I programmi processano i dati del linguaggio umano attraverso una serie librerie di elaborazione del testo per la classificazione, tokenizzazione, stemming, tagging, parsing e ragionamento semantico, (*NLTK.org*).

Lo strumento di interesse per questo studio è il framework VADER (Valence Aware Dictionary for sEntiment Reasoning), appartenente al pacchetto `nltk`.
VADER è un programma lexicon based, ciò significa che per ogni parola presente nel suo dizionario, assegna sia valori di *polarity*, ovvero colloca la parola nel continumm della valenza negatività-positività, sia valori di *intensity*, quindi di quantità/intensità della valenza. VADER è uno strumento sentence-level, ovvero può essere applicato direttamente al testo senza necessariamente fare un lavoro di preprocessamento del testo (tokenizzare, lemmatizzare, etc.).
Essendo lexicon-based, sono presenti diversi limiti, tra cui la mancata contestualizzazione della parola nella frase, tuttavia se si riesce ad avere un gran numero di osservazioni, sfrutta la Wisdom of the Crowd (WotC), un approccio che si basa sull'idea che la conoscenza collettiva di un gruppo di persone, espressa attraverso le loro opinioni aggregate, possa essere affidabile come alternativa alla conoscenza degli esperti. Questo aiuta ad acquisire una stima valida per il punteggio di valenza del sentiment di ogni testo togliendolo dal contesto. Kennedy et al. 2021.
Un altro modo per ovviare alla mancata contestualizzazione della parola sono le 5 euristiche di VADER (Hutto e Gilbert 2014):

- La punteggiatura, per esempio "!", aumenta la grandezza dell'intensità senza modificare l'orientamento semantico. Per esempio: "Il gelato è freddo!!!" è più intenso di "Il gelato è freddo".

- La capitalizzazione, in particolare l'uso di ALL-CAPS per enfatizzare una parola rilevante per il sentiment in presenza di altre parole non

capitalizzate, aumenta l'intensità del sentiment senza modificare l'orientamento semantico. Per esempio: "Il gelato è FREDDO" trasmette più intensità di "Il gelato è freddo".

- Gli intensificatori (come gli avverbi) influenzano l'intensità del sentiment aumentando o diminuendo l'intensità. Per esempio: "il gelato è estremamente freddo." è più intenso di "Il gelato è freddo", mentre "Il gelato è leggermente freddo" ne riduce l'intensità.

- Spostamenti di polarità dovuti alle congiunzioni. La congiunzione avversativa "ma" segnala uno spostamento di polarità del sentiment, con il sentiment del testo che segue il periodo dominante. Per esempio: "Il gelato è freddo, ma è sopportabile." ha un sentiment misto, con il secondo periodo che detta la valutazione complessiva. Catturando la negazione della polarità, esaminando la sequenza contigua di 3 elementi che precedono una caratteristica lessicale carica di sentiment, catturiamo quasi il 90% dei casi in cui la negazione inverte la polarità del testo. Per esempio, una frase negativa sarebbe "Il gelato non è poi così freddo".

Si aggiunge, inoltre, una sesta euristica di VADER se si utilizza il pacchetto `Translator` con la funzione `emoji.translate` in grado di tradurre in testo le emoji presenti e farle "leggere" a VADER, caratteristica molto utile in questo particolare tipo di dati tratto dai social media.

In quanto strumento sentence level (significa che in input non vuole una lista di token ma una sentence), gli output per ogni short text (o sentence) in uscita sono quattro (1.2):

E' importante sottolineare che i valori degli output negative, neutral e positive non corrispondono al conteggio di parole associate all'etichetta, ma ad un calcolo matematico tra l'intensity e la polarity della parola.

Sulla variabile `compound` si parlerà più approfonditamente nelle prossime sezioni, per il momento è necessario capire che è una variabile normalizzata.

Label	Descrizione
negative	valore continuo da 0 a 1 di parole neutrali
neutral	valore continuo da 0 a 1 di parole neutrali
positive	valore continuo da 0 a 1 di parole positive
compound	valore continuo da -1 a +1 che indica la polarità della sentence

Tabella 1.1: Output di VADER

Variabili iniziali	Descrizione
ID testo	Codice identificativo di ogni short text
ID Autore	Codice identificativo di ogni autore
Testo	Testo grezzo estratto in lingua originale
Fonte	Social media di provenienza
Data	Data e orario di stampa del tweet
Timestamp	Stampa del Timestamp
Reddito	fascia di reddito del partecipante
DPCM	DPCM in vigore durante la stampa dello short text
Età	età di ogni partecipante
Genere	genere di ogni partecipante

Tabella 1.2: Dataset iniziale

1.3 Creare variabili: dal dataset iniziale a quello finale

Il dataset iniziale era strutturato in 10 colonne che fungevano da variabili e 38.000 righe, ognuna delle quali comprendeva uno short text diverso nella colonna `Testo`. I dati consistevano in una raccolta di dati testuali (commenti, direct, caption) provenienti da due tipologie di Social Media (Facebook e Instagram).
Il primo passo è stato quello di tradurre in inglese ogni riga della colonna `Testo` attraverso l'API di Google Translate, dato che il dizionario di VADER riconosce solo la lingua inglese.

Per poter effettuare una sentiment analysis che potesse creare una variabile di risposta adattabile al modello statistico, si è reso necessario manipolare i dati per creare un nuovo dataset.
L'ambiente di lavoro utilizzato è stato **Python** e le librerie `pandas` e `pyreadr`.
E' stata creata una classe chiamata `Tweet` in cui, fissati i DPCM, sono stati uniti tutti gli short text di ogni ID Autore in un unico corpus, perciò per ogni DPCM era presente per ogni autore un corpus di short text. In questa maniera svolgere la sentiment analysis ha avuto un focus *Document level-scope* per farsi un'idea generale del sentiment durante il DPCM.
La variabile su cui VADER ha lavorato è stata quella del testo, installando il pacchetto `nltk` e importando la libreria `nltk.sentiment.vader` e utilizzando la funzione `SentimentIntensityAnalyzer`.

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
# FUNZIONE PER CALCOLARE POLARITY SCORE PER OGNI FRASE
def nltk_sentiment(sentence):
    nltk_sentiment = SentimentIntensityAnalyzer()
    score = nltk_sentiment.polarity_scores(sentence)
    return score
# FINE FUNZIONE
```

Dopodichè per ogni short text è stata chiamata la funzione `nltk-sentiment(sentence)` che ha riportato i punteggi delle label mostrate nella 1.1.

Per concludere, nel dataset ultimato sono state scelte soltanto alcune variabili utili al fine di poter effettuare un'analisi esplorativa:

Per visualizzare tutto il codice commentato per la manipolazione dei dati e lo svolgimento della sentiment analysis tramite VADER, vedere Appendice A.

ID Autore	Codice identificativo di ogni autore
Testo	Corpus di short text per ogni autore in inglese
Genere	Genere di ogni partecipante
DPCM	DPCM in vigore durante la stampa dello short text
Reddito	Fascia di reddito del partecipante
Compound	Valore compreso da -1 a +1 che indica la polarità del sentiment

Tabella 1.3: Nuovo Dataset

Capitolo 2

Modello Binomiale e VADER compound

2.1 Modello Binomiale e GLM a Effetti Misti

I Modelli Misti Lineari Generalizzati (GLMM) si presentano come un'estensione dei modelli lineari generalizzati (GLM). La caratteristica principaleè la presenza, oltre agli effetti misti, degli effetti random o casuali. I Modelli Lineari Generalizzati sono considerati, quindi, una generalizzazione dei modelli lineari, dove però i dati appartenenti all'analisi in esame possono avere anche una distribuzione non normale dell'errore (Salvan, Sartori e Pace 2020).
I GLMM consentono quindi l'analisi di variabili di risposta che presentano, invece che distribuzioni normali semplici, distribuzioni di tipo arbitrario ed una funzione arbitraria, detta anche funzione di collegamento, della variabile detta "variabile di risposta". Operano, quindi, variando in modo lineare attraverso i cosiddetti predittori, bypassando l'ipotesi che sia la risposta stessa a dover variare in modo lineare (*Wikipedia, Modello Lineare* 2013).

La scelta di utilizzare il modello a effetti misti GLMM piuttosto che il semplice a effetti fissi GLM deriva dalla struttura dei dati del caso studio in esame, che presentano misure ripetute. Infatti, un utilizzo di modelli a effet-

ti fissi avrebbe fatto perdere la parte di variabilità data dalle misure ripetute.

Esiste una ampia disponibilità di distribuzioni messe a disposizione dai GLMM, per poter effettuare regressioni su dataset. Per gli obiettivi di questo lavoro di tesi, il modello che verrà trattato sarà quello a dati binari: il modello binomiale e la sua funzione di collegamento per la regressione.
Il modello binomiale è un modello a dati binari che si rivela uno strumento utile per effettuare analisi in scenari in cui la variabile di risposta è dicotomica. Alcuni esempi applicativi sono: segnalare presenza/assenza, successo/insuccesso, funzionante/guasto (Salvan, Sartori e Pace 2020).
E' possibile identificare due gruppi principali in cui sono suddivisi i modelli a dati binari: modelli a dati raggruppati e modelli a dati non raggrupati. Per il primo gruppo, il modello statistico binomiale non sarà un modello binomiale elementare $Bi(1, \pi_i)$, ma una distribuzione binomiale $Bi(m_i, \pi_i)$ (Salvan, Sartori e Pace 2020). Tuttavia, se si tengono in considerazione variabili casuali $Y_i, ...Y_n$, in entrambi i casi si potrà utilizzare la seguente formula:

$$m_i Y_i \sim Bi(m_i, \pi_i) \tag{2.1}$$

La funzione di legame per i dati binari su cui questa sessione sarà focalizzata è la funzione di legame logistica, oppure logit, il cosiddetto legame canonico. La formula carattersitica si presenta come segue:

$$g(\mu_i) = log(\mu_i/1 - \mu_i) = x_i\beta \tag{2.2}$$

da cui si ricava il *modello di regressione logistica* o *modello logit*, la cui funzione è $F(z) = e^z/(1 + e^z)$.

2.2 Modello Binomiale e la variabile Compound in VADER

Come mostrato nel capitolo precedente, la variabile `compound`, che dà in output VADER, varia in un range [-1:+1]. Per adattarla al modello binomiale,

Compound prima	Compound dopo
$x \geq 0.05$	1
$x \leq -0.05$	1
$-0.05 > x > 0.05$	0

Tabella 2.1: Compound Binomiale

è stato scelto, seguendo le direttive del codice della libreria nltk (*NLTK.org*), di rendere la variabile binomiale nel seguente modo:

E' stata adoperata questa scelta per misurare assenza/presenza di contenuto emotivo in cui positivo e negativo diventano 1 e il neutro diventa 0.
Nell'applicazione del modello sul software statistico R, mostrato nella sezione seguente, la nuova variabile `compound` verrà chiamata `emotion-cat` per comodità.

2.3 Applicazione del modello su R

Il modello binomiale è stato applicato sul software statistico R tramite il pacchetto glmer che deriva dal pacchetto lme4. Consiste nell'applicazione dei Modelli Misti Lineari Generalizzati e famiglia binomiale.

Per farlo è stato preparato il prima il dataset dato in uscita da Python:

```
library(readxl)
dataset_ultimo <- read_excel("C:/Users/elisa/Desktop/Tesi ordinata/dataset
    _ultimo.xlsx")
View(dataset_ultimo)
datax = dataset_ultimo[,-c(1,3,4)]
```

stabilendo come nuovo dataset su cui svolgere le operazioni il dataset `datax`.

Una volta fatta questa operazione, sono state assegnate le variabili come numeriche o come categoriali attraverso i comandi `as.factor` e `as.numeric`

affinchè il modello potesse leggere i dati:

```
datax$dpcm_num = as.numeric(datax$dpcm)
datax$Genere_cat = as.factor(datax$Genere)
datax$Genere_cat = factor(x = datax$Genere_cat,)
datax$Reddito = factor(x = datax$Reddito,labels = c("10–15","15–26","26–55
    ","55–75","75–120","–10","120–"))
datax$Reddito = relevel(datax$Reddito,ref = "–10")
datax$TitoloStudio = as.factor(x = datax$TitoloStudio)
colnames(datax)[c(2,19)] = c("ID","emotion")
datax$emotion_cat = as.factor(datax$emotion)
```

La nostra variabile dipendente, ovvero il compound, è diventato una variabile categoriale che è stata chiamata `emotion-cat`, le altre variabili categoriali sono: `Genere` e `Reddito`, mentre come variabili numeriche `dpcm`.
La riga `colnames(datax)[c(2,19)] = c("ID","emotion")` è servita per rinominare le colonne che inizialmente erano `Author` e `compound`.

Una prima analisi esplorativa è stata fatta plottando le covariate DPCM, Reddito e Genere con la variabile dipendente "emotion-cat", mostrati in Figure 2.1, 2.2, e 2.3. Risulta chiara una presenza di emotività in funzione di DPCM, Reddito e Genere. E' stata quindi approfondita l'analisi statistica, con i metodi descritti nelle seguenti sezioni.

2.3.1 Utilizzo del pacchetto GLMER

Per affrontare l'analisi dati discussa, si è adottato il pacchetto GLMER, della library `lme4` di R. La library `lme4` mette a disposizione diversi modelli e fornisce funzioni per l'adattamento e l'analisi di modelli misti: lineare `lmer`, lineare generalizzato `glmer` e non lineare `nlmer`(Bates et al. 2014). Questa library utilizza metodi di algebra lineare moderni ed efficienti come implementati nel pacchetto `Eigen` e utilizza classi di riferimento per evitare la copia eccessiva di oggetti di grandi dimensioni; è quindi una più ampia probabità di risultare sia più veloce che più efficiente, in termini di memoria, rispetto ad altri approcci, come `nlme`, ovvero i modelli a effetti misti non lineari, che

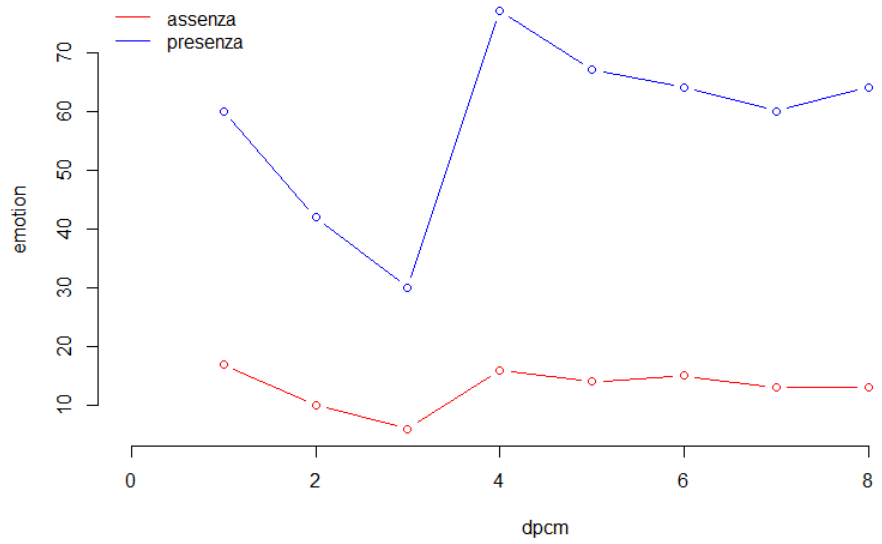

Figura 2.1: Plot "emotion-cat" vs DPCM

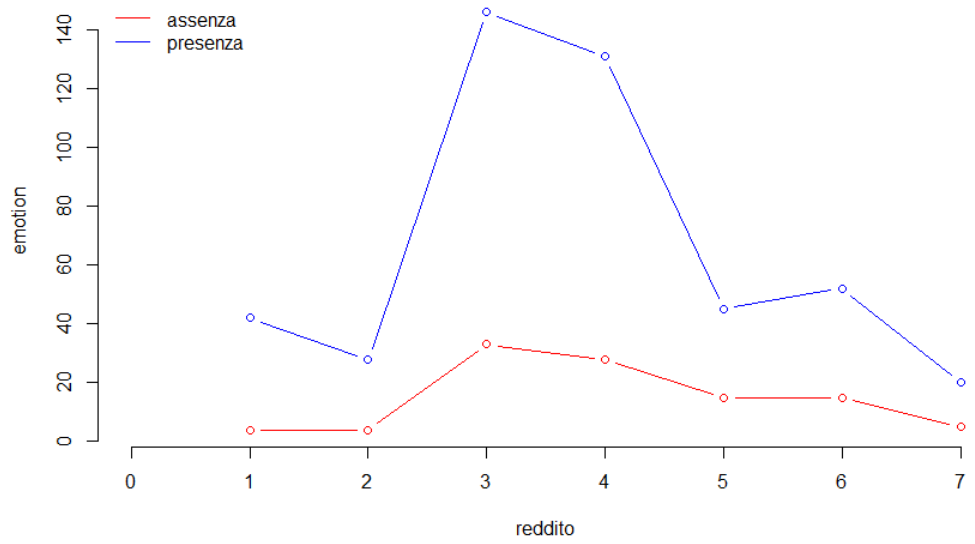

Figura 2.2: Plot "emotion-cat" vs Reddito

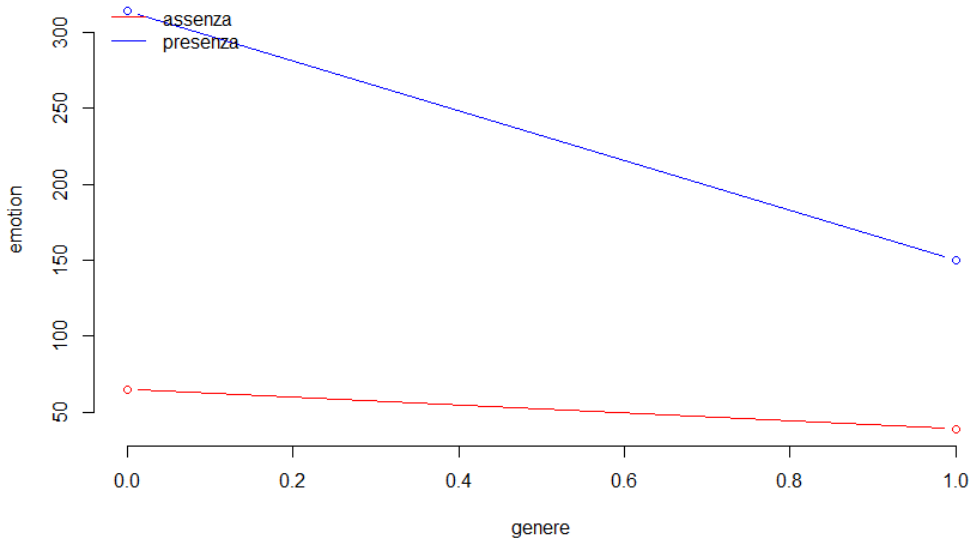

Figura 2.3: Plot "emotion-cat" vs Genere

invece adattano la funzione generica a un modello a effetti misti non lineare nella formulazione descritta in Lindstrom e Bates (Lindstrom e Bates 1990), con effetti casuali annidati. All'interno della library `lme4`, la funzione `glmer` offre un adattamento di modelli a effetti misti lineari generalizzati (GLMM), dove sia gli effetti fissi che gli effetti casuali sono specificati tramite la formula del modello. La struttura default della funzione si presenta come segue:

```
glmer(formula, data = NULL, family = gaussian
    , control = glmerControl()
    , start = NULL
    , verbose = 0L
    , nAGQ = 1L
    , subset, weights, na.action, offset, contrasts = NULL
    , mustart, etastart
    , devFunOnly = FALSE)
```

Nello specifico, `formula` è un oggetto formula lineare a due lati che descrive sia la parte a effetti fissi che a effetti casuali del modello, con la risposta a sinistra di un operatore e i termini, separati da operatori +, a destra. I termini a

effetti casuali sono contraddistinti da barre verticali ("|") che separano le espressioni per le matrici di progettazione dai fattori di raggruppamento. Per l'analisi oggetto di discussione, sono state investigate diverse tipologie di `formula`, come qui di seguito descritto:

```
mod0 = lme4::glmer(formula = emotion_cat~1+(1|ID),data=datax,family=
    binomial)
mod1 = lme4::glmer(formula = emotion_cat~dpcm_num+(1|ID),data=datax,family
    =binomial)
mod2 = lme4::glmer(formula = emotion_cat~Reddito+(1|ID),data=datax,family=
    binomial)
mod3 = lme4::glmer(formula = emotion_cat~dpcm_num+Reddito+(1|ID),data=
    datax,family=binomial)
mod4 = lme4::glmer(formula = emotion_cat~dpcm_num*Reddito+(1|ID),data=
    datax,family=binomial)
mod5 = lme4::glmer(formula = emotion_cat~Genere_cat+(1|ID),data=datax,
    family=binomial)
```

Più in dettaglio:

- La prima formula rappresenta un'analisi relativa solo alla variabilità dell'individuo, ovvero ad ID;
- La seconda formula rappresenta un'analisi relativa solo alla variabilità del DPCM, includendo ID;
- La terza formula rappresenta un'analisi relativa solo alla variabilità del reddito, includendo ID;
- La quarta formula rappresenta un'analisi relativa alla variabilità del reddito e del DPCM, includendo ID;
- La quinta formula rappresenta un'analisi relativa alla presenza di moderazione, ovvero di possibile correlazione tra DPCM e Reddito;
- La sesta formula rappresenta un'analisi relativa alla verifica di variabilità del genere, maschile o femminile, includendo ID.

L'input `data=datax` rappresenta il database contenente le variabili nominate in `formula`. L'input `family=binomial` rappresenta la specifica dell'analisi di interesse, ovvero binomiale.

2.3.2 Analisi del modello migliore tramite AIC

Un modo generale per confrontare i modelli a livello singolo (modelli che non includono effetti casuali o variabili latenti) è l'Akaike Information Criterion (AIC) (Profillidis e Botzoris 2019). Il criterio informativo di Akaike (AIC) è stato sviluppato dallo statistico giapponese Hirotugu Akaike. È una misura statistica per la valutazione comparativa tra modelli di serie temporali. Tuttavia, poiché l'AIC non si basa su un test di ipotesi, non può garantire la qualità di un modello rispetto ad altri modelli. Pertanto, nel caso in cui tutti i modelli oggetto di valutazione si adattino male rispetto a un dato insieme di dati o osservazioni, l'AIC indicherà solo il modello che si adatta un po' meglio ai dati o alle osservazioni disponibili rispetto agli altri.

L'AIC fornisce una stima delle informazioni perse quando viene utilizzato un modello specifico per rappresentare il processo che ha generato i dati. In un tale approccio, un modello bilancia tra la bontà dell'adattamento e la complessità.

Matematicamente, l'AIC è calcolato dalla seguente equazione:

$$AIC = -2 \times \frac{l}{n} + 2 \times \frac{k}{n} \tag{2.3}$$

dove:

- n rappresenta il numero di dati o osservazioni;
- k rappresenta il numero di parametri stimati (regressori + intercetta);
- l rappresenta la funzione di verosimiglianza logaritmica (assumendo errori normalmente distribuiti).

Quando si confrontano molti modelli alternativi, quello con il valore AIC minimo assicura un buon equilibrio tra bontà di adattamento e complessità. E' stato quindi implentanto il criterio AIC akaike per la stima del modello migliore, con il seguente codice R:

```
#criterio AIC akaike
AIC(mod0,mod1,mod2,mod3,mod4, mod5)
```

Capitolo 3

Discussione dei risultati

In questa sezione verranno presentati e discussi i risultati ottenuti per le sei `formula`: mod0, mod1, mod2, mod3, mod4 e mod5, rappresentanti i diversi approcci utilizzati per la stima e l'analisi dell'espressione dell'interazione sociale in un momento storico in cui l'unica interazione sociale concessa era contactless: i social media durante la pandemia da covid19, con riferimento temporale al primo lockdown in Italia tra Marzo e Aprile del 2020. Verrà inoltre presentata l'analisi tramite applicazione AIC Akaike.

3.1 Formula mod0

La funzione mod0:

```
mod0 = lme4::glmer(formula = emotion_cat~1+(1|ID),data=datax,family=
    binomial)
```

rappresenta un'analisi relativa solo alla variabilità dell'individuo, ovvero ad ID. Applicando la funzione `glmer`, sono stati ottenuti i seguenti risultati:

```
> summary(mod0)
Generalized linear mixed model fit by maximum likelihood (Laplace
    Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: emotion_cat ~ 1 + (1 | ID)
   Data: datap
```

```
     AIC      BIC   logLik deviance df.resid
   520.7    529.4   -258.4    516.7      566

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.5574  0.2772  0.2948  0.4079  1.0435

Random effects:
 Groups Name        Variance Std.Dev.
 ID     (Intercept) 1.35     1.162
Number of obs: 568, groups:  ID, 98

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.7957     0.1949   9.212   <2e-16 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1
            1
```

E' risultato come il modello non contiene termini, al di fuori di una costante. Non è possibile, perciò, effettuare un'analisi statistica rilevante tramite la formula `mod0`.

3.2 Formula mod1

La funzione mod1:

```
mod1 = lme4::glmer(formula = emotion_cat~dpcm_num+(1|ID),data=datax,family
    =binomial)
```

rappresenta un'analisi relativa solo alla variabilità del DPCM, includendo ID.

Applicando la funzione `glmer`, sono stati ottenuti i seguenti risultati:

```
> summary(mod1)
Generalized linear mixed model fit by maximum likelihood (Laplace
    Approximation) ['glmerMod']
 Family: binomial  ( logit )
```

```
Formula: emotion_cat ~ dpcm_num + (1 | ID)
   Data: datap

     AIC      BIC   logLik deviance df.resid
     522      535     -258      516      565

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.7685  0.2674  0.3088  0.4218  1.0828

Random effects:
 Groups Name        Variance Std.Dev.
 ID     (Intercept) 1.364    1.168
Number of obs: 568, groups:  ID, 98

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.58903    0.30833   5.154 2.55e-07 ***
dpcm_num     0.04507    0.05305   0.850    0.396
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1
            1

Correlation of Fixed Effects:
         (Intr)
dpcm_num -0.773
```

E' risultato come la formula adottata consente di effettuare una regressione dei dati analizzati, con l'output grafico descritto in Figura 3.1. L'analisi ha portato quindi ad un trend in cui la variabile risulta avere un trend crescente al susseguirsi dei DPCM.

3.3 Formula mod2

La funzione mod2:

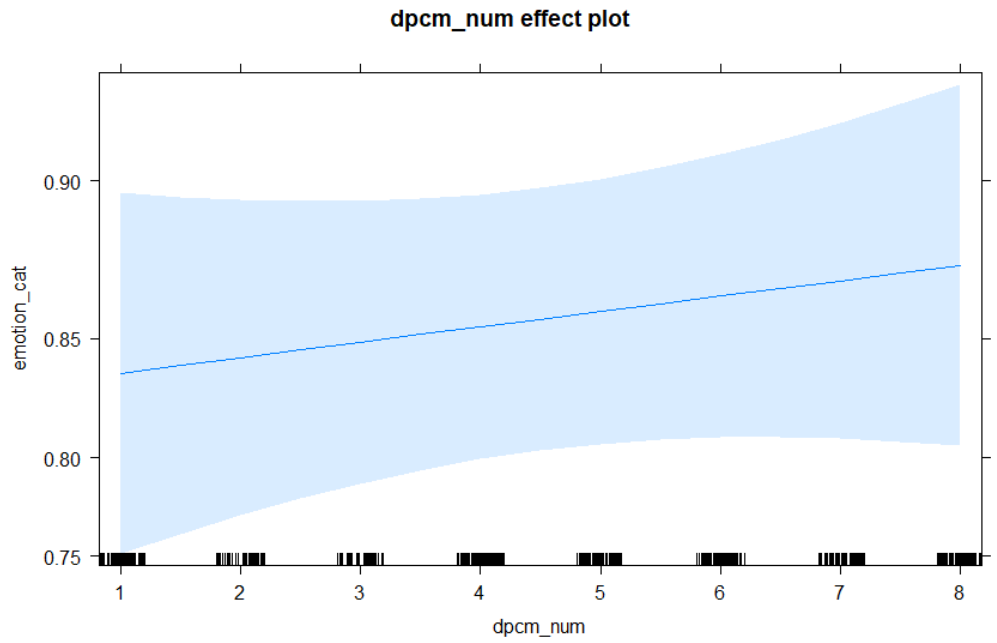

Figura 3.1: Analisi della formula mod1 tramite glmer

```
mod2 = lme4::glmer(formula = emotion_cat~Reddito+(1|ID),data=datax,family=
    binomial)
```

rappresenta un'analisi relativa solo alla variabilità del reddito, includendo ID.

Applicando la funzione `glmer`, sono stati ottenuti i seguenti risultati:

```
> summary(mod2)
Generalized linear mixed model fit by maximum likelihood (Laplace
    Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: emotion_cat ~ Reddito + (1 | ID)
   Data: datap

     AIC      BIC   logLik deviance df.resid
   529.0    563.8   -256.5    513.0      560

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.1408  0.2516  0.3231  0.4139  1.1115
```

```
Random effects:
 Groups Name        Variance Std.Dev.
 ID     (Intercept) 1.219    1.104
Number of obs: 568, groups:  ID, 98

Fixed effects:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)     2.6103     0.7037   3.709 0.000208 ***
Reddito10-15   -0.3589     1.0436  -0.344 0.730909
Reddito15-26   -0.8522     0.7626  -1.117 0.263798
Reddito26-55   -0.6965     0.7719  -0.902 0.366862
Reddito55-75   -1.3143     0.8564  -1.535 0.124857
Reddito75-120  -1.2235     0.8348  -1.466 0.142756
Reddito120-    -0.9874     1.0547  -0.936 0.349159
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1
            1

Correlation of Fixed Effects:
            (Intr) R10-15 R15-26 R26-55 R55-75 R75-12
Reddit10-15 -0.664
Reddit15-26 -0.911  0.614
Reddit26-55 -0.895  0.607  0.830
Reddit55-75 -0.813  0.547  0.749  0.739
Reddt75-120 -0.835  0.561  0.768  0.758  0.684
Reddito120- -0.660  0.444  0.608  0.600  0.541  0.555
```

E' risultato come la formula adottata non consente di effettuare una regressione dei dati analizzati, a causa dell'alta volatilità dei dati analizzati, come mostrato attraverso l'output grafico descritto in Figura 3.2. L'analisi quindi non consente di ottenere un trend rilevante a livello statistico in cui la variabile risulta avere un trend crescente/decrescente in funzione del Reddito.

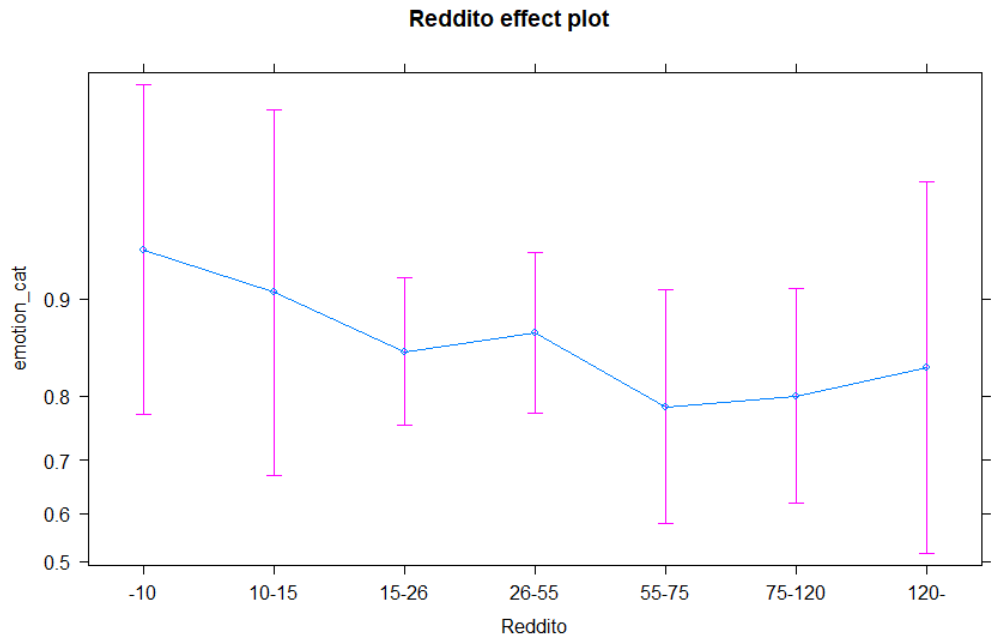

Figura 3.2: Analisi della formula mod2 tramite glmer

3.4 Formula mod3

La funzione mod3:

```
mod3 = lme4::glmer(formula = emotion_cat~dpcm_num+Reddito+(1|ID),data=
    datax,family=binomial)
```

rappresenta un'analisi relativa alla variabilità del reddito e del DPCM, includendo ID.

Applicando la funzione `glmer`, sono stati ottenuti i seguenti risultati:

```
> mod3 = lme4::glmer(formula = emotion_cat~dpcm_num+Reddito+(1|ID),, data=
    datap,family=binomial, control=glmerControl(optimizer="bobyqa",optCtrl
    =list(maxfun=2e5)))
> summary(mod3)
Generalized linear mixed model fit by maximum likelihood (Laplace
    Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: emotion_cat ~ dpcm_num + Reddito + (1 | ID)
   Data: datap
```

```
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05)
    )

     AIC      BIC   logLik deviance df.resid
   530.3    569.4   -256.2    512.3      559

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2679  0.2573  0.3234  0.4214  1.1534

Random effects:
 Groups Name        Variance Std.Dev.
 ID     (Intercept) 1.232    1.11
Number of obs: 568, groups:  ID, 98

Fixed effects:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.40271    0.74540   3.223  0.00127 **
dpcm_num       0.04451    0.05299   0.840  0.40094
Reddito10-15  -0.36144    1.04642  -0.345  0.72979
Reddito15-26  -0.84629    0.76441  -1.107  0.26824
Reddito26-55  -0.69032    0.77381  -0.892  0.37234
Reddito55-75  -1.31984    0.85852  -1.537  0.12421
Reddito75-120 -1.21437    0.83708  -1.451  0.14686
Reddito120-   -1.00123    1.05777  -0.947  0.34387
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1
            1

Correlation of Fixed Effects:
            (Intr) dpcm_n R10-15 R15-26 R26-55 R55-75 R75-12
dpcm_num    -0.324
Reddit10-15 -0.627 -0.003
Reddit15-26 -0.865  0.008  0.614
Reddit26-55 -0.850  0.009  0.607  0.830
Reddit55-75 -0.766 -0.010  0.546  0.748  0.738
```

```
Reddt75-120 -0.793  0.010  0.560  0.767  0.757  0.684
Reddito120- -0.618 -0.017  0.444  0.607  0.599  0.541  0.555
```

E' risultato come la formula adottata non consente di effettuare una regressione dei dati analizzati, a causa dell'alta volatilità dei dati, principalmente per il Reddito, come mostrato attraverso l'output grafico descritto in Figura 3.3. L'analisi quindi non consente di ottenere un trend rilevante a livello statistico. Il problema di base è che si hanno più osservazioni nel set di dati per ID , ma alcune probabilmente presentano lo stesso valore di risposta, quindi l'effetto casuale è confuso con tutto il resto. In particolare, il gruppo Reddito e le sue interazioni sembrano essere problematici.

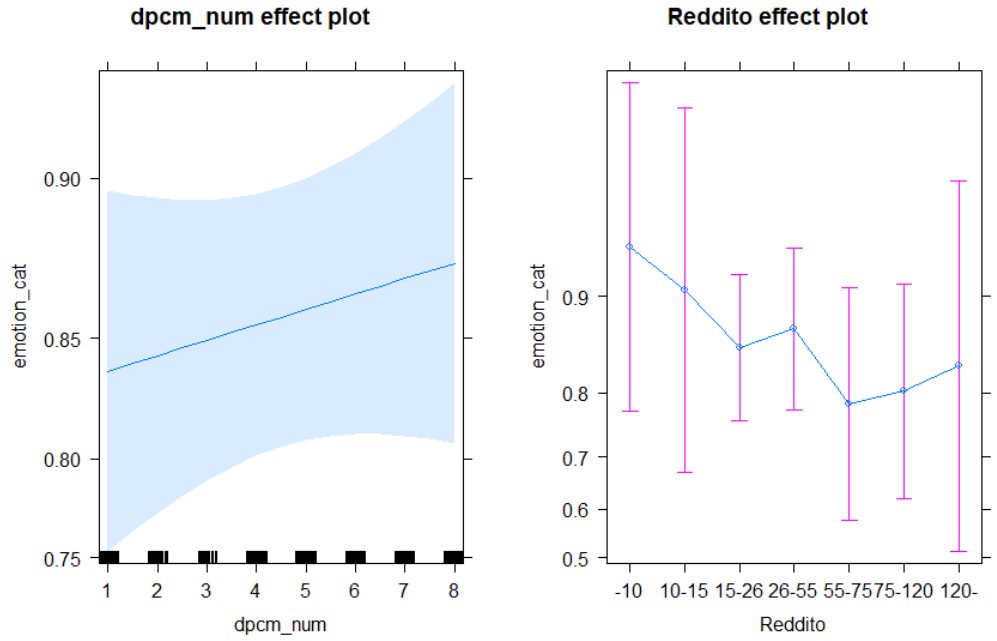

Figura 3.3: Analisi della formula mod3 tramite glmer

3.5 Formula mod4

La funzione mod4:

```
mod4 = lme4::glmer(formula = emotion_cat~dpcm_num*Reddito+(1|ID),data=
    datax,family=binomial)
```

rappresenta un'analisi relativa alla presenza di moderazione, ovvero di possibile correlazione tra DPCM e Reddito.

Applicando la funzione `glmer`, sono stati ottenuti i seguenti risultati:

```
> mod4 = lme4::glmer(formula = emotion_cat~dpcm_num*Reddito+(1|ID), data=
    datap,family=binomial, control=glmerControl(optimizer="bobyqa",optCtrl
    =list(maxfun=2e5)))
> summary(mod4)
Generalized linear mixed model fit by maximum likelihood (Laplace
    Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: emotion_cat ~ dpcm_num * Reddito + (1 | ID)
   Data: datap
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05)
    )

     AIC      BIC   logLik deviance df.resid
   538.0    603.2   -254.0    508.0      553

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2502  0.2329  0.3195  0.4187  1.3028

Random effects:
 Groups Name        Variance Std.Dev.
 ID     (Intercept) 1.383    1.176
Number of obs: 568, groups:  ID, 98

Fixed effects:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)           2.85292    1.44311   1.977   0.0481 *
dpcm_num             -0.04420    0.25247  -0.175   0.8610
Reddito10-15         -1.24524    1.99038  -0.626   0.5316
Reddito15-26         -1.58032    1.52793  -1.034   0.3010
Reddito26-55         -0.50244    1.56599  -0.321   0.7483
Reddito55-75         -2.30475    1.68575  -1.367   0.1716
Reddito75-120        -1.09953    1.68201  -0.654   0.5133
```

```
Reddito120-            -2.38893    1.97023  -1.213   0.2253
dpcm_num:Reddito10-15   0.19025    0.35793   0.532   0.5951
dpcm_num:Reddito15-26   0.15834    0.26867   0.589   0.5556
dpcm_num:Reddito26-55  -0.03956    0.27325  -0.145   0.8849
dpcm_num:Reddito55-75   0.20769    0.29629   0.701   0.4833
dpcm_num:Reddito75-120 -0.03104    0.29695  -0.105   0.9168
dpcm_num:Reddito120-    0.29204    0.34568   0.845   0.3982
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1
           1

Correlation matrix not shown by default, as p = 14 > 12.
Use print(x, correlation=TRUE)  or
    vcov(x)        if you need it
```

E' risultato come non esiste una correlazione. Analizzando la matrice, si ottiene infatti:

```
> vcov(mod4)
14 x 14 Matrix of class "dpoMatrix"
                      (Intercept)    dpcm_num Reddito10-15 Reddito15-26
                         Reddito26-55 Reddito55-75 Reddito75-120
(Intercept)              2.0825652 -0.31498994   -2.0774851   -2.0789195
      -2.0694398   -2.0849552    -2.0708136
dpcm_num                -0.3149899  0.06374266    0.3150246    0.3150118
       0.3150585    0.3149743     0.3150571
Reddito10-15            -2.0774851  0.31502459    3.9615942    2.0765975
       2.0742407    2.0780611     2.0746080
Reddito15-26            -2.0789195  0.31501177    2.0765975    2.3345716
       2.0728480    2.0800144     2.0735030
Reddito26-55            -2.0694398  0.31505855    2.0742407    2.0728480
       2.4523269    2.0671897     2.0803873
Reddito55-75            -2.0849552  0.31497432    2.0780611    2.0800144
       2.0671897    2.8417643     2.0690367
Reddito75-120           -2.0708136  0.31505712    2.0746080    2.0735030
       2.0803873    2.0690367     2.8291543
Reddito120-             -2.0825034  0.31499618    2.0774925    2.0789116
```

```
       2.0695717    2.0848574     2.0709274
dpcm_num:Reddito10-15    0.3155643 -0.06374063   -0.5974512   -0.3152808
      -0.3145388   -0.3157496    -0.3146489
dpcm_num:Reddito15-26    0.3156792 -0.06373926   -0.3151961   -0.3516101
      -0.3144230   -0.3159068    -0.3145566
dpcm_num:Reddito26-55    0.3142725 -0.06374622   -0.3148465   -0.3146793
      -0.3684382   -0.3140036    -0.3155785
dpcm_num:Reddito55-75    0.3167405 -0.06373283   -0.3154542   -0.3158192
      -0.3134300   -0.4247162    -0.3137729
dpcm_num:Reddito75-120   0.3135639 -0.06375094   -0.3146761   -0.3143549
      -0.3163886   -0.3130423    -0.4287437
dpcm_num:Reddito120-     0.3162341 -0.06373675   -0.3153344   -0.3155894
      -0.3139146   -0.3166569    -0.3141567
                        Reddito120- dpcm_num:Reddito10-15 dpcm_num:
                            Reddito15-26 dpcm_num:Reddito26-55
(Intercept)              -2.0825034            0.31556431
    0.31567923            0.31427246
dpcm_num                 0.3149962           -0.06374063
    -0.06373926           -0.06374622
Reddito10-15             2.0774925           -0.59745117
    -0.31519608           -0.31484645
Reddito15-26             2.0789116           -0.31528083
    -0.35161014           -0.31467925
Reddito26-55             2.0695717           -0.31453879
    -0.31442300           -0.36843819
Reddito55-75             2.0848574           -0.31574956
    -0.31590684           -0.31400361
Reddito75-120            2.0709274           -0.31464886
    -0.31455655           -0.31557845
Reddito120-              3.8818241           -0.31556115
    -0.31567529           -0.31428929
dpcm_num:Reddito10-15   -0.3155612            0.12811211
    0.06379460            0.06368450
dpcm_num:Reddito15-26   -0.3156753            0.06379460
    0.07218394            0.06367052
dpcm_num:Reddito26-55   -0.3142893            0.06368450
```

```
    0.06367052            0.07466727
dpcm_num:Reddito55-75   -0.3167208            0.06387710
    0.06390656            0.06355198
dpcm_num:Reddito75-120  -0.3135915            0.06362961
    0.06360307            0.06390492
dpcm_num:Reddito120-    -0.5651318            0.06383799
    0.06385853            0.06360999
                      dpcm_num:Reddito55-75 dpcm_num:Reddito75-120 dpcm_
                          num:Reddito120-
(Intercept)                        0.31674051             0.31356388
             0.31623414
dpcm_num                          -0.06373283            -0.06375094
            -0.06373675
Reddito10-15                      -0.31545422            -0.31467612
            -0.31533435
Reddito15-26                      -0.31581923            -0.31435491
            -0.31558941
Reddito26-55                      -0.31342998            -0.31638863
            -0.31391456
Reddito55-75                      -0.42471617            -0.31304233
            -0.31665693
Reddito75-120                     -0.31377292            -0.42874368
            -0.31415669
Reddito120-                       -0.31672078            -0.31359151
            -0.56513178
dpcm_num:Reddito10-15              0.06387710             0.06362961
             0.06383799
dpcm_num:Reddito15-26              0.06390656             0.06360307
             0.06385853
dpcm_num:Reddito26-55              0.06355198             0.06390492
             0.06360999
dpcm_num:Reddito55-75              0.08778969             0.06337263
             0.06404621
dpcm_num:Reddito75-120             0.06337263             0.08818046
             0.06348464
dpcm_num:Reddito120-               0.06404621             0.06348464
```

```
                0.11949206
```

La formula adottata non consente di effettuare una regressione dei dati analizzati, come mostrato attraverso l'output grafico descritto in Figura 3.4. L'analisi quindi non consente di ottenere un trend rilevante a livello statistico.

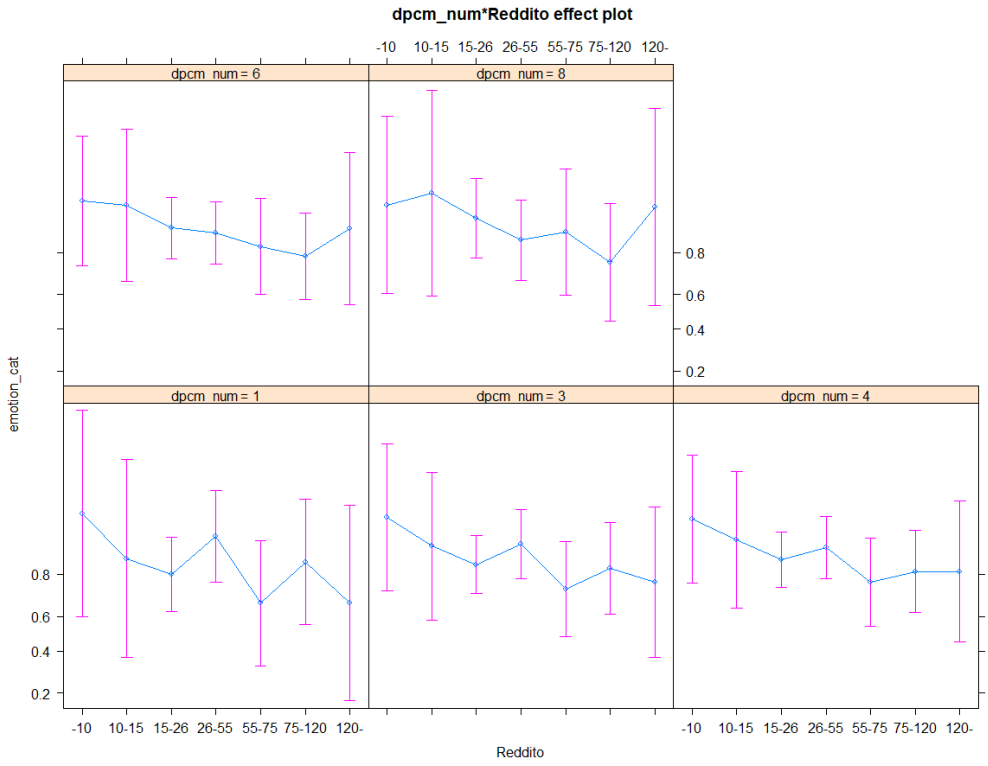

Figura 3.4: Analisi della formula mod4 tramite glmer

3.6 Formula mod5

La funzione mod5:

```
mod5 = lme4::glmer(formula = emotion_cat~Genere_cat+(1|ID),data=datax,
    family=binomial)
```

rappresenta un'analisi relativa alla verifica di variabilità del genere, maschile o femminile, includendo ID.

Applicando la funzione `glmer`, sono stati ottenuti i seguenti risultati:

```
> summary(mod5)
Generalized linear mixed model fit by maximum likelihood (Laplace
    Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: emotion_cat ~ Genere_cat + (1 | ID)
   Data: datap

     AIC      BIC   logLik deviance df.resid
   522.2    535.3   -258.1    516.2      565

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.6087  0.2708  0.3117  0.4184  1.0218

Random effects:
 Groups Name        Variance Std.Dev.
 ID     (Intercept) 1.333    1.154
Number of obs: 568, groups:  ID, 98

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.8845     0.2351   8.016 1.09e-15 ***
Genere_catM  -0.2608     0.3642  -0.716    0.474
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1
            1

Correlation of Fixed Effects:
            (Intr)
Genere_catM -0.563
```

La formula adottata ha prodotto il seguente output grafico descritto in Figura 3.4. Tuttavia, il numero di campione di genere femminile supera molto il numero di genere maschile. L'analisi quindi non consente di ottenere un trend rilevante a livello statistico.

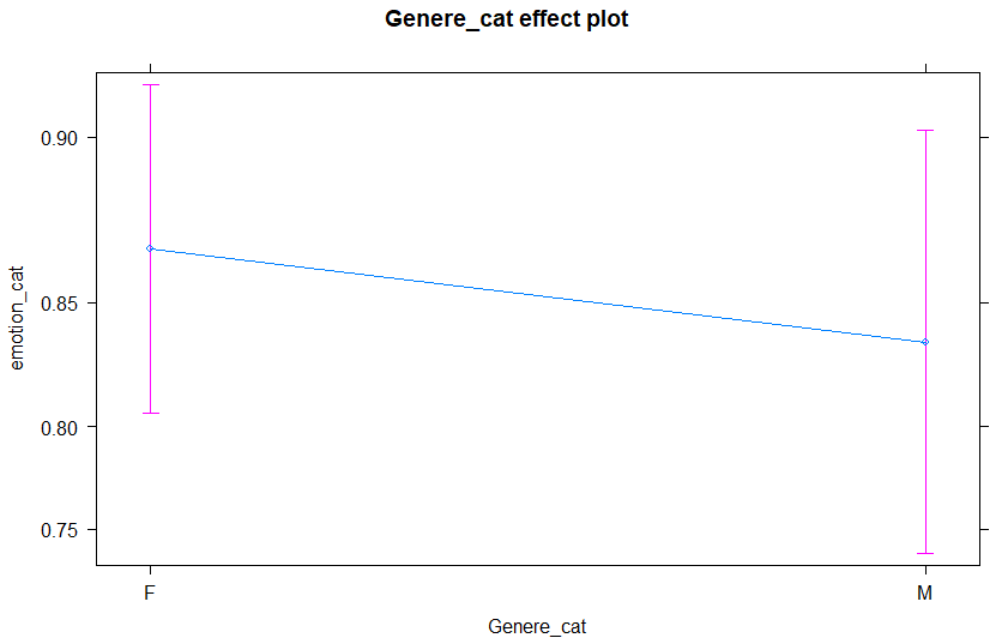

Figura 3.5: Analisi della formula mod5 tramite glmer

3.7 Applicazione Akaike Information Criterion (AIC)

E' stato infine applicato l'AIC, e sono stati ottenuti i seguenti risultati:

```
> #criterio AIC akaike
> AIC(mod0,mod1,mod2,mod3,mod4, mod5)
      df      AIC
mod0  2 520.7413
mod1  3 522.0188
mod2  8 529.0286
mod3  9 530.3225
mod4 15 538.0397
mod5  3 522.2306
```

La formula mod1:

```
mod1 = lme4::glmer(formula = emotion_cat~dpcm_num+(1|ID),data=datax,family
    =binomial)
```

risulta essere fornire il risultato migliore, tra quelli analizzati, e presenta le seguenti prestazioni:

```
> performance::r2(mod1)
# R2 for Mixed Models

  Conditional R2: 0.295
     Marginal R2: 0.002
```

3.8 Limiti del modello e dello strumento

Si è visto come, per analizzare il caso studio oggetto di questo lavoro di tesi, è stato applicato sul software statistico R tramite il pacchetto glmer che deriva dal pacchetto lme4, ovvero l'analisi si è basata sull'applicazione dei Modelli Misti Lineari Generalizzati e famiglia binomiale. Tale applicazione ha portato ad una formula rilevante, la formula mod1, in cui però, è stato possibile solo valutare l'effetto DPCM sull'emotività, escludendo il genere ed il reddito. Inoltre, sebbene l'applicazione del criterio AIC abbia individuato il mod1 la formula migliore, le prestazioni risultanti si sono rilevate molto basse:

```
> performance::r2(mod1)
# R2 for Mixed Models
     Conditional R2: 0.295
     Marginal R2: 0.002
```

La presenza di alta volatilità dei dati e di stagionalità (emotività al variare di DPCM) può aver condizionato l'applicazione, ed i Modelli Misti Lineari Generalizzati con famiglia binomiale non si sono rivelati modelli chiave per la risoluzione di tale applicazione, non garantendo una prospettiva funzionale su tutte le covariate.

Una prospettiva funzionale sui dati ha importanti vantaggi. Non è chiaro, tuttavia, come modellare serie temporali che vengono registrate con una frequenza elevata, soprattutto quando è coinvolta la stagionalità (multipla, nel nostro caso la presenza di 8 DPCM).

Text	Compound value
Le brutte intenzioni la maleducazione la tua brutta figura di ieri sera	-0,8689

Tabella 3.1: Esempio di short text ironico/riferimento culturale

Una serie temporale univariata non stazionaria potrebbe essere stazionaria se modellata come serie temporale funzionale, perché confrontiamo i dati che sono ragionevole confrontare(stesse persone, ma con reazioni emotive differenti da un punto di vista temporale per ogni DPCM).

L'analisi funzionale dei dati (FDA) è un'area di ricerca attiva e può essere utilizzata in varie applicazioni, e potrebbe rivelarsi utile per il caso studio in esame. Matematicamente una serie temporale funzionale è una raccolta di dati reali indicizzati in base al tempo (come una serie temporale univariata), ma osservando le funzioni associate invece che dati a valori reali. Come continuazione e sviluppo futuro del presente lavoro di tesi, si applicheranno quindi tecniche di analisi funzionali di dati per serie temporali, con l'obiettivo di ottenere un miglior riscontro e grado di correlazione.

I limiti dello strumento, VADER, riguardano le sue caratteristiche lexicon-based: basandosi sul lessico, non riesce a cogliere al meglio il contesto, in particolar modo l'ironia. Considerando anche che i social media premiano molto i contenuti ironici o contenuti che riguardano la cultura pop (si pensi ai rebranding di Tavernello, Zuegg, Unieuro), questo potrebbe essere stato un limite non da poco. Per fare qualche esempio: tra gli short text era presente nella 3.1 a cui VADER ha dato un punteggio molto negativo, mentre il commento è palesemente ironico per il lettore che conosce il riferimento culturale a cui è legata la frase.

Per concludere sui limiti dello strumento, data la natura dei social media molto legata a subculture e reti sociali con riferimenti ad argomenti che nascono all'interno della rete e che si appropriano di un lessico esclusivo, un modello lexicon-based potrebbe far perdere dell'informazione preziosa. Questa problematica potrebbe essere in parte risolta attraverso l'uso di Topic

Model che vanno a scovare la gli argomenti dei dati testuali e integrarli con una sentiment analysis.

Bibliografia

Bates, Douglas et al. (gen. 2014). *Package Lme4: Linear Mixed-Effects Models Using Eigen and S4*.

Hutto, C.J. e E. Gilbert (2014). «VADER: a parsimonious rule-based model for sentiment analysis of social media text». In: *Proceedings of the eight International AAAI Conference on Weblogs and Social Media*, pp. 1–10.

Kennedy, B. et al. (2021). «Text Analysis for Psychology: Methods, Principles, and Practices». In:

Liddy, E. (2001). *Natural Language Processing*. New York: Marcel Decker, Inc.

Lindstrom, M.J. e D.M. Bates (1990). «Nonlinear Mixed Effects Models for Repeated Measures Data». In: *Biometrics* 46, pp. 673–687.

NLTK.org. URL: https://www.nltk.org/.

Profillidis, V.A. e G.N. Botzoris (2019). *Chapter 6 - Trend Projection and Time Series Methods, Editor(s): V.A. Profillidis, G.N. Botzoris, Modeling of Transport Demand*. Amsterdam, Netherlands: Elsevier.

Salvan, A., N. Sartori e L. Pace (2020). *Modelli Lineari Generalizzati*. Springer Verlag.

Wikipedia, Modello Lineare (2013). URL: http://it.wikipedia.org/wiki/Regressione_lineare.

Appendice A

Codice Python utilizzato

```
import nltk
#nltk.download('vader_lexicon')
import pandas as pd
import pyreadr
from nltk.sentiment.vader import SentimentIntensityAnalyzer
# FUNZIONE PER CALCOLARE POLARITY SCORE PER OGNI FRASE
def nltk_sentiment(sentence):
    nltk_sentiment = SentimentIntensityAnalyzer()
    score = nltk_sentiment.polarity_scores(sentence)
    return score
# FINE FUNZIONE
# LETTURA DEL FILE
filename = "./dataset.Rdata"
dataset = pyreadr.read_r(filename)
#print(dataset.keys()) # stampa objects del dataset: c'   solo "Testo_
    Finale_Social_Filtered"
df1 = dataset["Testo_Finale_Social_Filtered"] # estraggo il dataframe dall
    'oggetto "Testo_Finale_Social_Filtered"
#print(df1.head()) # stampa di prova

from tweet import Tweet
from emoji_translate.emoji_translate import Translator
```

```
emo = Translator(exact_match_only=False, randomize=True)

dataset_author = list(df1["author"].to_numpy()) # prendo solo i dati della
    colonna "text" e li trasformo in un array
dataset_dpcm = list(df1["DPCM"].to_numpy())
with open('traduzioni.txt') as f:
    dataset_text = f.readlines()

tweets = []
for i in range(0, len(dataset_author)):
    dataset_text[i] = emo.demojify(dataset_text[i])
    tweets.append(Tweet(dataset_author[i], dataset_text[i], dataset_dpcm[i
        ]))

#print(tweets[50].author) # = 51esimo in Rdata
#print(tweets[50].text)
#print(tweets[50].dpcm)
#dataset_text[5580].encode("latin1")
#print(dataset_text[5580])
#append_df_to_excel("to_translate.xlsx", df1["text"], header = None, index
    = False)
#print(dataset_text[0:10]) # stampa di prova
# FINE LETTURA DEL FILE

# PER OGNI FRASE DEL DATASET, CHIAMO LA FUNZIONE nltk_sentiment
nltk_results = [nltk_sentiment(row) for row in dataset_text]
for res in nltk_results:
    if res['neg'] > res['pos'] and res['neg'] > res['neu']:
        res['neg'] = 1
        res['pos'] = 0
        res['neu'] = 0
    elif res['pos'] > res['neg'] and res['pos'] > res['neu']:
        res['pos'] = 1
        res['neg'] = 0
        res['neu'] = 0
    else:
```

```
        res['neu'] = 1
        res['pos'] = 0
        res['neg'] = 0

#CREO IL DATAFRAME DEI RISULTATI
results_df = pd.DataFrame(nltk_results)

text_df = pd.DataFrame(dataset_text, columns = ['text'])
author_df = pd.DataFrame(dataset_author, columns = ['author'])
dpcm_df = pd.DataFrame(dataset_dpcm, columns = ['dpcm'])
nltk_df = text_df.join(results_df)
nltk_df = author_df.join(nltk_df)
nltk_df = dpcm_df.join(nltk_df)
# SCRIVO I RISULTATI SU FILE EXCEL
nltk_df.to_excel("output_21102021.xlsx")
```

Appendice B

Codice R utilizzato

Codice B.1: Codice R library(readxl) $\text{dataset}_u ltimo < -read_e xcel(”C : /Users/elisa/Desktop/Tesiordinata/dataset_u ltimo.xlsx”)View(dataset_u ltimo)datax$

```
#inizializzazione variabili
str(datax)
datax$dpcm_num = as.numeric(datax$dpcm)
datax$Genere_cat = as.factor(datax$Genere)
datax$Genere_cat = factor(x = datax$Genere_cat,)
datax$dpcm_cat = factor(x = datax$dpcm_num,ordered = TRUE,)
datax$Reddito = factor(x = datax$Reddito,labels = c("10–15","15–26","26–55
    ","55–75","75–120","–10","120–"))
datax$Reddito = relevel(datax$Reddito,ref = "–10")
datax$Regione = as.factor(x = datax$Regione)
datax$Regione = factor(x = datax$Regione,levels = unique(datax$Regione),
    labels = c("N","N","N","C","C","N","N","N","S","S","C","S","N"))
datax$TitoloStudio = as.factor(x = datax$TitoloStudio)
datax$StatoCivile = factor(x = datax$StatoCivile,levels = unique(datax$
    StatoCivile),labels = c("Single","Fidanz","Fidanz","Coniug"))
datax$Accesso.Social = factor(x = datax$Accesso.Social,levels = unique(
    datax$Accesso.Social),labels = c("10–","1–5","6–10","–1"))
datax$Accesso.Social = relevel(x = datax$Accesso.Social,ref ="–1")
```

```
datax$TitoloStudio = factor(x = datax$TitoloStudio,levels = unique(datax$
    TitoloStudio),labels = c("L","NL","L","NL"))
colnames(datax)[c(2,18:19)] = c("ID","compound_cat","emotion")
datax$compound_cat = as.factor(datax$compound_cat)
datax$emotion_cat = as.factor(datax$emotion)

#plottare con la funzione di densit
#densx <- dbinom(datax$emotion, 800, NA)
#plot(datax$emotion_cat, densx, ylim=c(min(X),max(X))
##
barplot(table(datax$emotion_cat))
table(datax$emotion_cat)

#plottare emozioni ~ dpcm
X = xtabs(formula = ~emotion_cat+dpcm_num,data = datax)
plot(x = 1:11,y=X[1,],bty="n",type="b",xlim=c(0,12),xlab="dpcm",ylab="
    emotion",ylim=c(min(X),max(X)),col="red")
points(x = 1:11,y=X[2,],col="blue",type="b")
legend("topleft",legend = c("assenza","presenza"),col = c("red","blue"),
    bty = "n",lty=1)

#plottare emozioni ~ genere
X = xtabs(formula = ~emotion_cat+Genere_cat,data = datax)
plot(x = 0:1,y=X[1,],bty="n",type="b",xlim=c(0,1),xlab="genere",ylab="
    emotion",ylim=c(min(X),max(X)),col="red")
points(x = 0:1,y=X[2,],col="blue",type="b")
legend("topleft",legend = c("assenza","presenza"),col = c("red","blue"),
    bty = "n",lty=1)

table(datax$Genere_cat)
barplot(table(datax$Genere_cat))

#plottare emozioni ~ reddito
X = xtabs(formula = ~emotion_cat+Reddito,data = datax)
plot(x = 1:7,y=X[1,],bty="n",type="b",xlim=c(0,7),xlab="reddito",ylab="
    emotion",ylim=c(min(X),max(X)),col="red")
```

```
points(x = 1:7,y=X[2,],col="blue",type="b")
legend("topleft",legend = c("assenza","presenza"),col = c("red","blue"),
    bty = "n",lty=1)

table(datax$Eta)
barplot(table(datax$Eta))

#plottare emozioni ~ e t
X = xtabs(formula = ~emotion_cat+Eta,data = datax)
plot(x = as.numeric(colnames(X)),y=X[1,],bty="n",type="b",xlab="reddito",
    ylab="emotion",ylim=c(min(X),max(X)),col="red")
points(x = as.numeric(colnames(X)),y=X[2,],col="blue",type="b")
legend("topright",legend = c("assenza","presenza"),col = c("red","blue"),
    bty = "n",lty=1)

#plottare emozioni ~ stato civile
X = xtabs(formula = ~emotion_cat+StatoCivile,data = datax)
plot(x = 1:3,y=X[1,],bty="n",type="b",xlab="reddito",ylab="emotion",ylim=c
    (min(X),max(X)),col="red")
points(x = 1:3,y=X[2,],col="blue",type="b")
legend("topright",legend = c("assenza","presenza"),col = c("red","blue"),
    bty = "n",lty=1)

#plottare emozioni ~ Accesso social
X = xtabs(formula = ~emotion_cat+Accesso.Social,data = datax)
plot(x = 1:4,y=X[1,],bty="n",type="b",xlab="reddito",ylab="emotion",ylim=c
    (min(X),max(X)),col="red")
points(x = 1:4,y=X[2,],col="blue",type="b")
legend("topright",legend = c("assenza","presenza"),col = c("red","blue"),
    bty = "n",lty=1)

#plottare emozioni ~ titolo di studio
X = xtabs(formula = ~emotion_cat+TitoloStudio,data = datax)
plot(x = 0:1,y=X[1,],bty="n",type="b",xlab="reddito",ylab="emotion",ylim=c
    (min(X),max(X)),col="red")
points(x = 0:1,y=X[2,],col="blue",type="b")
```

```
legend("topright",legend = c("assenza","presenza"),col = c("red","blue"),
    bty = "n",lty=1)

library(ggplot2)
ggplot(data=datax,aes(x=dpcm_num,y=emotion))



mod0 = lme4::glmer(formula = emotion_cat~1+(1|ID),data=datax,family=
    binomial)
mod1 = lme4::glmer(formula = emotion_cat~dpcm_num+(1|ID),data=datax,family
    =binomial)
mod2 = lme4::glmer(formula = emotion_cat~Reddito+(1|ID),data=datax,family=
    binomial)
mod3 = lme4::glmer(formula = emotion_cat~dpcm_num+Reddito+(1|ID),data=
    datax,family=binomial)
mod4 = lme4::glmer(formula = emotion_cat~dpcm_num*Reddito+(1|ID),data=
    datax,family=binomial)
mod5 = lme4::glmer(formula = emotion_cat~Genere_cat+(1|ID),data=datax,
    family=binomial)
#criterio di informazioni di Akaike
AIC(mod0,mod1,mod2,mod3,mod4, mod5)
#vedere R^2
performance::r2(mod1)

#summary dei modelli
summary(mod3)
summary(mod1)
summary(mod0)
summary(mod4)

#plot col pacchetto effects
plot(effects::allEffects(mod0))
plot(effects::allEffects(mod2))

plot(effects::allEffects(mod1))
plot(effects::allEffects(mod3))
```

```
plot(effects::allEffects(mod5))
```