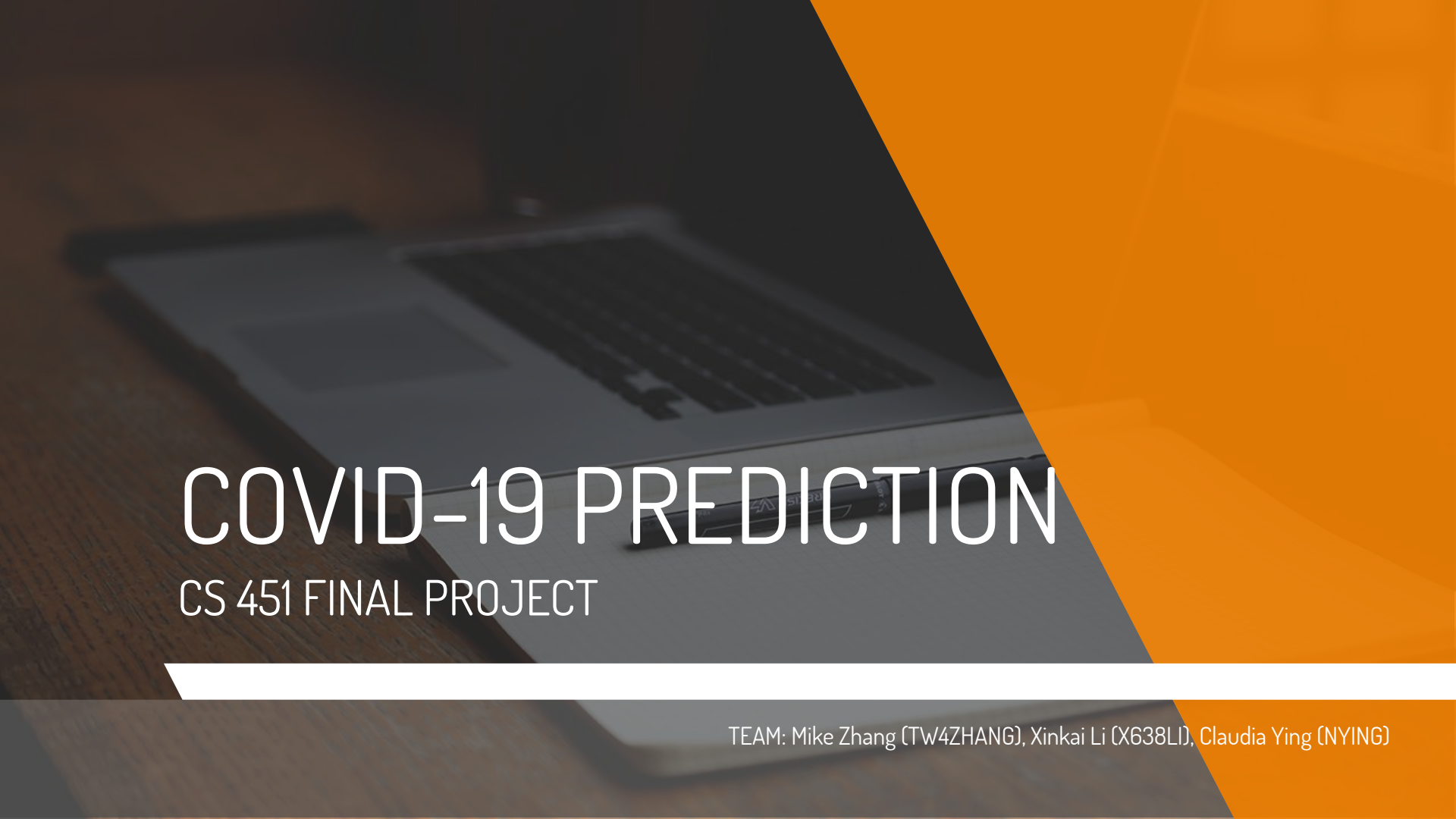


COVID-19 PREDICTION

CS 451 FINAL PROJECT

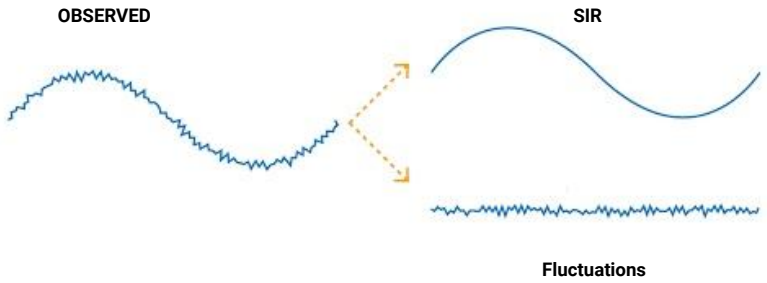
TEAM: Mike Zhang (TW4ZHANG), Xinkai Li (X638LI), Claudia Ying (NYING)

Idea

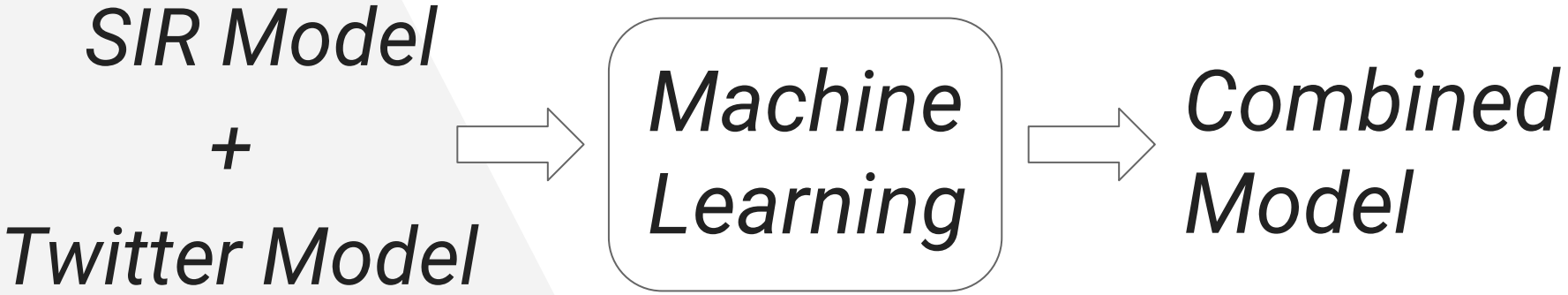
Use Twitter data to improve the performance of an epidemiological model like SIR to predict the future covid-19 infection case

Why Will it Work?

- SIR model
  - Adv.: simple, reasonably predictive
  - Dis.: over-simplified
- Fluctuations
  - Events such as quarantine, border closure.
  - Twitter



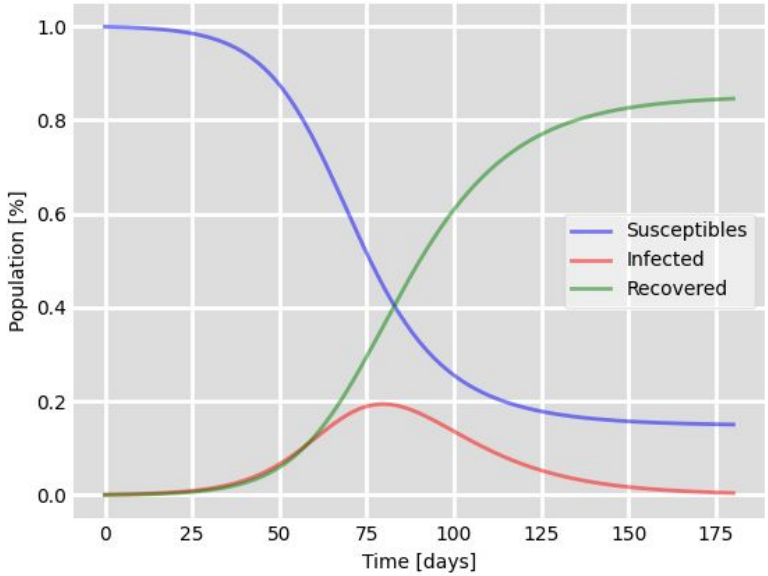
Methodology


SIR Model
+
Twitter Model
Machine Learning
Combined Model

SIR Model

- 3 differential equations
  - susceptible (S), infected (I), and resistant (R).
  -

$$\begin{aligned}
\frac{dS}{dt} &= -\beta \frac{SI}{N} \\
\frac{dI}{dt} &= \beta \frac{SI}{N} - \gamma I \\
\frac{dR}{dt} &= \gamma I \\
N &= S + I + R
\end{aligned}$$



Twitter Significant Word Extraction

(March 4 to March 31) && (English)

8 Million Tweets

Tokenize
Normalize
Stop Word
Lemmatize

Stem
Unigram &
Bigram
BOW

1000 most frequently occurred Unigrams/Bigrams

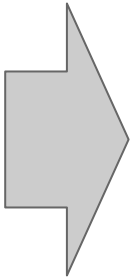
Calculate Correlation

*Top **20 words** having strongest correlation with # of new cases*

… Like This

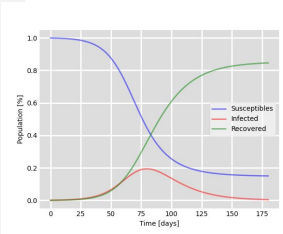
```
df.limit(10).toPandas().drop(columns='user_id')
```

	status_id	created_at	text	lang
0	1238253442063310848	2020-03-13T00:00:00Z	The UFC is about to be the most popular sport ...	en
1	1238253441778098177	2020-03-13T00:00:00Z	The great toilet paper depression of 2020 #Toi...	en
2	1238253440486313988	2020-03-13T00:00:00Z	The 'Spotlight Show' with @janeyleegrace on @u...	en
3	1238253439051870208	2020-03-13T00:00:00Z	Because we all the time in the world right? @s...	en
4	1238253440821649408	2020-03-13T00:00:00Z	French pastry chef shows off Easter eggs model...	en
5	1238253442034020354	2020-03-13T00:00:00Z	ICYMI - Hour 2 of #TheGamePlan with @DaveWNSP ...	en
6	1238253441564266496	2020-03-13T00:00:00Z	With rising #Coronavirus cases in India, which...	en
7	1238253441517928448	2020-03-13T00:00:00Z	#ICYMI: #Ontario #MPPs may temporarily suspend...	en
8	1238253440603541504	2020-03-13T00:00:00Z	Despite having only 3 confirmed #coronavirus c...	en
9	1238253440461135873	2020-03-13T00:00:00Z	Autonomous #Robots Are Helping Kill #Coronavir...	en



	word	correlation
0	covid	0.627775
1	coronaviru	0.487047
2	peopl	0.486824
3	get	0.440345
4	amp	0.603396
...	...	...
994	wife	0.217813
995	coronaviruschalleng	0.163871
996	predict	0.579344
997	novel_coronaviru	0.452516
998	whitehous	0.528342

ML Combination



+

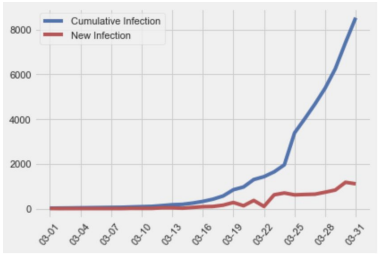
Simple Linear Regression

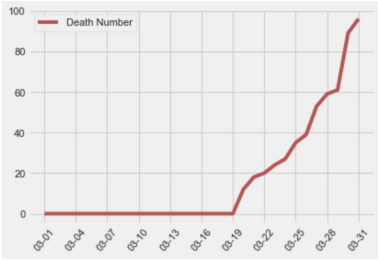
Final Model

Implementation

Which Data to Use?

- Data about covid-19 from www.canada.ca
  - # Confirmed, Susceptible, Recovery cases for covid-19 from March 4 to March 31st
  - Population of Canada





- Tweets
  - Hashtag #Covid19, #Coronavirus, etc
  - ~40M Tweets from March 4 to March 31st
  - ~8M Tweets known to be in English

SIR Model Implementation

- Clean using Spark SQL
- Using least square method to determine the parameters for SIR
- Solve ODEs for SIR model using deSolve library
- Optimize parameters using nlminb function
- Use the SIR model to make analysis and predictions

```
library(deSolve)
N <- 10000000 # population
I0 <- 24 # initial infected case
RM0 <- 24*0.15 # initial recover case
S0 <- N - I0 - RM0 # inital susceptible population
init <- c(S = S0, I = I0, R = RM0) #
# define the parameter as constance
pars <- c(beta =  0.22763126, gamma = 0.03032535, N = N)
sir <- function(time, state, pars) {
  with(as.list(c(state, pars)), {
    dS <- -beta * S * I/N
    dI <- beta * S * I/N - gamma * I
    dR <- gamma * I
    return(list(c(dS, dI, dR)))
  })
}
march_time = seq(1, 31, by = 1)
march_res.sir <- as.data.frame(ode(y = init, times = 1:31, func = sir, parms = pars))
```

Twitter Model Pipeline & Implementation

Download

Spark SQL

Process

APACHE Spark ML Pipelines + John Snow LABS NLP

Transform

Store

Pearson Correlation Coefficient = $\rho(x,y) = \frac{\sum[(xi - x) * (yi - y)]}{\sigma x * \sigma y}$

Calculate

Final Model Implementation

- Spark SQL + Apache Spark ML Pipelines
  - Spark SQL to combine all data
  - Spark ML to train a Linear Regression model

```
df_cases = df_cases.select(
        'date',
        F.col('numconf').cast('Long'),
        F.lit(1).alias('temp'))

df_sir = df_sir.select(
        'date',
        F.col('predict_infection').cast('Float'),)

df_words = df_words.select('date', *most_corr_cols)
```

```
window = Window.partitionBy('temp').orderBy('date')

data = df_words.join(df_cases, on='date', how='right')\
            .join(df_sir, on='date', how='left')\
            .select(
                'date',
                F.date_add('date', 4).alias('prediction_date'),
                F.lead('numconf', 4).over(window).alias('label'),
                'numconf',
                F.lead(F.col('predict_infection').alias('baseline_prediction'),4)\
                        .over(window).alias('sir_prediction'),
                *[F.col(c).cast('Long') for c in most_corr_cols]
            )
```

```
lr = LinearRegression(featuresCol='features', labelCol='label',
                      maxIter=1000, regParam=0.5)
model = lr.fit(train_data)
```

```
prediction_train = model.transform(train_data)
prediction_test = model.transform(test_data)
prediction_full = model.transform(full_data)
```

```
evaluator = RegressionEvaluator(labelCol="label", predictionCol="prediction",
                                metricName="rmse")
```

```
rmse_train = evaluator.evaluate(prediction_train)
rmse_test = evaluator.evaluate(prediction_test)
print("Root Mean Squared Error (RMSE) on training data = %g" % rmse_train)
print("Root Mean Squared Error (RMSE) on test data = %g" % rmse_test)
```

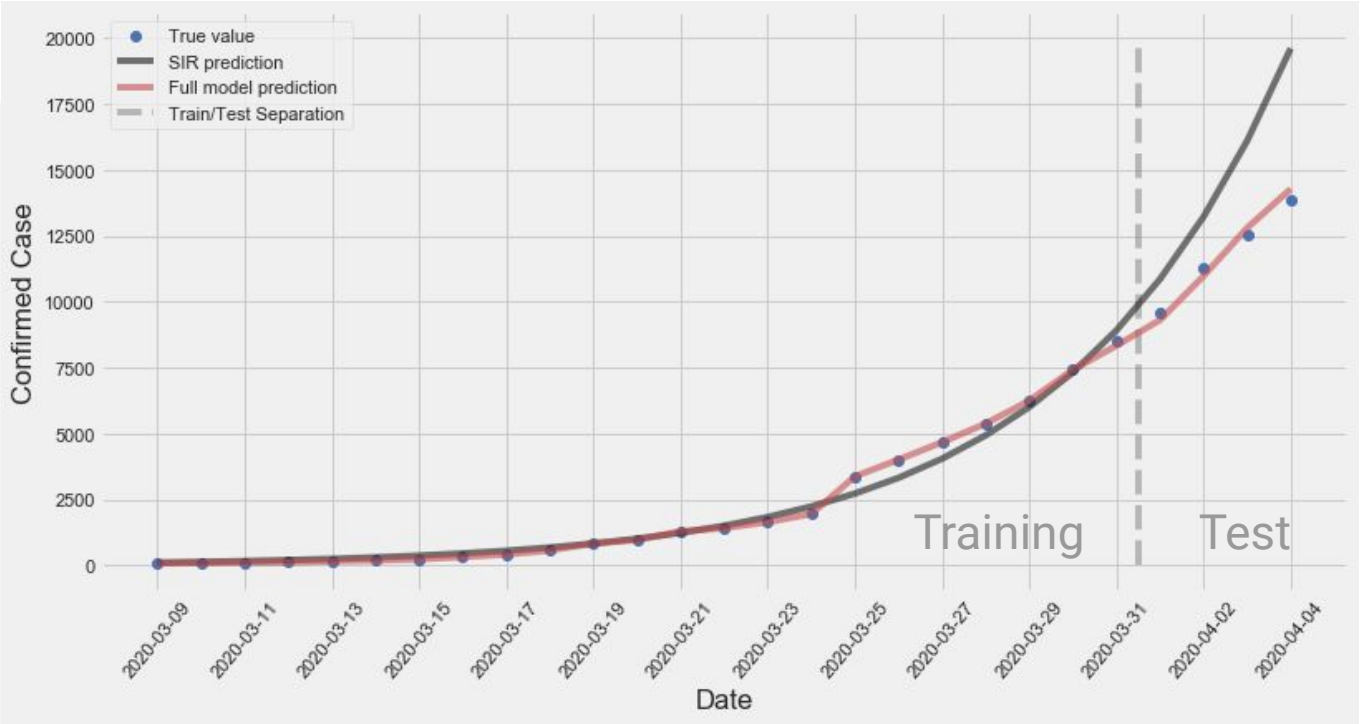
```
Root Mean Squared Error (RMSE) on training data = 5.24159
Root Mean Squared Error (RMSE) on test data = 317.821
```

Result

Predicted Results Comparison

	April 01	April 02	April 03	April 04
SIR (baseline)	10863	13228	16107	19611
Combined SIR + Twitter	9237	10932	12951	14598
True Data	**9595**	**11268**	**12519**	**13882**

Let's Draw Them Together



How Good is Our Model ?

	RMSE		MAPE	
Data	**Train**	**Test**	**Train**	**Test**
SIR (baseline)	290.425	3575.761	25.892%	**25.134%**
Combined SIR + Twitter	78.339	484.970	3.238%	**3.830%**

RMSE: Root Mean Squared Error
MAPE: Mean Absolute Percentage Error

Some Other Insights

- The model is good for prediction window of 1-7 days
- 4th-day prediction is the best
- Possible reasons:
  - Social media has short-term effects
  - A delay in receiving COVID-19 test results
  - The baseline model is not good enough
- Advice: STAY AT HOME!