42186
Model-based machine learning
Project Description

Antek Skrobisz, s213612
Oliver Sande, s174032
Oliver Svane Olsen s184299
Piriya Sureshkumar, s184302

March 31, 2023

Project Description

Dataset and Research Question

This project set out to predict the delay time of domestic flights in the US and aims at answering the following questions:

- How can the delay of aircraft be predicted?
- How much does weather impact the delay time of the aircrafts?
- What limitations are there in the data and what strategies can be used to overcome them?

The dataset used is part of the Reporting Carrier On-Time Performance Dataset that contains information on domestic flights in the US reported to the United States Bureau of Transportation Statistics from January 2018 to July 2022. The dataset is available for download at Kaggle. Each flight is described in the dataset with basic information such as time, departure time, arrival time at the airport, the number of minutes the time was delayed, and some information about the reason the flight was delayed. In Table 1 the relevant features that will be used in the project are presented. The data will be enriched with weather features from Meteostat Developers API with information about wind speed, wind direction, temperature, precipitation, and visibility. Initially, the modeling will be limited to a specific airport, later results from different airports will be compared.

Feature Name	**Feature Description**
FlightDate	Departure date
Airline	Airline
Flight_Number_Marketing_Airline	Flight number
Origin	Origin of flight trip
Dest	Destination of flight trip
Cancelled	If the trip got cancelled
Diverted	If the flight was rescheduled to another destination
CRSDepTime	Time the flight is schedule to departure
DepTime	Actual departure time
DepDelayMinutes	Number of minutes the flight is delayed
OriginCityName	Origin city name
OriginStateName	Origin state name
DestCityName	Destination city name
DestStateName	Destination state name
TaxiOut	Time of the aircrafts movment on the ground at departure
TaxiIn	Time of the aricrafts movement on the ground at arrival
CRSArrTime	Time the flight is schedule to arrive
ArrTime	Actual arrival time
ArrDelayMinutes	Minutes the flight is delayed at arrival

Table 1: Relvevant features in the Reporting Carrier On-Time Performance Dataset for this project.

Drafts of Models

At least two different models will be tried out, a linear regression and a AR(1) model. For the linear regression model, all the timestamp related features are not included. The rest of the variables are treated as input variables and the target variable is as mentioned the arrival delay time. The generative process for this model is:

1. Draw coefficients $\beta \sim \mathcal{N}(\boldsymbol{\beta} \mid \mathbf{0}, \lambda \mathbf{I})$
2. For each feature vector $\mathbf{x}_n$
   (a) Draw target $y_n \sim \mathcal{N}\left(y_n \mid \boldsymbol{\beta}^\top \mathbf{x}_n, \sigma^2\right)$

The PGM for this generative process is seen in Figure 1.

When implementing the AR(1)-model, the input variable is the arrival time timestamp and the target variable is the arrival delay times. The rest of the features including the weather features are treated as external features ($\mathbf{x_t}$).The generative process for this Ar(1) process is:

1. Draw transition coefficients $\boldsymbol{\beta}$ for the hidden states, $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\lambda}_1)$
2. Draw global variance for the observations, $\sigma^2 \sim \text{HalfCauchy}\left(\sigma^2 \mid \lambda_2\right)$
3. Draw global variance for the transitions, $\tau \sim \text{HalfCauchy}\left(\tau \mid \lambda_2\right)$
4. Draw first hidden state, $h_1 \sim \mathcal{N}\left(h_1 \mid \mu_0 + \mathbf{w}\mathbf{x}_1, \tau_0\right)$
5. Draw second hidden state, $h_2 \sim \mathcal{N}\left(h_2 \mid \beta_1 h_1 + \mathbf{w}\mathbf{x_2}, \tau\right)$
6. For each time t $\in (1)$:
   (a) Draw observation noise, $\epsilon_t \sim \mathcal{N}\left(\epsilon_t \mid 0, \sigma^2\right)$
   (b) Draw observation, $y_t \sim \mathcal{N}\left(y_t \mid h_t, \epsilon_t\right)$
7. for each time t $\in (2, \ldots, T)$:
   (a) Draw transition noise, $r_t \sim \mathcal{N}\left(r_t \mid 0, \tau\right)$
   (b) Draw transition, $h_t \sim \mathcal{N}\left(h_t \mid \beta_1 \cdot h_{t-1} + \mathbf{w}\mathbf{x_t}, r_t\right)$
   (c) Draw observation noise, $\epsilon_t \sim \mathcal{N}\left(\epsilon_t \mid 0, \sigma^2\right)$
   (d) Draw observation, $y_t \sim \mathcal{N}\left(y_t \mid h_t, \epsilon_t\right)$

The PGM for this generative process is also seen in Figure 1.

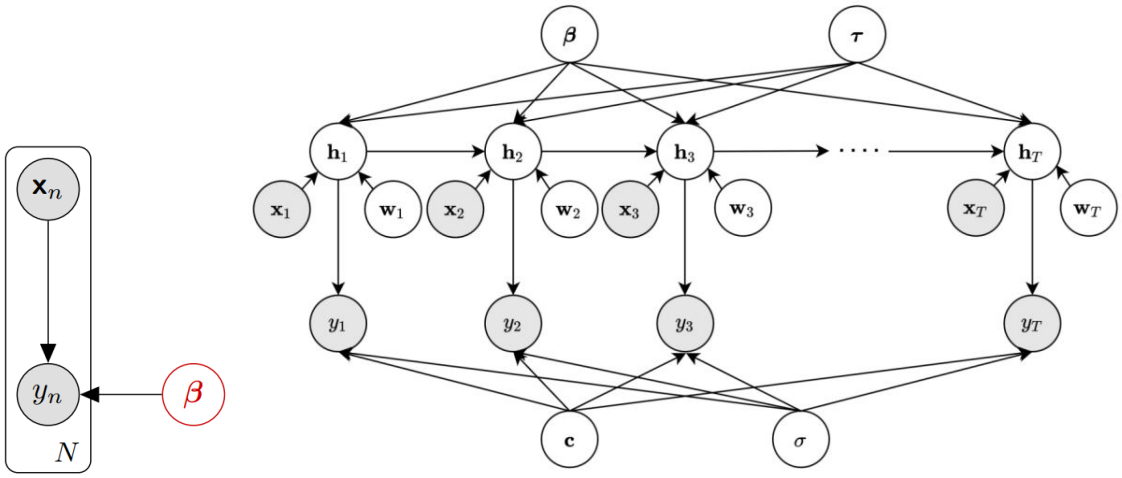


Figure 1: PGMs of the linear regression model and the AR(1) model.

notebook

March 31, 2023

1 MBML 2023

1.1 INIT

1.1.1 Load Packages

[ ]:
```
import pandas as pd
import kaggle
import os
import shutil
import requests
import urllib
from urllib.request import urlopen, urlretrieve
from io import BytesIO
from zipfile import ZipFile
import matplotlib.pyplot as plt
import plotly.express as px
from IPython.display import Image
import numpy as np


from src.data import extract, load, transform
```

1.1.2 Set Flags

[ ]:
```
pd.set_option('display.max_colwidth', None)
pd.set_option('display.max_columns', None)

DATA_DIR = "data/"
```

1.2 Data

1.2.1 Extract Data

[ ]:
```
# Download Fligt Delay Dataset form Kaggle
kaggle.api.authenticate()
kaggle.api.dataset_download_files(
    "robikscube/flight-delay-dataset-20182022",
```

```
    path=DATA_DIR,
    unzip=True,
)
for filename in os.listdir(DATA_DIR):
    f = os.path.join(DATA_DIR, filename)
    if f.endswith(".parquet") or filename == "Airlines.csv":
        pass
    else:
        if os.path.isfile(f):
            os.remove(f)
        else:
            shutil.rmtree(f)
```

[ ]:
```
# Download Location of airports
urlretrieve(
    "https://raw.githubusercontent.com/lxndrblz/Airports/main/airports.csv",
    DATA_DIR + "airports.csv"
    )
```

1.2.2 Transform Data

1.2.3 Load Data

[ ]:
```
main_df = extract.combine_parquet(data_path = "data/")
main_df['count'] = 1
airport_df = pd.read_csv('data/airports.csv')
airline_df = pd.read_csv('data/Airlines.csv')
```

1.2.4 Define display options for later export to pdf

[ ]:
```
pd.set_option('display.notebook_repr_html', True)

def _repr_latex_(self):
    return "\centering{%s}" % self.to_latex()

pd.DataFrame._repr_latex_ = _repr_latex_  # monkey patch pandas DataFrame
```

[ ]:
```
# silence future warnings
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

[ ]:
```
# set max number of rows of dataframe
pd.set_option('display.max_rows', 80)
```

1.2.5 Describe Data

Display two first rows of data

[ ]:
```
# transposing dataframe and displaying in two different cells for compatability␣
↪with pdf conversion
pd.DataFrame(main_df.iloc[:2,:30].T)
```

[ ]:

	0	1
FlightDate	2018-01-23 00:00:00	2018-01-24 00:00:00
Airline	Endeavor Air Inc.	Endeavor Air Inc.
Origin	ABY	ABY
Dest	ATL	ATL
Cancelled	False	False
Diverted	False	False
CRSDepTime	1202	1202
DepTime	1157.0	1157.0
DepDelayMinutes	0.0	0.0
DepDelay	-5.0	-5.0
ArrTime	1256.0	1258.0
ArrDelayMinutes	0.0	0.0
AirTime	38.0	36.0
CRSElapsedTime	62.0	62.0
ActualElapsedTime	59.0	61.0
Distance	145.0	145.0
Year	2018	2018
Quarter	1	1
Month	1	1
DayofMonth	23	24
DayOfWeek	2	3
Marketing_Airline_Network	DL	DL
Operated_or_Branded_Code_Share_Partners	DL_CODESHARE	DL_CODESHARE
DOT_ID_Marketing_Airline	19790	19790
IATA_Code_Marketing_Airline	DL	DL
Flight_Number_Marketing_Airline	3298	3298
Operating_Airline	9E	9E
DOT_ID_Operating_Airline	20363	20363
IATA_Code_Operating_Airline	9E	9E
Tail_Number	N8928A	N800AY

[ ]:
```
pd.DataFrame(main_df.iloc[:2,30:].T)
```

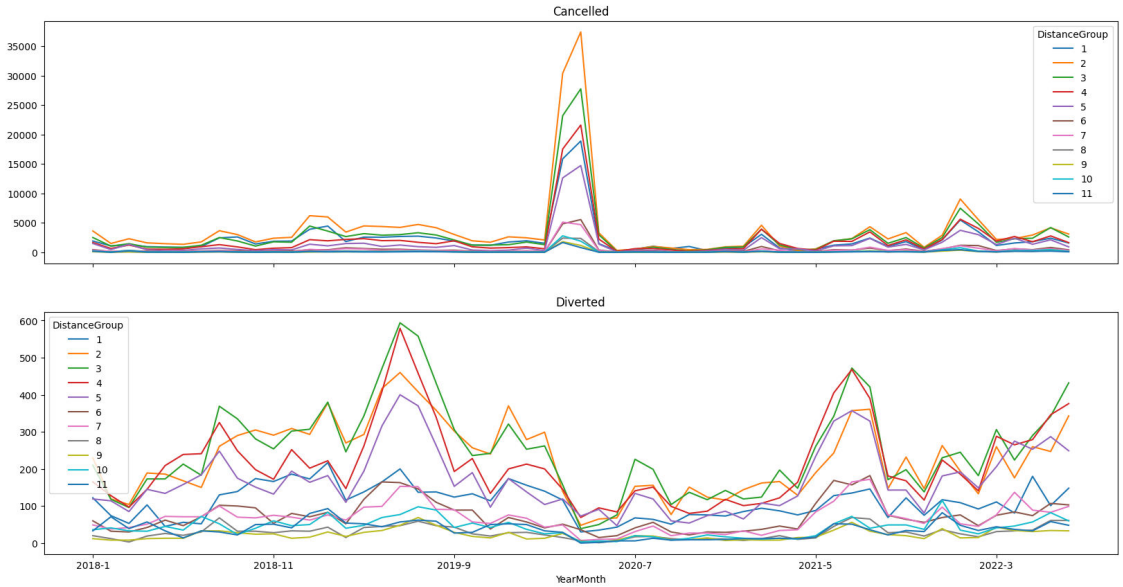
[ ]:

	0	1
Flight_Number_Operating_Airline	3298	3298
OriginAirportID	10146	10146
OriginAirportSeqID	1014602	1014602
OriginCityMarketID	30146	30146
OriginCityName	Albany, GA	Albany, GA
OriginState	GA	GA
OriginStateFips	13	13
OriginStateName	Georgia	Georgia
OriginWac	34	34
DestAirportID	10397	10397
DestAirportSeqID	1039707	1039707
DestCityMarketID	30397	30397
DestCityName	Atlanta, GA	Atlanta, GA
DestState	GA	GA
DestStateFips	13	13
DestStateName	Georgia	Georgia
DestWac	34	34
DepDel15	0.0	0.0
DepartureDelayGroups	-1.0	-1.0
DepTimeBlk	1200-1259	1200-1259
TaxiOut	14.0	13.0
WheelsOff	1211.0	1210.0
WheelsOn	1249.0	1246.0
TaxiIn	7.0	12.0
CRSArrTime	1304	1304
ArrDelay	-8.0	-6.0
ArrDel15	0.0	0.0
ArrivalDelayGroups	-1.0	-1.0
ArrTimeBlk	1300-1359	1300-1359
DistanceGroup	1	1
DivAirportLandings	0.0	0.0
count	1	1

[ ]:
```
# Plot cancellations and Diverted by length of flight
g_main_df = main_df.groupby(["Year","Month","DistanceGroup"]).sum().unstack().
  ↪reset_index()
g_main_df["YearMonth"] = g_main_df["Year"].astype(str)+"-"+g_main_df["Month"].
  ↪astype(str)

subplots = 2
fig, ax = plt.subplots(subplots,1,figsize = (20,10))
prediction_cols = ["Cancelled","Diverted"]
for i in range(subplots):
```

```
    g_main_df[[prediction_cols[i],"YearMonth"]].
↪plot(x="YearMonth",y=prediction_cols[i],kind = "line", legend = True,ax =␣
↪ax[i],sharex = True,title = prediction_cols[i])
```

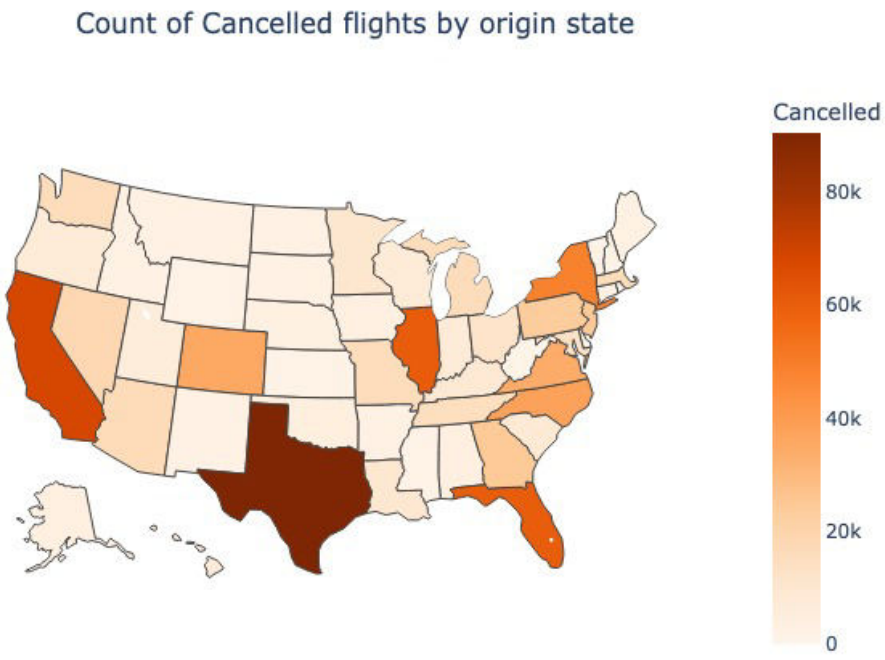


[ ]:
```
g_main_df = main_df.groupby(["OriginState"]).sum()
```

[ ]:
```
# plot a choropleth with color range by count per state
fig = px.choropleth(g_main_df['Cancelled'].reset_index(),
                    locations='OriginState',
                    locationmode="USA-states",
                    scope="usa",
                    color="Cancelled",
                    color_continuous_scale="Oranges",
                   )
# center the title
fig.update_layout(title_text='Count of Cancelled flights by origin state',␣
 ↪title_x=0.5)

# export plot to image to compatability with jupyter conversion
#fig.show()
im = fig.to_image("jpeg")
Image(im)
```

[ ]:

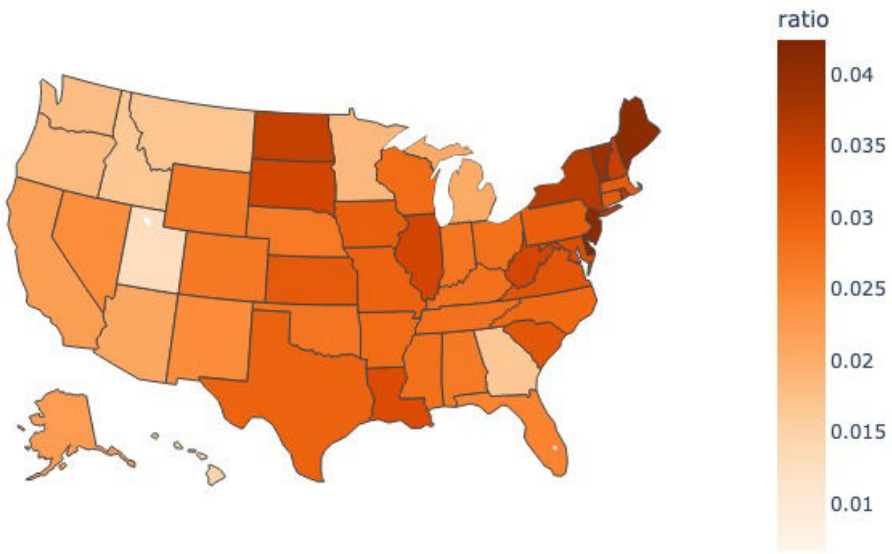


[ ]:
```
state_df = g_main_df[['Cancelled','count']].reset_index()
state_df["ratio"] = state_df['Cancelled']/state_df['count']

# plot a choropleth with color range by count per state
fig = px.choropleth(state_df,
                    locations='OriginState',
                    locationmode="USA-states",
                    scope="usa",
                    color="ratio",
                    color_continuous_scale="Oranges",
                   )
# center the title
fig.update_layout(title_text='Ratio of Cancelled flights by origin state',␣
 ↪title_x=0.5)

# export plot to image to compatability with pdf conversion
#fig.show()
im = fig.to_image("jpeg")
Image(im)
```

[ ]:


Ratio of Cancelled flights by origin state
ratio
0.04
0.035
0.03
0.025
0.02
0.015
0.01