JFK Data Analysis - Phase 4

Analysis Overview

This phase delivers predictive modeling using LASSO, GAM, and Random Forests, with cross-validation and feature evaluation.

Set Up

Reading the data, loading required libraries, and setting the seed.

```r
library(readxl)
data<-read_xlsx("all_data.xlsx")

library(dplyr)
library(lubridate)
library(fastDummies)
library(glmnet)
library(mgcv)
library(randomForest)
library(Metrics)
set.seed(1542)
```

Data transformation

The variables **id**,**flight_code**,**destination** were eliminated since they actually behave as identifiers. The raw original **timestamp** variable was converted to useful **hour** and **weekday**. Categorical columns that will be used were transformed to factors for safety.

```r
data<-data %>%
        mutate(timestamp=ymd_hms(timestamp),
               hour=hour(timestamp),
               weekday=wday(timestamp,label = TRUE),
               weekday=factor(weekday,ordered = FALSE),
               carrier=factor(carrier),
               wind=factor(wind)) %>%
        select(-c(id,timestamp,flight_code,
                  destination,taxi_10tile))
```

Functions

For efficiency and reproducibility two functions were created:

- **evaluation** which given the model and the test_data it returns basic metrics
- **lasso** which given x and y returns LASSO coefficient table and basic metrics.

```r
evaluation<-function(model,test_data){
        f_pred<-predict(model,newdata = test_data)
        f_mae<-mae(test_data$taxi_out,f_pred)
        f_rmse<-rmse(test_data$taxi_out,f_pred)
        f_ss_res<-sum((test_data$taxi_out-f_pred)^2)
        f_ss_tot<-sum((test_data$taxi_out-mean(test_data$taxi_out))^2)
        f_r2<-1-f_ss_res/f_ss_tot
        return(list(mae=f_mae,rmse=f_rmse,r2=f_r2))
}

lasso<-function(x,y){
        n<-nrow(x)
        set.seed(1542)
        train_idx<-sample(seq_len(n),size = 0.8*n)
        x_train<-x[train_idx,]
        y_train<-y[train_idx]
        x_test<-x[-train_idx,]
        y_test<-y[-train_idx]
        cv_model<-cv.glmnet(x_train,y_train)
        lambda<-cv_model$lambda.min


        model<-glmnet(x_train,y_train,lambda=lambda)

        f_pred<-predict(model,newx = x_test,s=lambda)
        f_mae<-mae(y_test,f_pred)
        f_rmse<-rmse(y_test,f_pred)
        f_ss_res<-sum((y_test-f_pred)^2)
        f_ss_tot<-sum((y_test-mean(y_test))^2)
        f_r2<-1-f_ss_res/f_ss_tot

        coeftable<-data.frame(variabes=row.names(coef(model)),
                              values=as.numeric(round(coef(model),4)))
        coeftable<-coeftable[order(abs(coeftable$values),
                                              decreasing = TRUE),]
        return(list(mae=f_mae,rmse=f_rmse,r2=f_r2,coef=head(coeftable,6)))

}
```

Lasso

The key focus is the LASSO coefficient table, with each run limited to one or two categorical factors to prevent overloading and reveal where meaningful variable weight appears.

```r
y1<-data$taxi_out
lasso_data1<- data %>%
        dummy_cols(select_columns = "weekday") %>%
        select(-c(carrier,wind,taxi_out))
x1<-model.matrix(~.,data = lasso_data1)
lasso(x1,y1)
```

```
## $mae
## [1] 5.285366
##
## $rmse
## [1] 6.640551
##
## $r2
## [1] 0.07195242
##
## $coef
##       variabes  values
## 1  (Intercept) 35.8855
## 15  weekdayTue -0.6391
## 12    pressure -0.5702
## 18  weekdayFri -0.5457
## 16  weekdayWed -0.5274
## 17  weekdayThu  0.4670
```

```r
y2<-data$taxi_out
lasso_data2<-data %>%
        dummy_cols(select_columns = "carrier") %>%
        select(-c(wind,weekday,taxi_out))
x2<-model.matrix(~.,data = lasso_data2)
lasso(x2,y2)
```

```
## $mae
## [1] 5.205609
##
## $rmse
## [1] 6.576505
##
## $r2
## [1] 0.08976735
##
## $coef
##       variabes  values
## 1  (Intercept) 36.0369
## 4    carrierAS  3.9640
## 5    carrierB6 -1.3453
## 7    carrierHA -1.2101
## 20    pressure -0.5542
## 27  carrier_HA -0.5308
```

```r
y3<-data$taxi_out
lasso_data3<-data%>%
        dummy_cols(select_columns = "wind") %>%
        select(-c(carrier,weekday,taxi_out))
x3<-model.matrix(~.,data=lasso_data3)
lasso(x3,y3)
```

```
## $mae
## [1] 5.2526
##
## $rmse
## [1] 6.619856
##
## $r2
## [1] 0.07772764
##
## $coef
##       variabes  values
## 1  (Intercept) 47.9262
## 11     windENE  1.5738
## 29    pressure -0.9722
## 16     windNNW -0.9020
## 12     windESE  0.8842
## 26     windWSW -0.7812
```

```
y4<-data$taxi_out
lasso_data4<-data%>%
        dummy_cols(select_columns = c("wind","carrier")) %>%
        select(-c(weekday,taxi_out))
x4<-model.matrix(~.,data=lasso_data4)
lasso(x4,y4)
```

```
## $mae
## [1] 5.161294
##
## $rmse
## [1] 6.542623
##
## $r2
## [1] 0.09912215
##
## $coef
##       variabes  values
## 1  (Intercept) 48.6340
## 4    carrierAS  3.9576
## 19     windENE  1.4590
## 5    carrierB6 -1.3364
## 7    carrierHA -1.2343
## 37    pressure -0.9690
```

```
y5<-data$taxi_out
lasso_data5<-data %>%
        select(departures,wind)
x5<-model.matrix(~.,data = lasso_data5)
lasso(x5,y5)
```

```
## $mae
## [1] 5.337542
##
## $rmse
## [1] 6.739177
##
## $r2
## [1] 0.04418083
```

```
##
## $coef
##       variabes  values
## 1  (Intercept) 16.6175
## 5      windENE  2.1540
## 17     windVAR -1.1943
## 20     windWSW -1.1349
## 8       windNE  0.8932
## 13      windSE -0.8437
```

GAM

To capture potential non-linear relationships, several GAM combinations were tested.

```
n<-nrow(data)
train_idx<-sample(seq_len(n),size = 0.8*n)
gam_data1<- data %>%
        select(dep_delay,departures,arrivals,temperature,wind,taxi_out)
train_data1<-gam_data1[train_idx,]
test_data1<-gam_data1[-train_idx,]
gam_model1<-gam(taxi_out~s(departures) +wind,data = train_data1,
                method = "REML")
evaluation(gam_model1,test_data1)
```

```
## $mae
## [1] 5.294927
##
## $rmse
## [1] 6.647421
##
## $r2
## [1] 0.05170853
```

```
gam_model2<-gam(taxi_out~s(dep_delay)+s(departures)+s(arrivals)
                +s(temperature)+wind, data = train_data1,
                method = "REML")
evaluation(gam_model2,test_data1)
```

```
## $mae
## [1] 5.25571
##
## $rmse
## [1] 6.596801
##
## $r2
## [1] 0.06609611
```

```
gam_data2<- data %>%
        select(dep_delay,departures,arrivals,temperature,carrier,taxi_out)
train_data2<-gam_data2[train_idx,]
test_data2<-gam_data2[-train_idx,]
```

```r
gam_model3<-gam(taxi_out~s(departures) +carrier+s(dep_delay)
                +s(arrivals)+s(temperature),data = train_data2,
                method = "REML")
evaluation(gam_model3,test_data2)
```

```
## $mae
## [1] 5.21823
##
## $rmse
## [1] 6.566307
##
## $r2
## [1] 0.07471013
```

Random Forests

In order to detect more complex interactions, Random Forests were used.

```r
rf1_train<-data[train_idx,] %>%
        select(c(departures,wind,taxi_out))
rf1_test<-data[-train_idx,] %>%
        select(c(departures,wind,taxi_out))
rf1<-randomForest(taxi_out~.,
                  data=rf1_train,
                  mtry=2,
                  maxnodes=30)
evaluation(rf1,rf1_test)
```

```
## $mae
## [1] 5.287931
##
## $rmse
## [1] 6.638225
##
## $r2
## [1] 0.05433067
```

```r
rf2_train<-data[train_idx,]
rf2_test<-data[-train_idx,]
rf2<-randomForest(taxi_out~.,
                  data=rf2_train,
                  mtry=6,
                  importance=TRUE,
                  maxnodes=500,
                  nodesize=150)
evaluation(rf2,rf2_test)
```

```
## $mae
## [1] 4.97365
##
## $rmse
## [1] 6.245785
##
## $r2
## [1] 0.1628381
```

Top Model Result

The Random Forest model including all usable variables delivers the best results:

```
evaluation(rf2,rf2_test)
```

```
## $mae
## [1] 4.97365
##
## $rmse
## [1] 6.245785
##
## $r2
## [1] 0.1628381
```

```
imp<- importance(rf2)
imp_table<-data.frame(variable=rownames(imp),importance=imp[,1],
                      row.names = NULL)
imp_table <- imp_table[order(imp_table$importance, decreasing = TRUE), ]
imp_table
```

```
##       variable importance
## 1      carrier   80.07580
## 4   departures   74.57532
## 9         wind   60.86407
## 2    dep_delay   42.39487
## 12    pressure   40.71500
## 6  temperature   40.09590
## 14     weekday   39.51985
## 11   wind_gust   33.06490
## 8     humidity   32.98834
## 5     arrivals   32.11200
## 13        hour   30.35654
## 7    dew_point   29.48188
## 10  wind_speed   28.25649
## 3     distance   21.59177
```

Summary

With these foundations in place, stakeholders now have a clearer view of the main drivers of taxi_out time and a starting point for further development of operational improvements at JFK. From an analytical perspective, departures appear to play a meaningful role in taxi_out performance and represent a sensible direction for further investigation.