

NYC Flights

Shreya Kaul

The NYC Flights dataset gives the data on all the flights that departed New York's airports (i.e. JFK, LGA or EWR) in 2013. The combined data about the flights and the weather gives the information about the scheduled and actual date and time of departure, origin and destination airports, scheduled and actual date and time of arrival, and time difference between actual and scheduled time of departure and arrival. It also gives the data about the weather condition at that time.

Number of observation $n=336776$

Number of variables $p=29$

Classes created for k-NN - Class 1: If the flight was on time Class 2: If the flight arrived late than the scheduled time.

Features in a dataset (columns) are the variables which give us information based on which analysis can be performed.

Two features used for this model, p_1 =wind_speed time and p_2 =distance

For the purpose of plotting, observations with the values 'NA' are removed and 150 observations are sampled randomly.

```
# Load standard libraries
```

```
library(tidyverse)
library(nycflights13)
library(ggplot2)
library(MASS)
library(class)
library(dplyr)
library(gmodels)
```

```
# Joining the flight and weather data
```

```
flights_weather <- left_join(flights, weather)
```

```
## Joining, by = c("year", "month", "day", "origin", "hour", "time_hour")
```

```
# Removing the NA values
```

```
weather_flight <- na.omit(flights_weather)
```

```
summary(weather_flight)
```

##	year	month	day	dep_time	sched_dep_time
##	Min. :2013	Min. : 1.000	Min. : 1.00	Min. : 1	Min. : 500
##	1st Qu.:2013	1st Qu.: 3.000	1st Qu.: 8.00	1st Qu.:1125	1st Qu.:1120
##	Median :2013	Median : 5.000	Median :15.00	Median :1448	Median :1442
##	Mean :2013	Mean : 5.919	Mean :15.36	Mean :1410	Mean :1397
##	3rd Qu.:2013	3rd Qu.: 9.000	3rd Qu.:23.00	3rd Qu.:1717	3rd Qu.:1700

```

## Max. :2013 Max. :12.000 Max. :31.00 Max. :2400 Max. :2359
## dep_delay arr_time sched_arr_time arr_delay
## Min. : -23.00 Min. : 1 Min. : 1 Min. : -74.000
## 1st Qu.: -5.00 1st Qu.:1311 1st Qu.:1325 1st Qu.: -16.000
## Median : -1.00 Median :1640 Median :1645 Median : -4.000
## Mean : 12.29 Mean :1588 Mean :1608 Mean : 7.287
## 3rd Qu.: 11.00 3rd Qu.:1925 3rd Qu.:1921 3rd Qu.: 14.000
## Max. :797.00 Max. :2400 Max. :2359 Max. :783.000
## carrier flight tailnum origin
## Length:72734 Min. : 1 Length:72734 Length:72734
## Class :character 1st Qu.: 605 Class :character Class :character
## Mode :character Median :1552 Mode :character Mode :character
## Mean :2016
## 3rd Qu.:3547
## Max. :6181
## dest air_time distance hour
## Length:72734 Min. : 21.0 Min. : 80 Min. : 5.00
## Class :character 1st Qu.: 83.0 1st Qu.: 502 1st Qu.:11.00
## Mode :character Median :128.0 Median : 828 Median :14.00
## Mean :148.7 Mean :1021 Mean :13.71
## 3rd Qu.:184.0 3rd Qu.:1372 3rd Qu.:17.00
## Max. :695.0 Max. :4983 Max. :23.00
## minute time_hour temp dewp
## Min. : 0.00 Min. :2013-01-01 05:00:00 Min. : 12.02 Min. : -9.94
## 1st Qu.:10.00 1st Qu.:2013-03-14 17:00:00 1st Qu.: 37.94 1st Qu.:17.06
## Median :29.00 Median :2013-05-26 09:00:00 Median : 51.08 Median :28.94
## Mean :26.66 Mean :2013-06-13 17:47:28 Mean : 53.62 Mean :32.50
## 3rd Qu.:45.00 3rd Qu.:2013-09-19 16:00:00 3rd Qu.: 68.00 3rd Qu.:48.92
## Max. :59.00 Max. :2013-12-30 18:00:00 Max. :100.04 Max. :75.02
## humid wind_dir wind_speed wind_gust
## Min. :13.95 Min. : 10.0 Min. : 4.603 Min. :16.11
## 1st Qu.:35.80 1st Qu.:220.0 1st Qu.:12.659 1st Qu.:20.71
## Median :44.16 Median :280.0 Median :16.111 Median :24.17
## Mean :46.75 Mean :253.9 Mean :16.523 Mean :24.91
## 3rd Qu.:54.77 3rd Qu.:310.0 3rd Qu.:19.563 3rd Qu.:27.62
## Max. :96.85 Max. :360.0 Max. :39.127 Max. :66.75
## precip pressure visib
## Min. :0.000000 Min. : 983.8 Min. : 0.120
## 1st Qu.:0.000000 1st Qu.:1010.8 1st Qu.:10.000
## Median :0.000000 Median :1015.4 Median :10.000
## Mean :0.001352 Mean :1015.6 Mean : 9.803
## 3rd Qu.:0.000000 3rd Qu.:1020.4 3rd Qu.:10.000
## Max. :0.530000 Max. :1040.4 Max. :10.000

```

```
str(weather_flight)
```

```

## Classes 'tbl_df', 'tbl' and 'data.frame': 72734 obs. of 28 variables:
## $ year : int 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
## $ month : int 1 1 1 1 1 1 1 1 1 1 1 ...
## $ day : int 1 1 1 1 1 1 1 1 1 1 1 ...
## $ dep_time : int 533 554 557 558 559 600 600 602 602 623 ...
## $ sched_dep_time: int 529 600 600 600 600 600 600 610 605 610 ...
## $ dep_delay : num 4 -6 -3 -2 -1 0 0 -8 -3 13 ...
## $ arr_time : int 850 812 709 753 941 851 837 812 821 920 ...
## $ sched_arr_time: int 830 837 723 745 910 858 825 820 805 915 ...

```

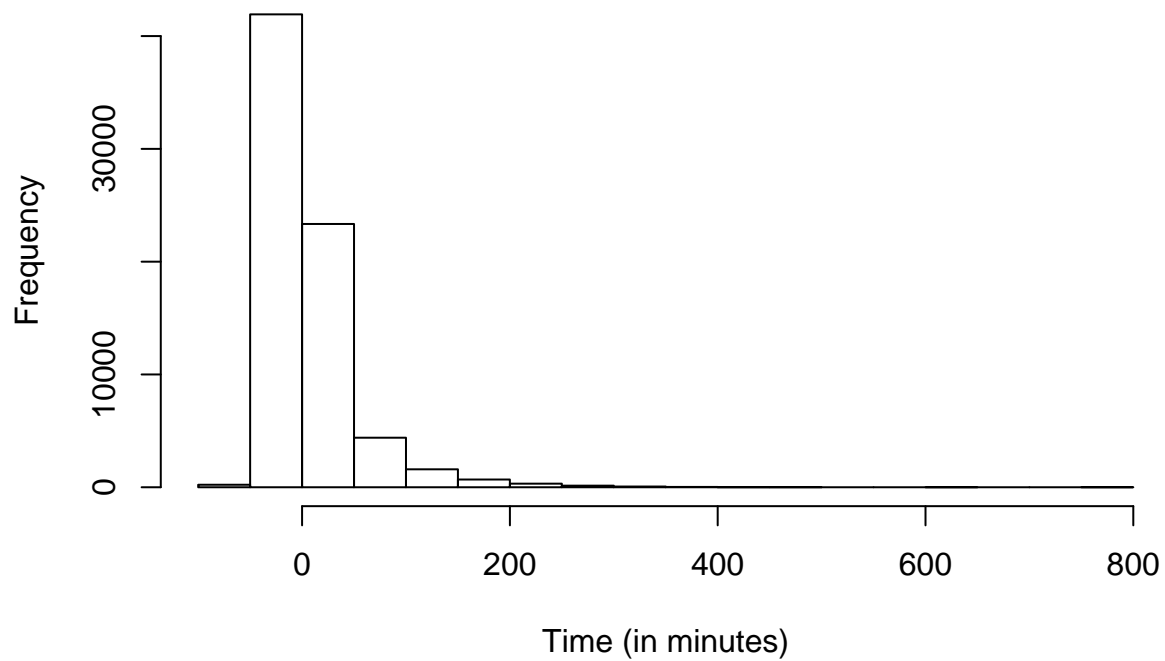
```
## $ arr_delay      : num  20 -25 -14 8 31 -7 12 -8 16 5 ...
## $ carrier        : chr   "UA" "DL" "EV" "AA" ...
## $ flight         : int  1714 461 5708 301 707 371 4650 1919 4401 1837 ...
## $ tailnum        : chr   "N24211" "N668DN" "N829AS" "N3ALAA" ...
## $ origin         : chr   "LGA" "LGA" "LGA" "LGA" ...
## $ dest           : chr   "IAH" "ATL" "IAD" "ORD" ...
## $ air_time       : num   227 116 53 138 257 152 134 170 105 153 ...
## $ distance       : num  1416 762 229 733 1389 ...
## $ hour           : num    5 6 6 6 6 6 6 6 6 6 ...
## $ minute         : num   29 0 0 0 0 0 0 10 5 10 ...
## $ time_hour      : POSIXct, format: "2013-01-01 05:00:00" "2013-01-01 06:00:00" ...
## $ temp           : num   39.9 39.9 39.9 39.9 39.9 ...
## $ dewp           : num   25 25 25 25 25 ...
## $ humid          : num   54.8 54.8 54.8 54.8 54.8 ...
## $ wind_dir       : num   250 260 260 260 260 260 260 260 260 260 ...
## $ wind_speed     : num   15 16.1 16.1 16.1 16.1 ...
## $ wind_gust      : num   21.9 23 23 23 23 ...
## $ precip         : num    0 0 0 0 0 0 0 0 0 0 ...
## $ pressure       : num  1011 1012 1012 1012 1012 ...
## $ visib          : num   10 10 10 10 10 10 10 10 10 10 ...
## - attr(*, "na.action")= 'omit' Named int  1 3 4 6 7 9 11 12 13 14 ...
## ..- attr(*, "names")= chr  "1" "3" "4" "6" ...
```

```
head(weather_flight)
```

```
## # A tibble: 6 x 28
##   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>   <int>         <int>      <dbl>      <int>         <int>
## 1  2013     1     1     533             529         4        850           830
## 2  2013     1     1     554             600        -6        812           837
## 3  2013     1     1     557             600        -3        709           723
## 4  2013     1     1     558             600        -2        753           745
## 5  2013     1     1     559             600        -1        941           910
## 6  2013     1     1     600             600         0        851           858
## # ... with 20 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dtm>, temp <dbl>, dewp <dbl>,
## #   humid <dbl>, wind_dir <dbl>, wind_speed <dbl>, wind_gust <dbl>,
## #   precip <dbl>, pressure <dbl>, visib <dbl>
```

```
hist(weather_flight$arr_delay, main = "Arrival Time Delays", xlab = "Time (in minutes)")
```

Arrival Time Delays

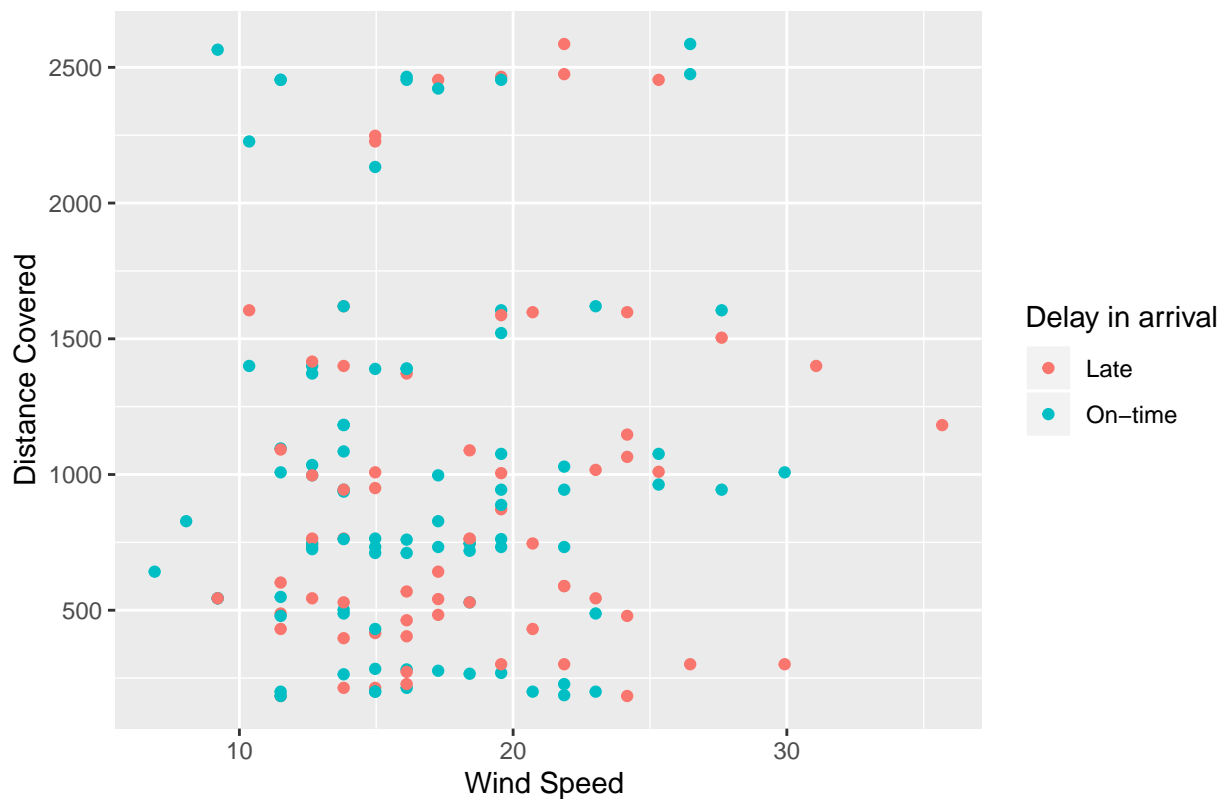


```
# Adding another column in the data frame, 'Class'
weather_flight$Class <- ifelse(weather_flight$arr_delay <=0,"On-time",
                               ifelse
                                 (weather_flight$arr_delay >0,"Late",NA))

# Randomly sampling 150 observations
set.seed(3)
weather_flight_rand <- sample(1:nrow(weather_flight),size = 150, replace = FALSE)
weather_flight_rand_df <- weather_flight[weather_flight_rand,]

# Plotting the graph between distance and wind speed
gg <- ggplot(weather_flight_rand_df, aes(weather_flight_rand_df$wind_speed, weather_flight_rand_df$distance))
print(gg)
```

Delay in arrival of flights based on the wind speed



```
# Creating a function for plotting of KNN for different values of K
func_knn <- function(k){

# Training Data
train <- rbind(weather_flight_rand_df[1:150,c('wind_speed','distance')])

# Test Data
test <- expand.grid(x=seq(min(train[,1]-1), max(train[,1]+1),
                        by=0.5),
                  y=seq(min(train[,2]-1), max(train[,2]+1),
                        by=50))

cl <- factor(c(weather_flight_rand_df$Class[1:150]))

classif <- knn(train=train, test=test, cl = cl, k = k, prob = TRUE)

prob <- attr(classif, "prob")

dataf <- bind_rows(mutate(test,
                          prob=prob,
                          cls='On-time',
                          prob_cls=ifelse(classif==cls,
                                           1, 0)),
                  mutate(test,
                          prob=prob,
                          cls='Late',
                          prob_cls=ifelse(classif==cls,
                                           1, 0)))
```

```

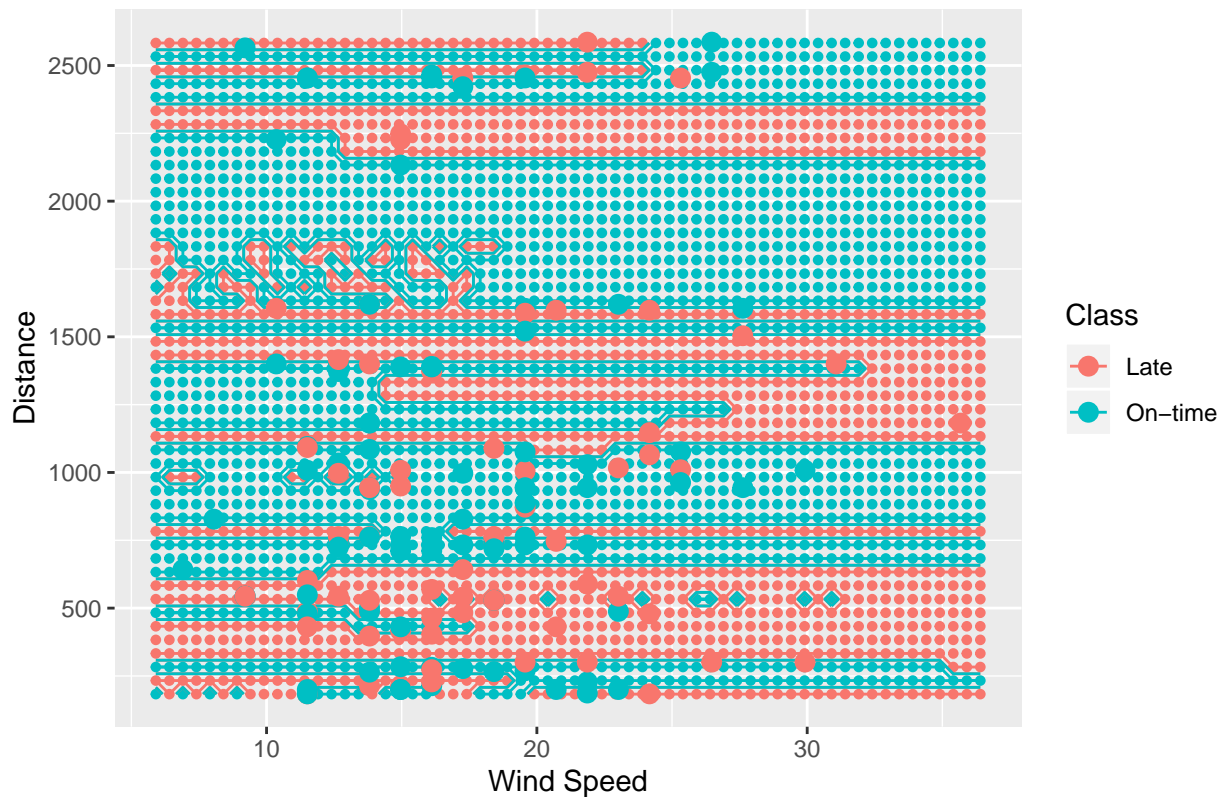
1, 0)))

# Plotting KNN
ggp <- ggplot(dataf) + labs(x='Wind Speed', y='Distance') + ggtitle (paste("Value of K =", as.character
geom_point(aes(x=x, y=y, col=cls),
            data = mutate(test, cls=classif),
            size=1.2) +
geom_contour(aes(x=x, y=y, z=prob_cls, group=cls, color=cls),
            bins=2, data = dataf) +
geom_point(aes(x=train$wind_speed, y=train$distance, col=cls), size=3, data=data.frame(x=train[,1], y=
print(ggp)
}

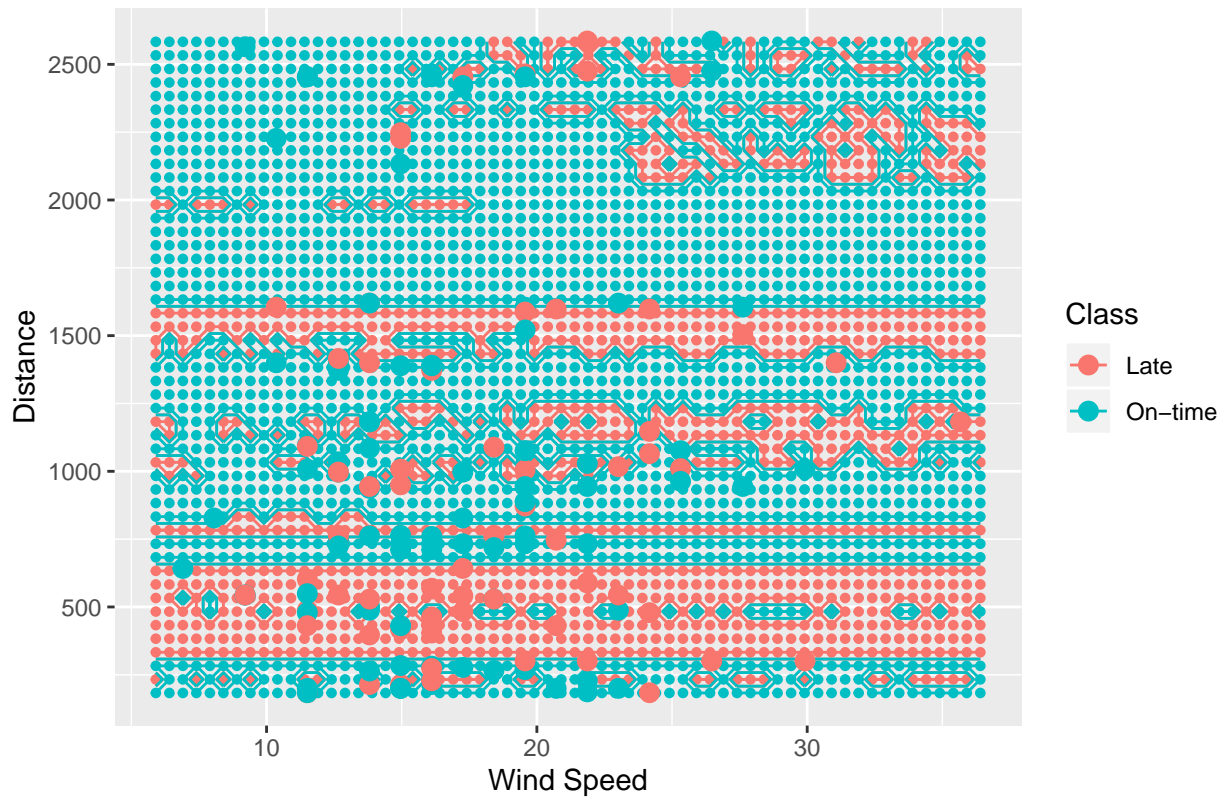
# Calling the function to plot KNN based on multiple values of K
for(i in seq(1, 75, 5)) {
  func_knn(i) }

```

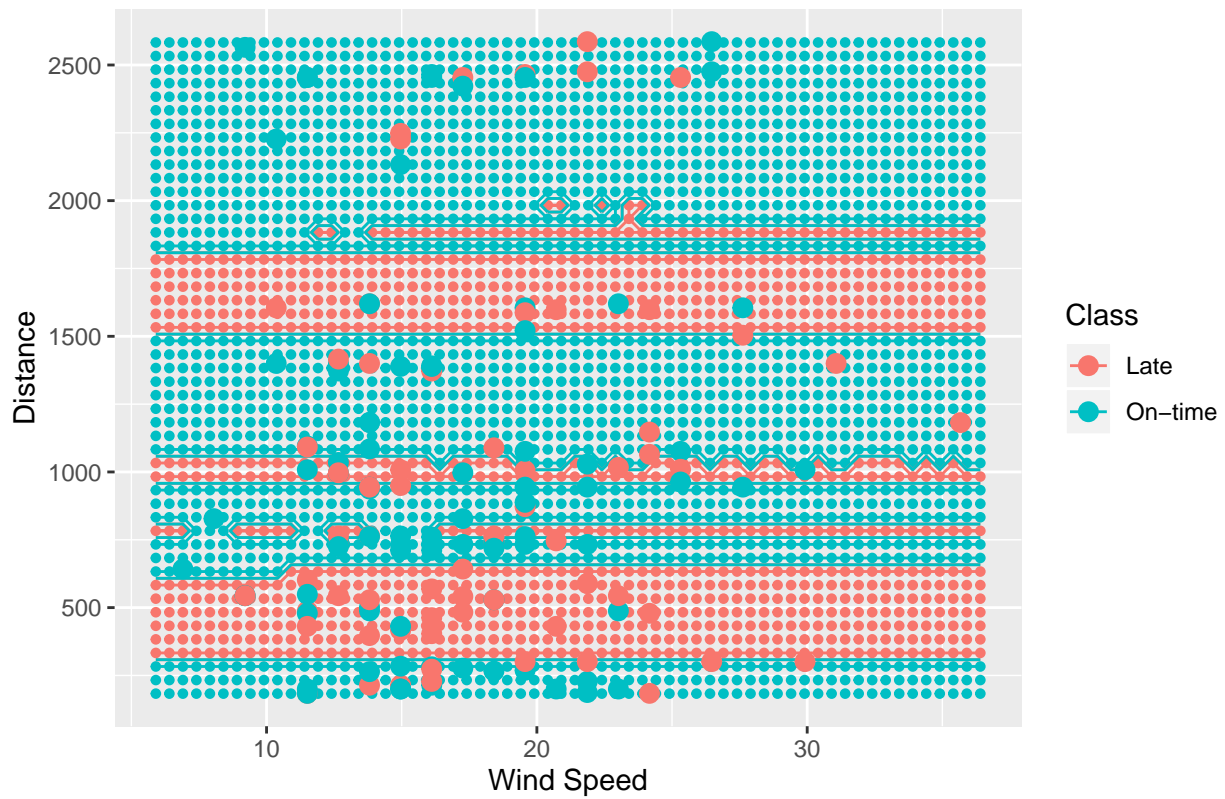
Value of K = 1



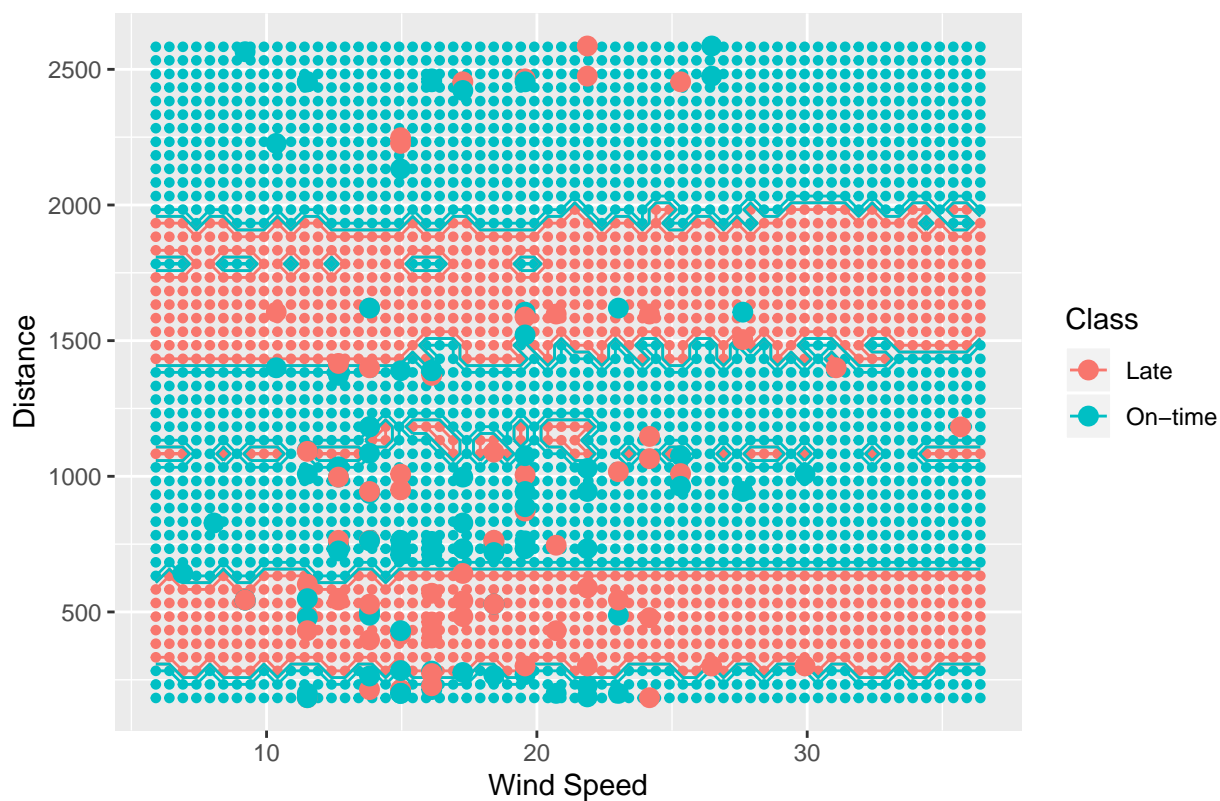
Value of K = 6



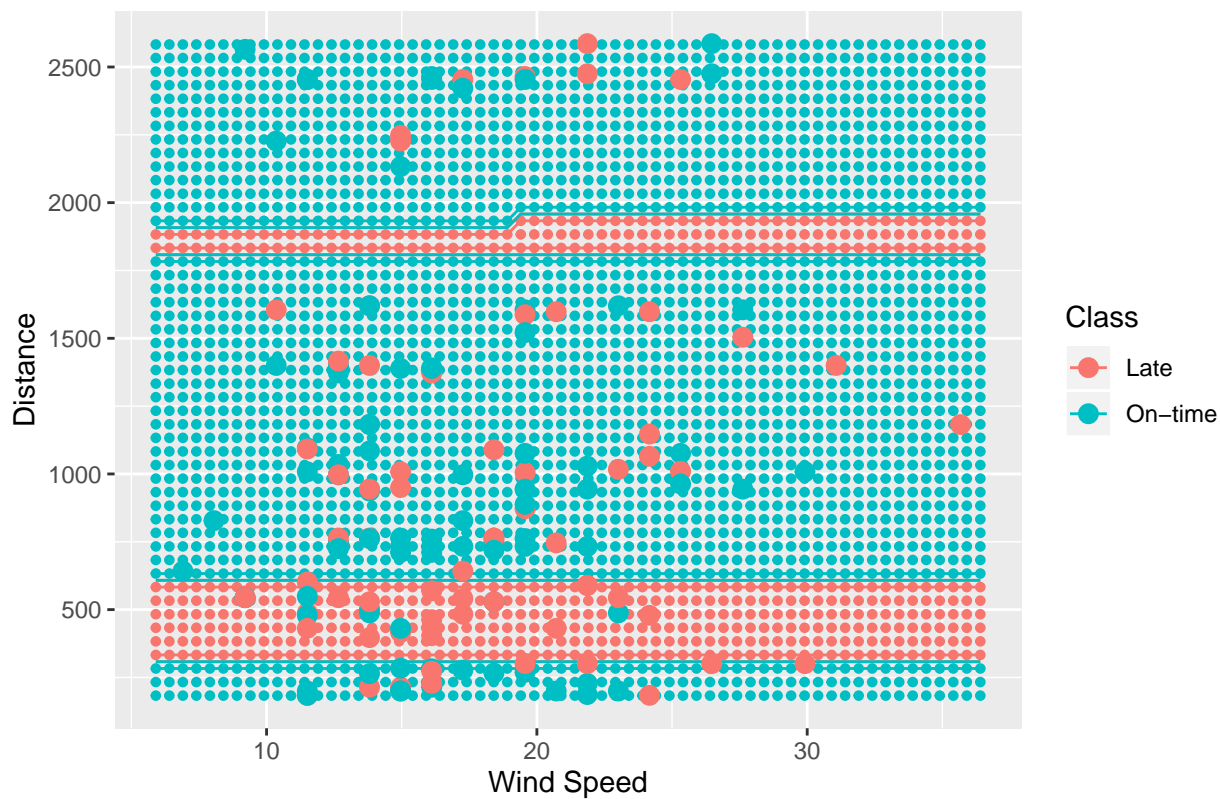
Value of K = 11



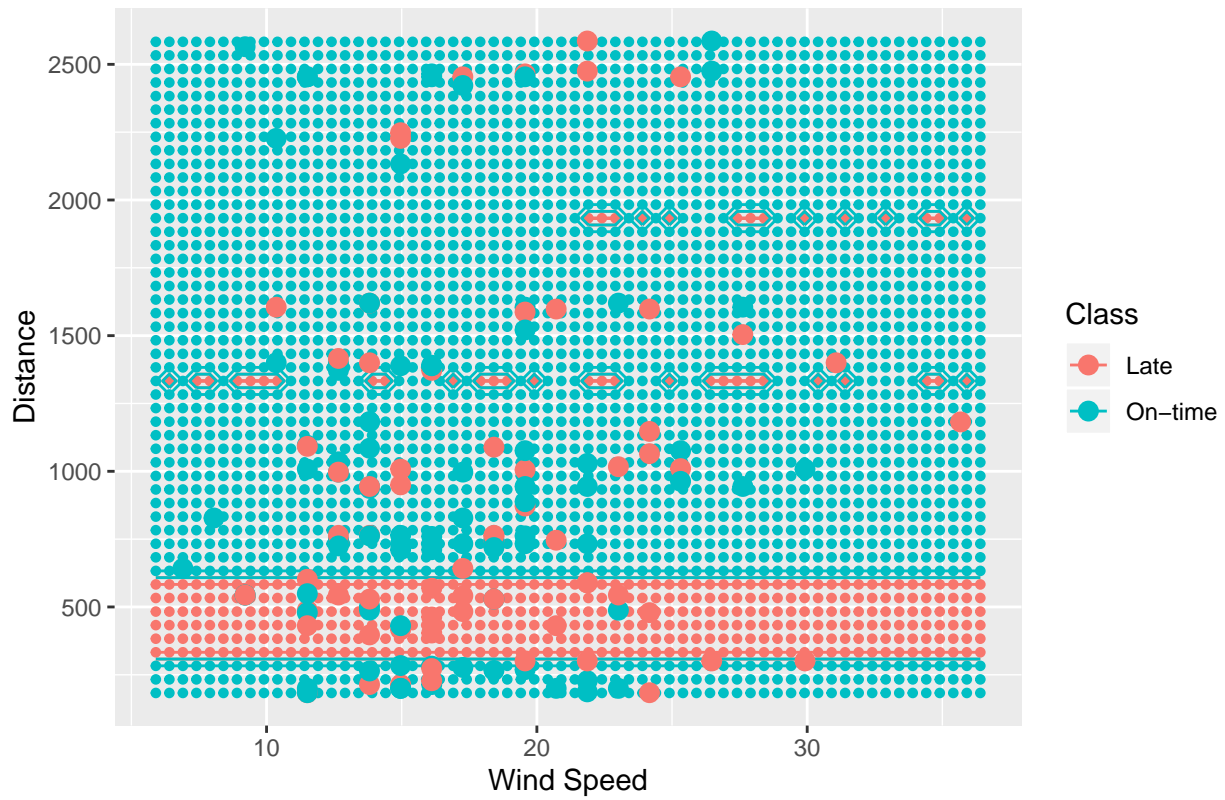
Value of K = 16



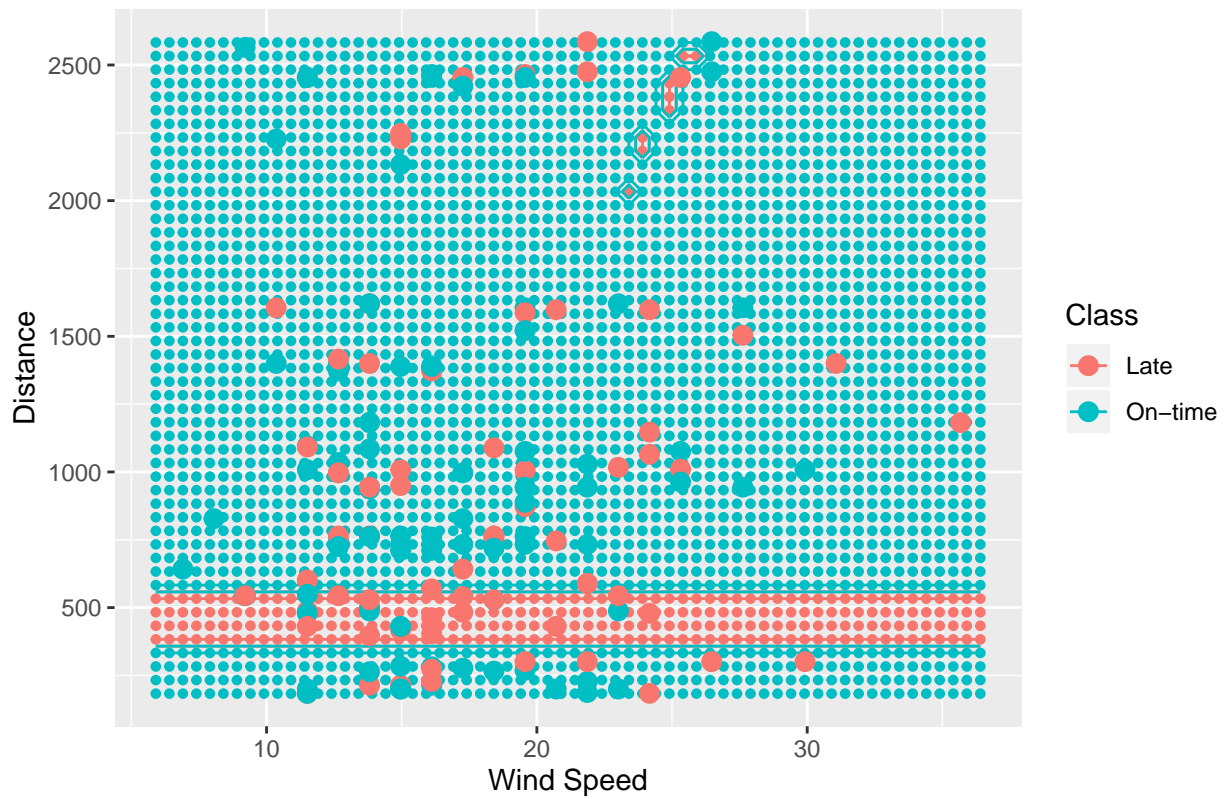
Value of K = 21



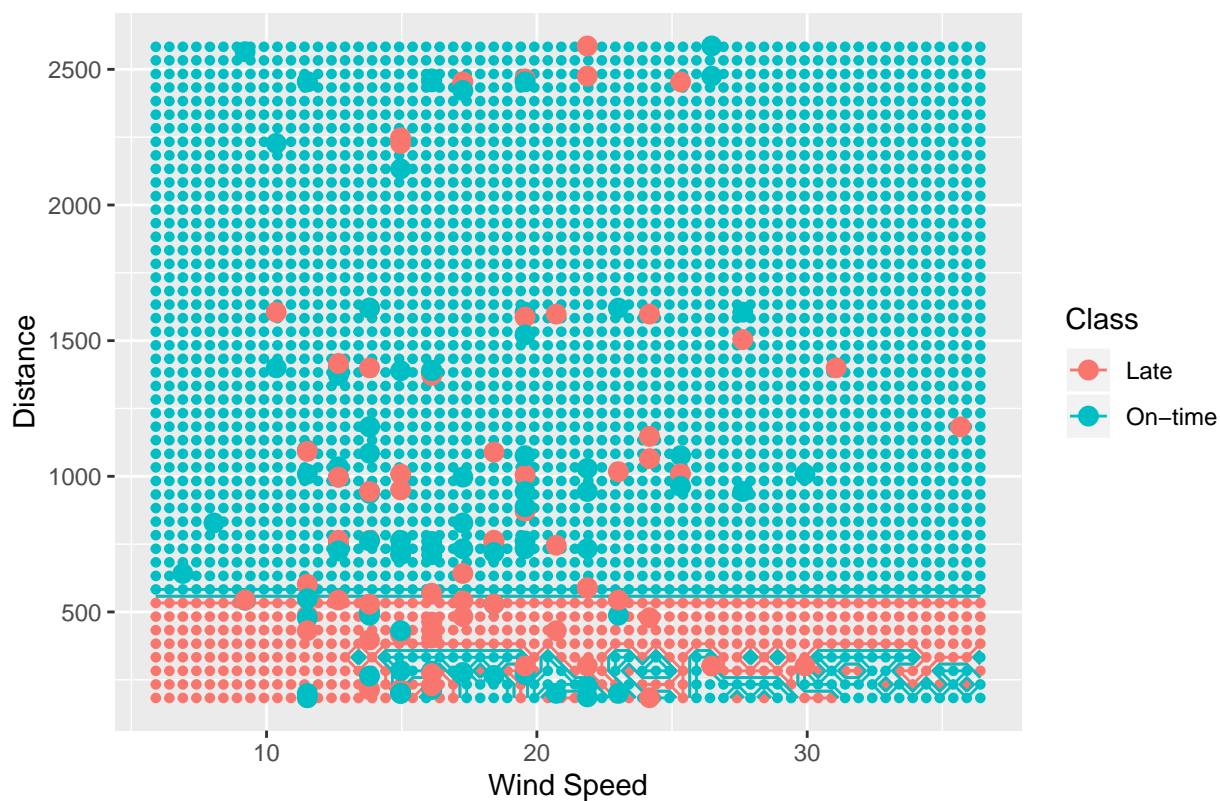
Value of K = 26



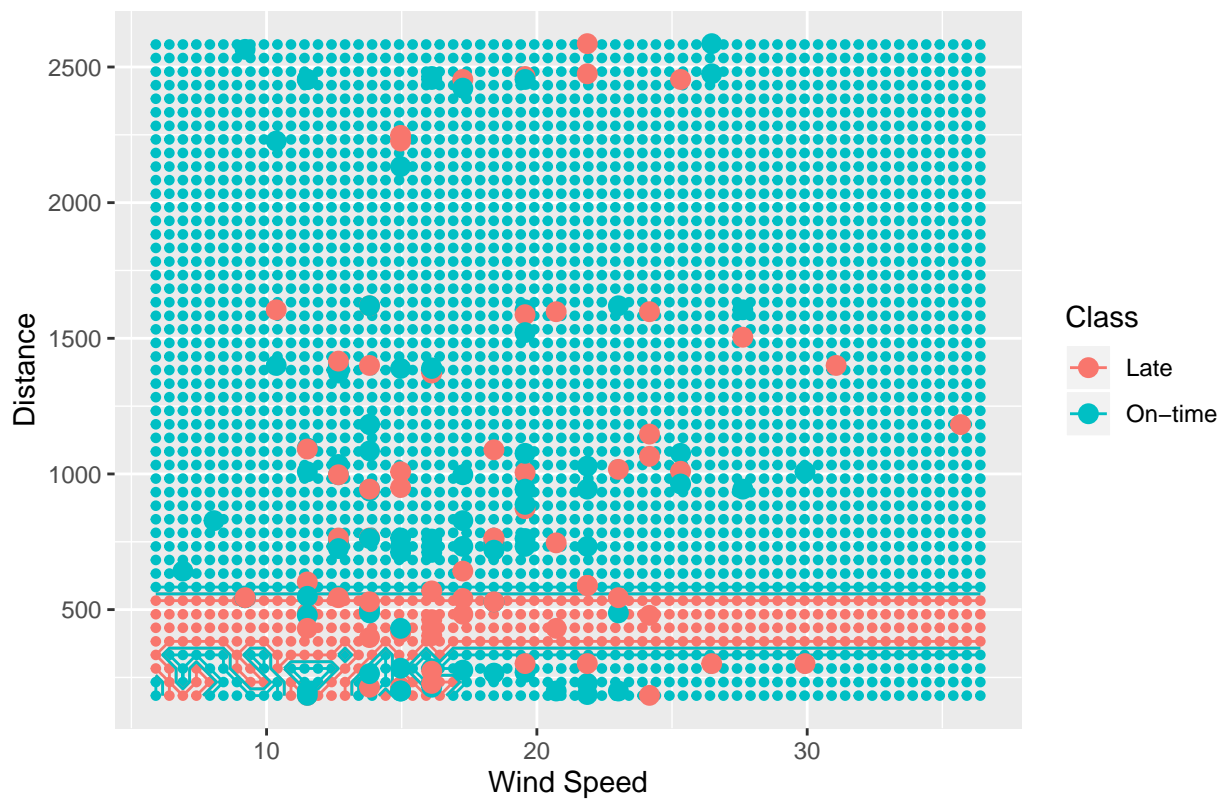
Value of K = 31



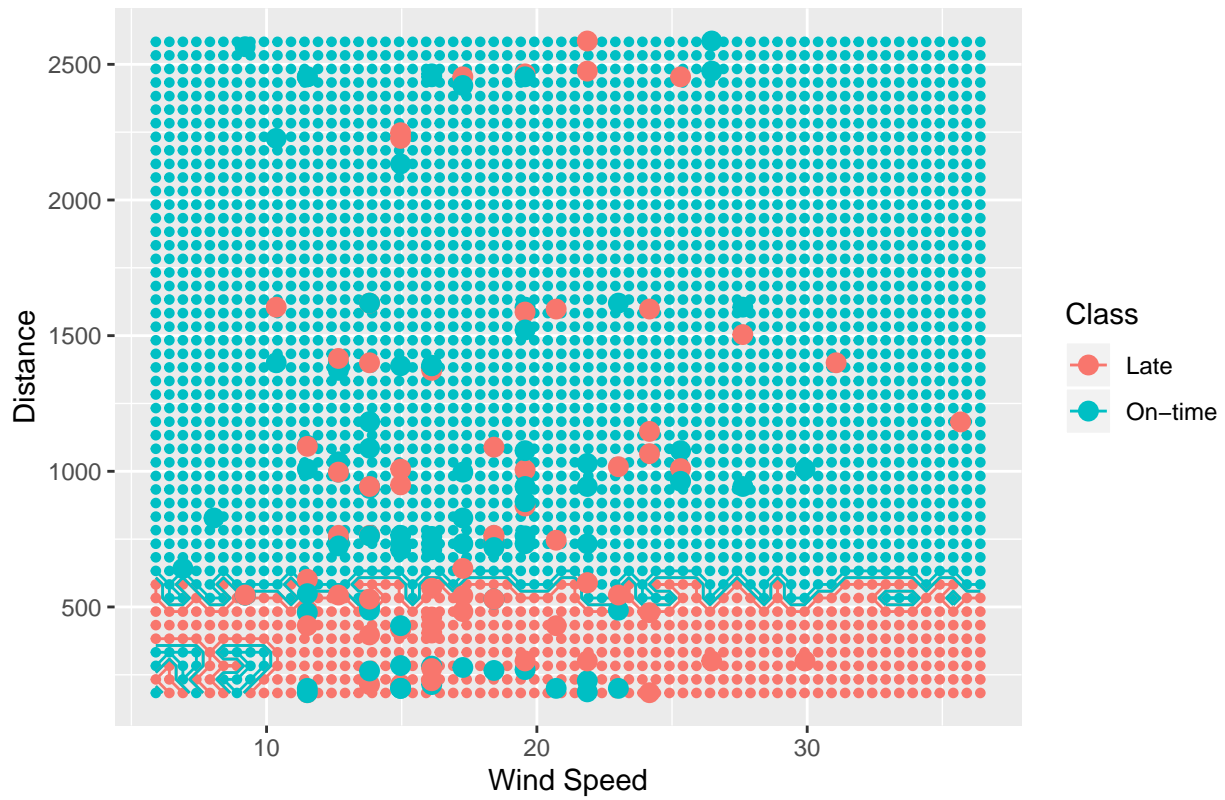
Value of K = 36



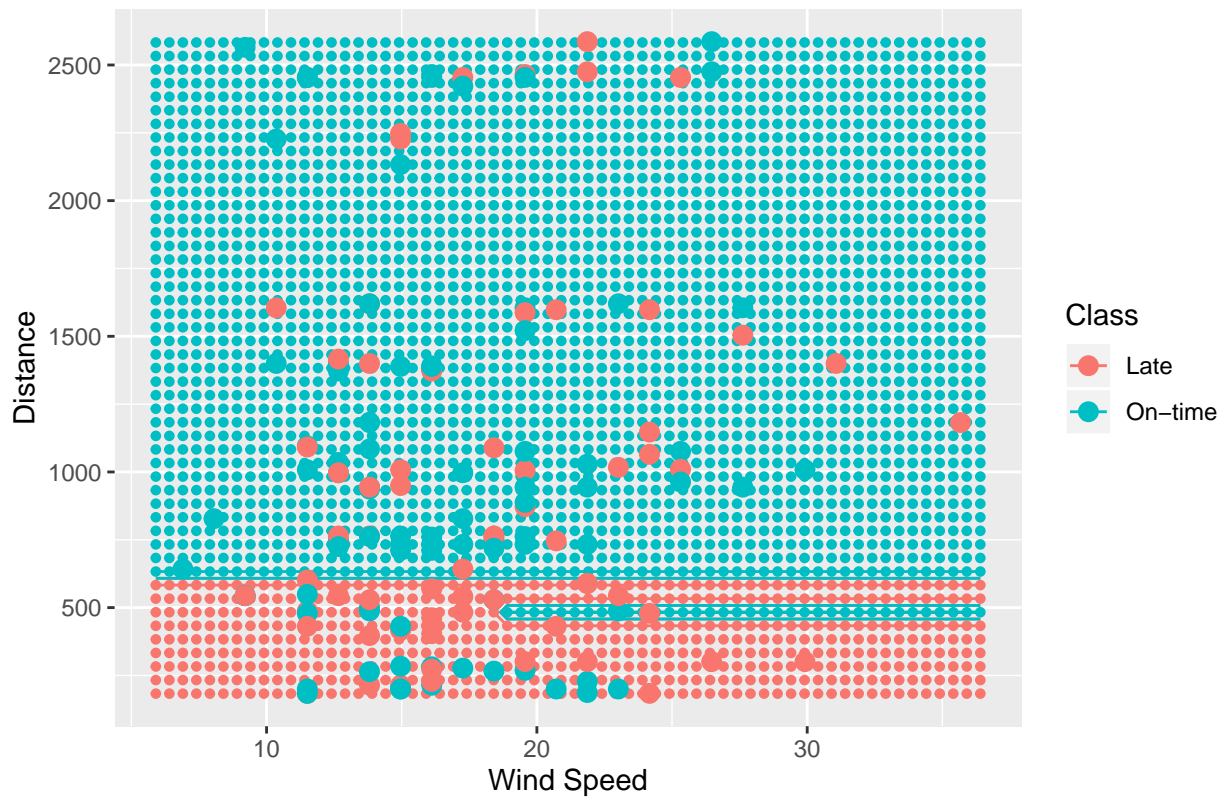
Value of K = 41



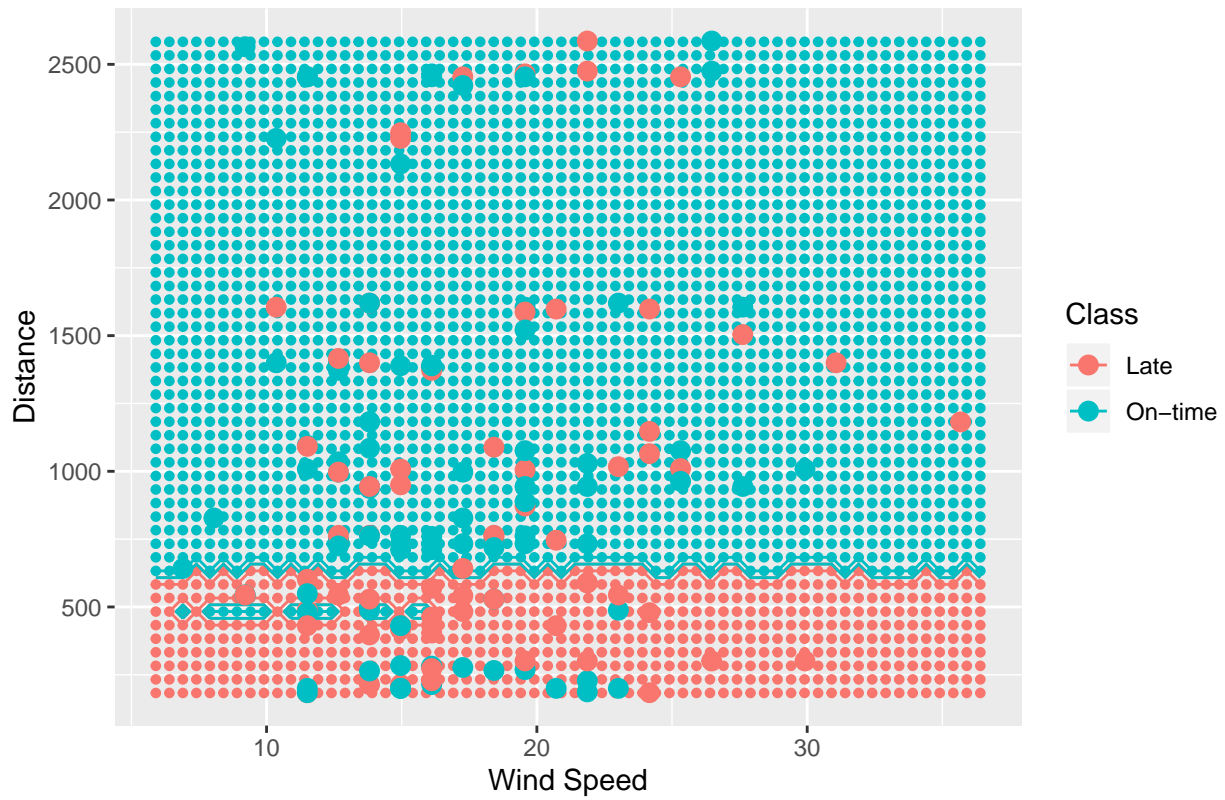
Value of K = 46



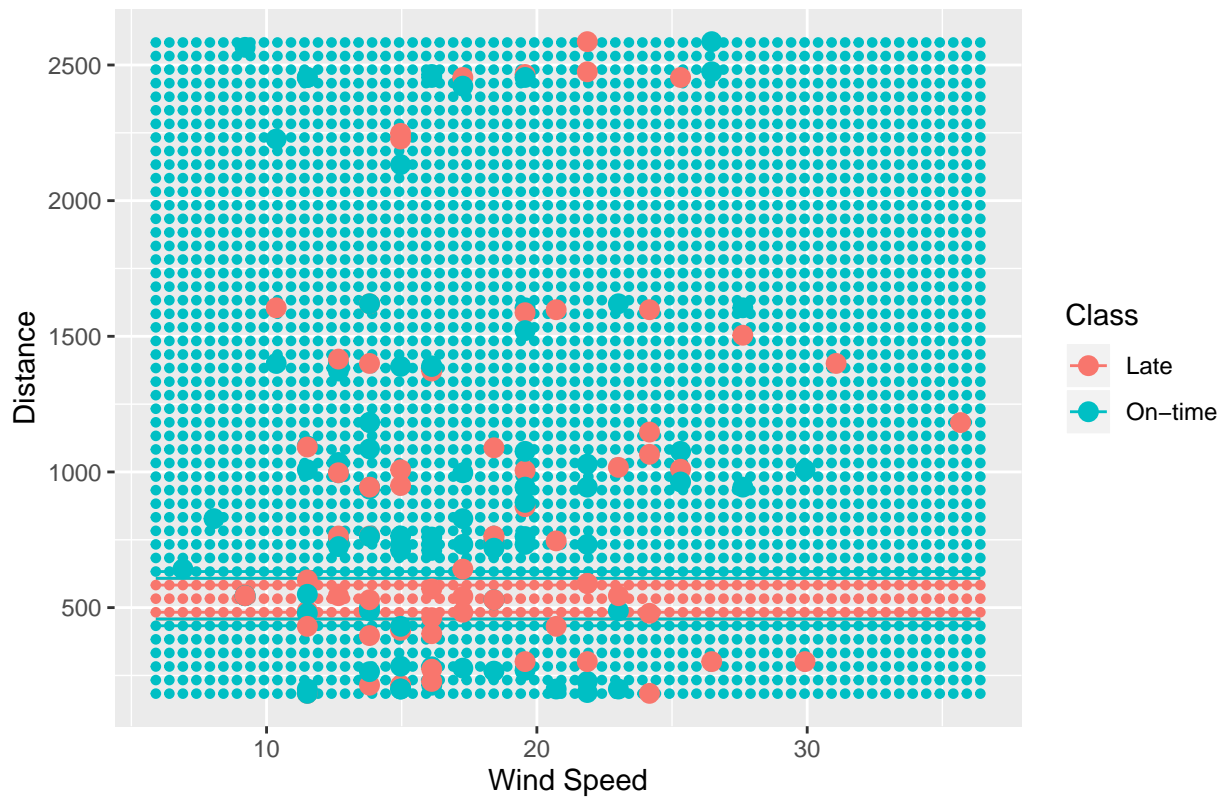
Value of K = 51



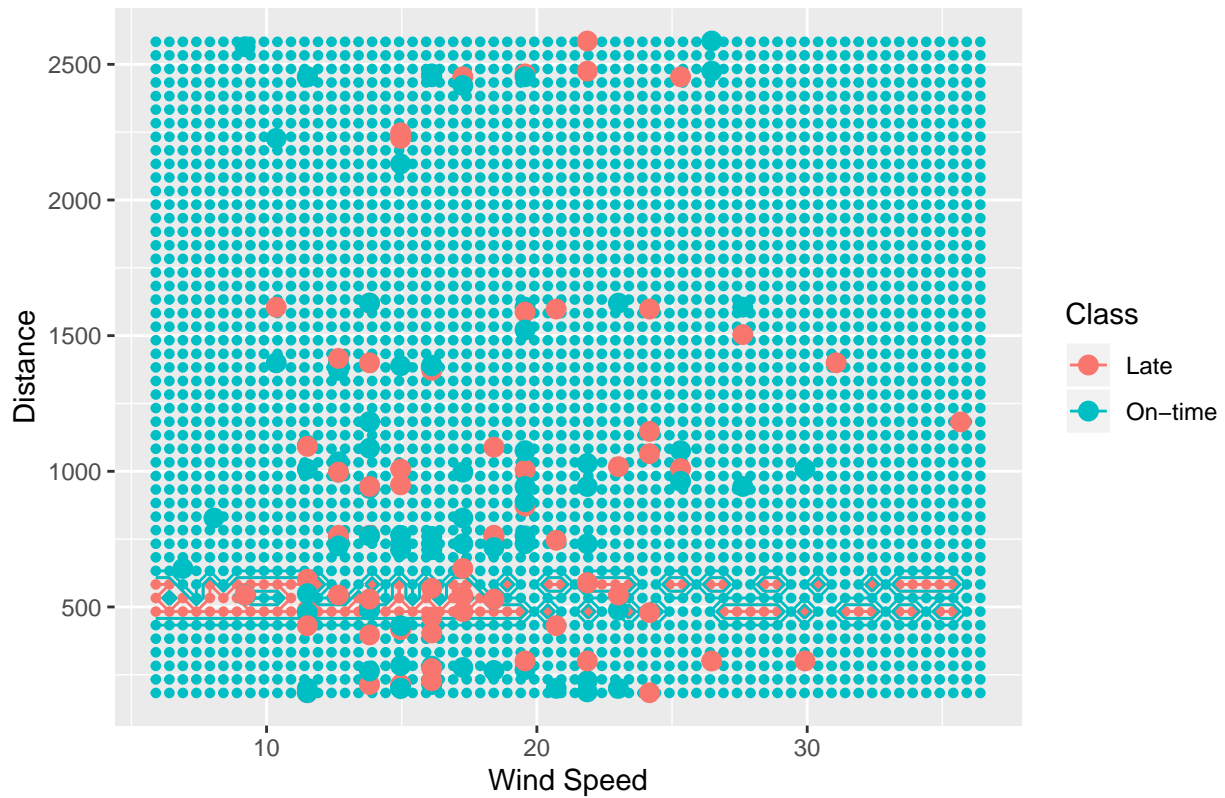
Value of K = 56



Value of K = 61

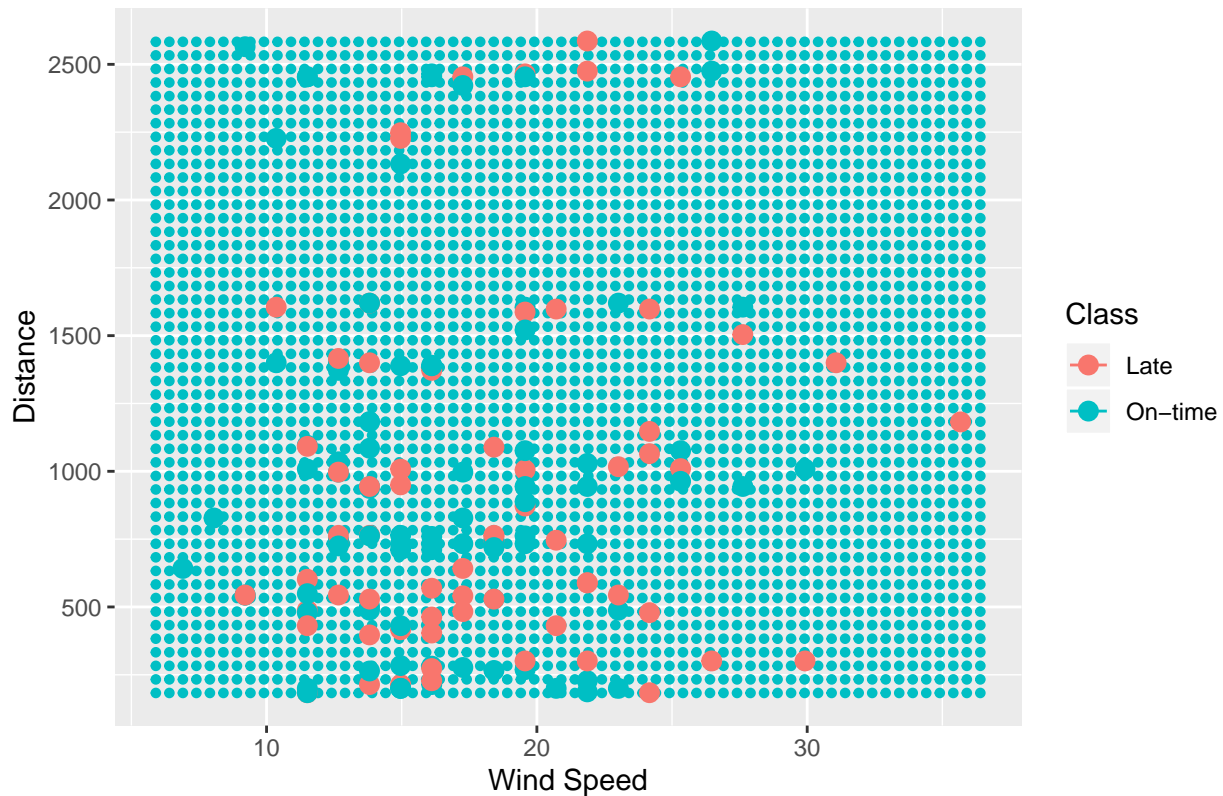


Value of K = 66



```
## Warning: Not possible to generate contour data
## Warning in grDevices::contourLines(x = sort(unique(data$x)), y =
## sort(unique(data$y)), : all z values are equal
## Warning: Not possible to generate contour data
```

Value of K = 71



```
# Function to calculate training error
func_accuracy <- function(k){

train <- rbind(weather_flight_rand_df[1:150,c('wind_speed','distance')])

cl <- factor(c(weather_flight_rand_df$Class[1:150]))

knn_accuracy <- knn(train=train, test=train, cl = cl, k = k)

data_tab <- table(knn_accuracy, cl)
matr <- as.matrix(data_tab)

# Calculating accuracy
accuracy <- sum(diag(matr))/length(cl)

# Calculating error
train_error <- (1 - accuracy)
print(paste("Training Error for K =", k, "is ", train_error))
}

for(i in seq(1, 75, 5)) {
  func_accuracy(i)
}
```

```
## [1] "Training Error for K = 1 is 0.0533333333333333"
## [1] "Training Error for K = 6 is 0.32"
## [1] "Training Error for K = 11 is 0.38"
## [1] "Training Error for K = 16 is 0.42"
```

```
## [1] "Training Error for K = 21 is 0.38"
## [1] "Training Error for K = 26 is 0.386666666666667"
## [1] "Training Error for K = 31 is 0.4"
## [1] "Training Error for K = 36 is 0.413333333333333"
## [1] "Training Error for K = 41 is 0.413333333333333"
## [1] "Training Error for K = 46 is 0.426666666666667"
## [1] "Training Error for K = 51 is 0.406666666666667"
## [1] "Training Error for K = 56 is 0.4"
## [1] "Training Error for K = 61 is 0.406666666666667"
## [1] "Training Error for K = 66 is 0.44"
## [1] "Training Error for K = 71 is 0.44"
```

Out of all the plots, the plot with $K = 11$ neither has high bias nor high variance. (It is neither too wiggly/curvy nor does one class cover the whole plot) Considering the bias-variance trade off, when either variance or the bias is too high, test error increases. So keeping that in mind, based on the plots created in the think $K = 11$ will give the smallest expected test error.

```
summary(lm(weather_flight$arr_delay~weather_flight$temp+weather_flight$dewp+weather_flight$humid+weather
```

```
##
## Call:
## lm(formula = weather_flight$arr_delay ~ weather_flight$temp +
##     weather_flight$dewp + weather_flight$humid + weather_flight$wind_dir +
##     weather_flight$wind_dir + weather_flight$wind_speed + weather_flight$wind_gust +
##     weather_flight$precip + weather_flight$pressure + weather_flight$visib,
##     data = weather_flight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -104.41  -22.49   -9.94    8.09   738.33
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    301.057399   26.834468   11.219 < 2e-16 ***
## weather_flight$temp      0.140113    0.093587    1.497  0.1344
## weather_flight$dewp     -0.042238    0.102716   -0.411  0.6809
## weather_flight$humid     0.244938    0.057761    4.241 2.23e-05 ***
## weather_flight$wind_dir  -0.013592    0.002218   -6.129 8.92e-10 ***
## weather_flight$wind_speed -0.118186    0.065608   -1.801  0.0716 .
## weather_flight$wind_gust  0.390119    0.056833    6.864 6.73e-12 ***
## weather_flight$precip    -9.303290   13.051536   -0.713  0.4760
## weather_flight$pressure  -0.280174    0.024806  -11.295 < 2e-16 ***
## weather_flight$visib     -3.175639    0.192232  -16.520 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.76 on 72724 degrees of freedom
## Multiple R-squared:  0.03219,    Adjusted R-squared:  0.03207
## F-statistic: 268.7 on 9 and 72724 DF,  p-value: < 2.2e-16
```