Project3

預測共享單車需求

109034543 黃荻雅

指導教授 : 邱銘傳 教授

Introduction

1

研究背景與目的

共享單車

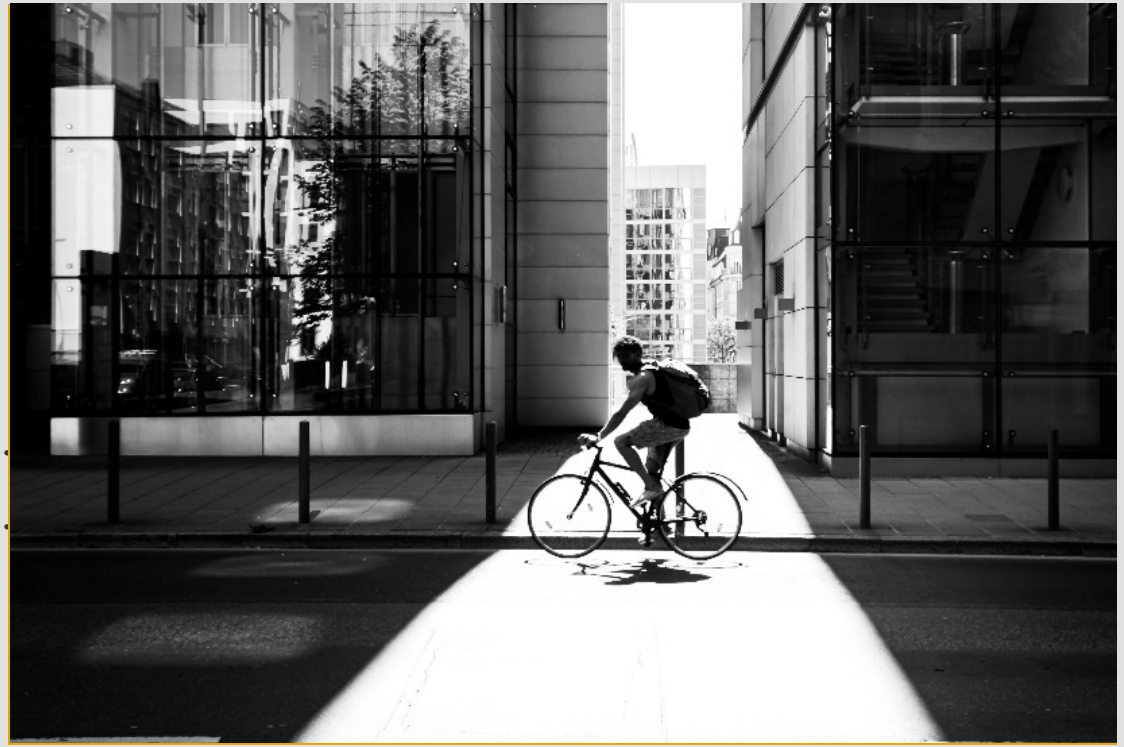
注重維護

無法及時滿足需求

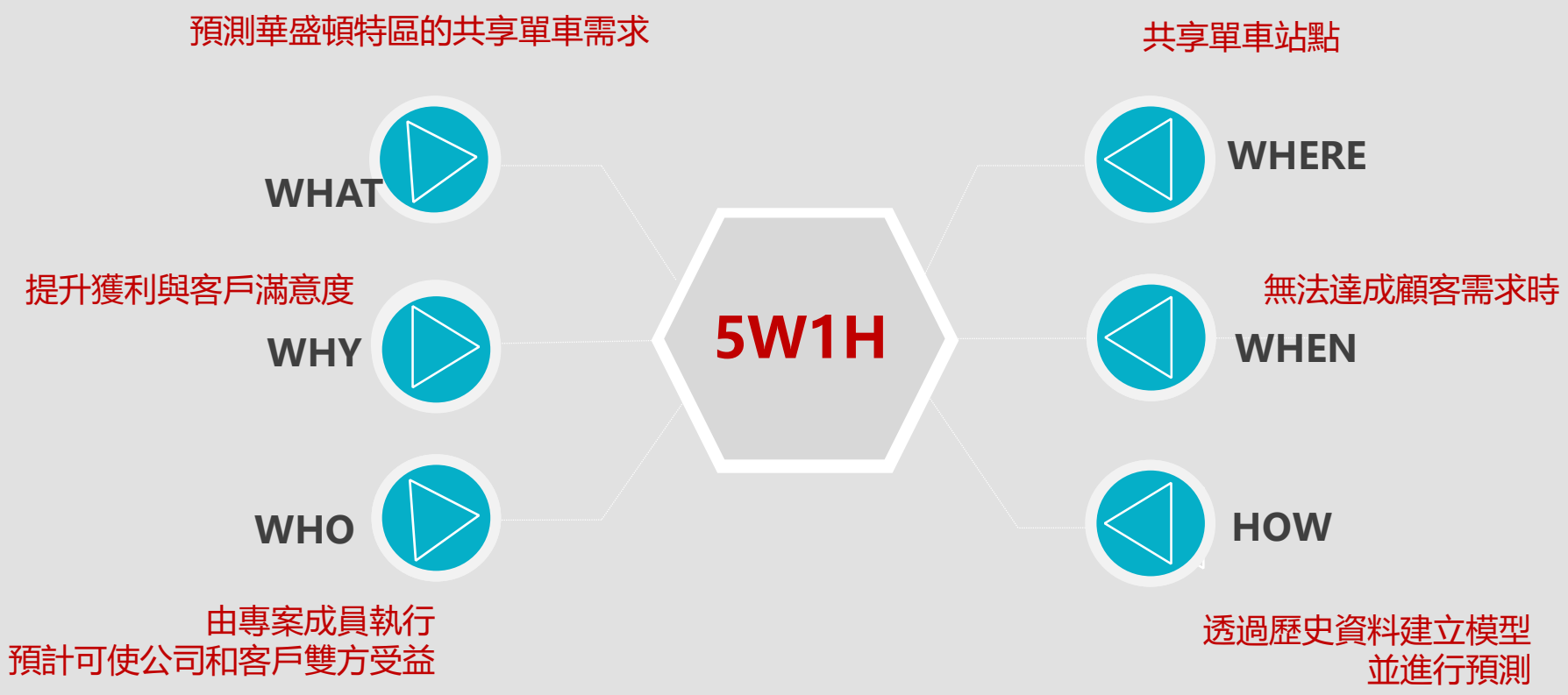
管理成本高昂

占用公共資源

華盛頓特區，2011~2012



5W1H分析

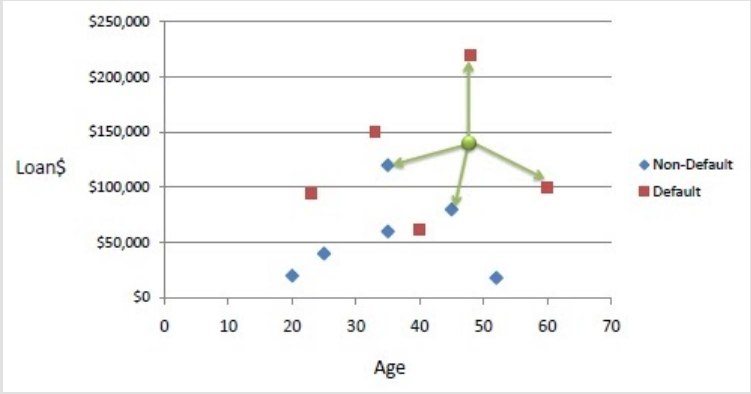

預測華盛頓特區的共享單車需求
WHAT
提升獲利與客戶滿意度
WHY
WHO
由專案成員執行
預計可使公司和客戶雙方受益
5W1H
共享單車站點
WHERE
無法達成顧客需求時
WHEN
HOW
透過歷史資料建立模型
並進行預測

方法

2

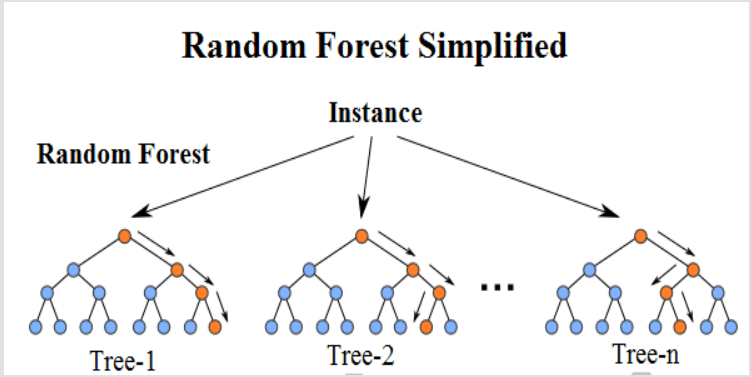
KNN演算法

- k近鄰分類法，k值決定一切
- 取近鄰k個樣本的平均值
- 運算速度快



Random Forest

- 本質是使用GINI的決策數
- Ensemble Method



03

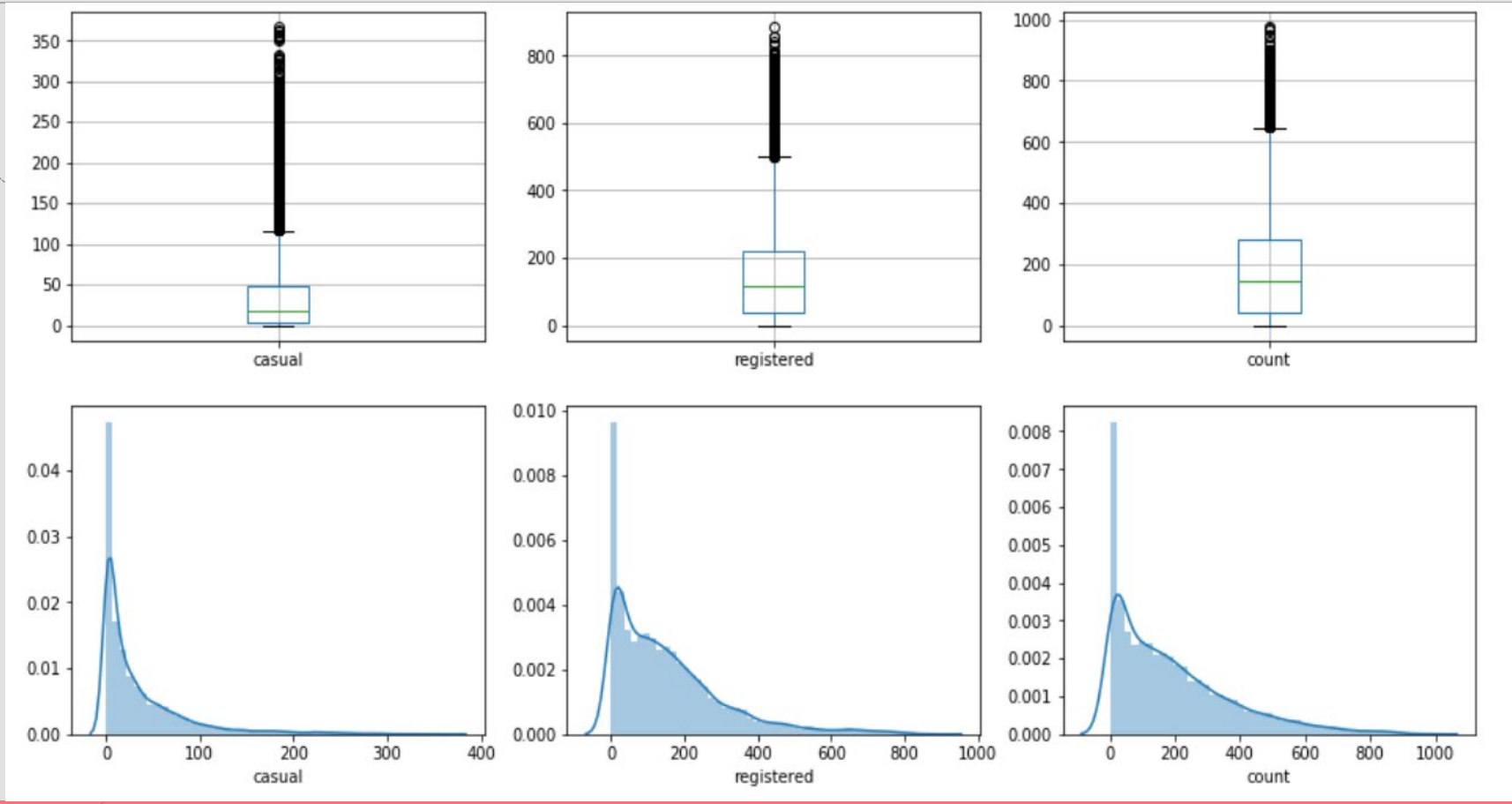
案例研究

資料前處理

- **處理時間資料**
- **歪斜的使用量資料**
- **Heatmap**
- **觀察各因子作用**

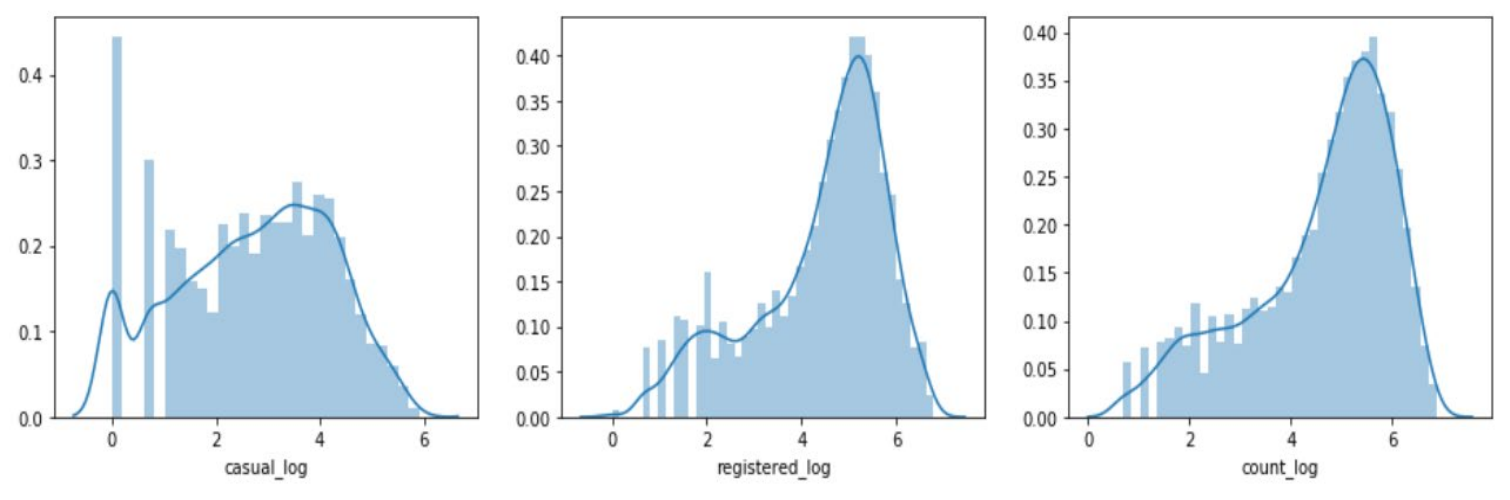
欄位名稱	說明
datetime	詳細到小時的日期
season	1:春天 2:夏天 3:秋天 4:冬天
holiday	0:非節日 1:節日
workingday	0:非工作日 1:工作日
weather	1:晴朗、微量的雲 2:有霧或雲 3:小雪、小雨、暴風 4:大雨、冰雹
temp	溫度(攝氏)
atemp	體感溫度(攝氏)
humidity	相對濕度
windspeed	風速
casual	沒註冊的使用者租借數量
registered	已註冊的使用者租借數量
count	總租借數量

資料前處理-歪斜的使用量資料



資料前處理-歪斜的使用量資料

- 資料取log



資料前處理-歪斜的使用量資料

- **初步比較兩者差異**

```
X_train, X_test, y_train, y_test = train_test_split(dt[factor], dt["count"], test_size = 0.3, random_s
rfmodel = RandomForestRegressor(n_estimators=3000, random_state=42, max_depth=30)
rfmodel.fit(X_train,y_train)
pred = rfmodel.predict(X_test)
MSE = mean_squared_error(pred, y_test)
MSLE = mean_squared_log_error(pred, y_test)
#測試集分數
print("RMSE =", MSE**0.5)
print("RMSLE =", MSLE**0.5)
```

```
RMSE = 154.13067529171568
RMSLE = 1.2845150860833485
```

未處理資料:

RMSE為154.13

RMSLE為1.28

```
pred = rfmodel.predict(X_test)
MSE = mean_squared_error(pred, y_test)
MSLE = mean_squared_log_error(pred, y_test)
#測試集分數
print("RMSE =", MSE**0.5)
print("RMSLE =", MSLE**0.5)
```

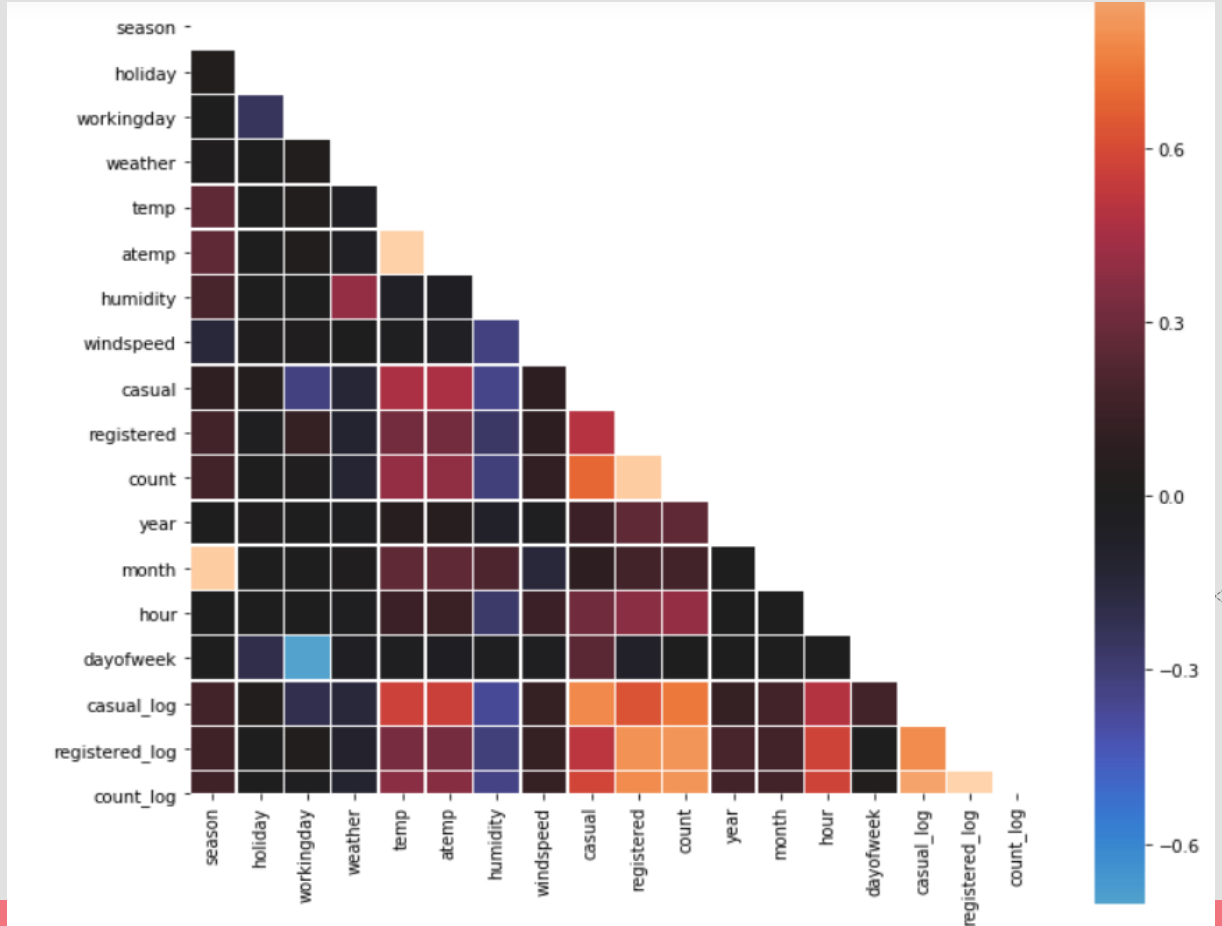
```
RMSE = 1.2691033479515628
RMSLE = 0.30814407619288725
```

資料取log:

RMSE為1.27

RMSLE為0.31

資料前處理-Heatmap

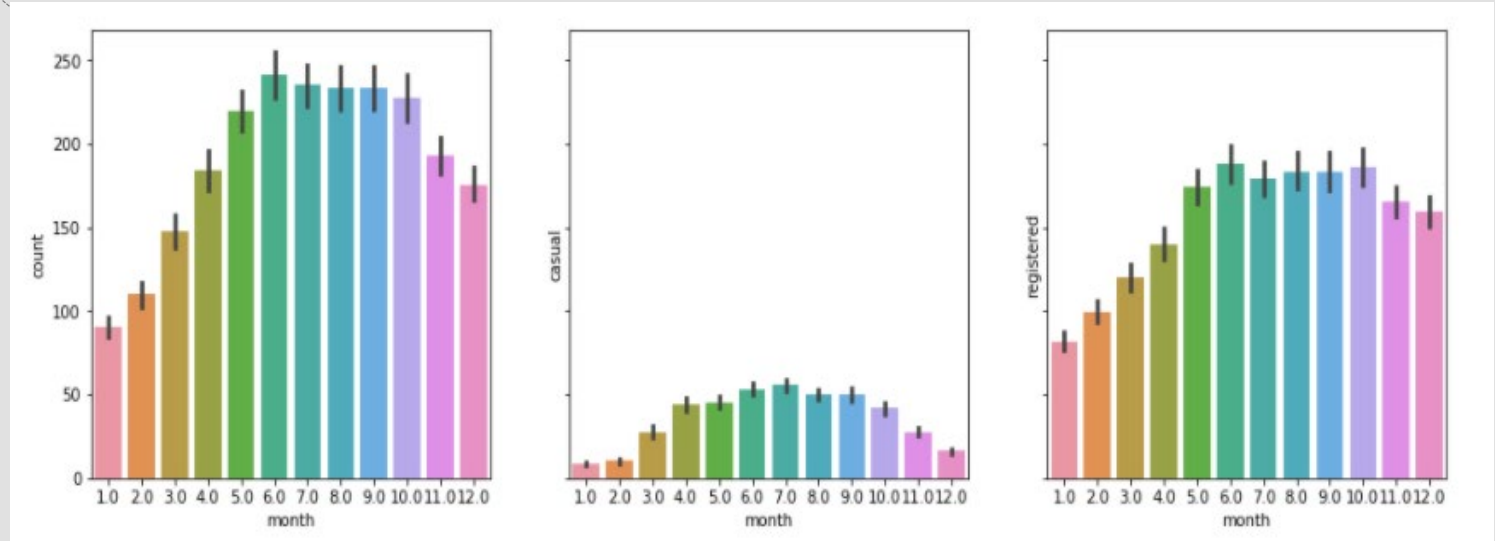


每個欄位對使用量皆沒有顯著影響

需要進一步做處理和排除

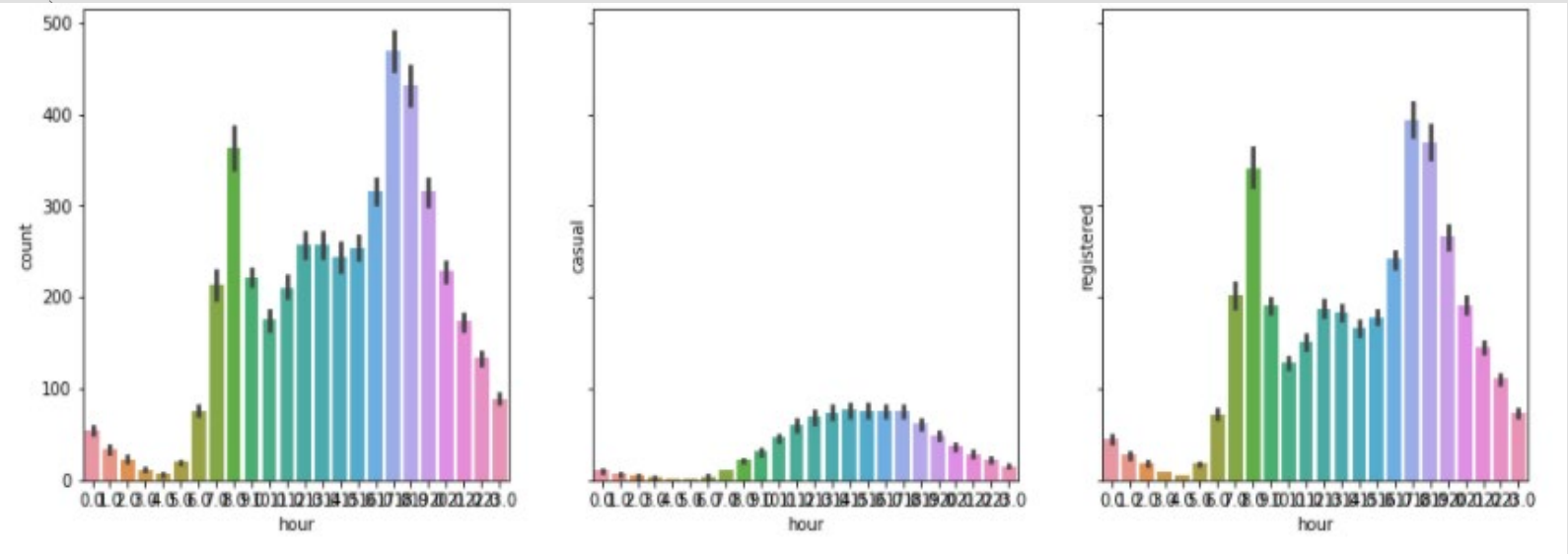
資料前處理-觀察各因子作用

- 三張圖趨勢相同，春天最低，高峰在夏、秋



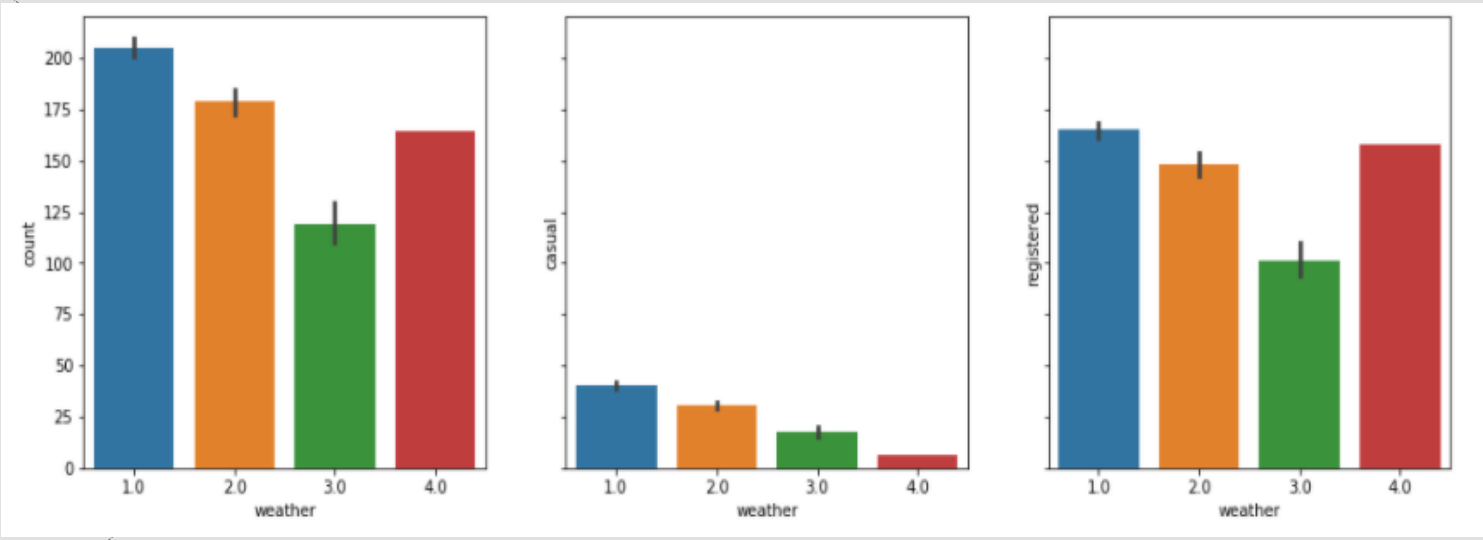
資料前處理-觀察各因子作用

- 未註冊使用者使用高峰在12-18，已註冊使用者的高峰則在上下班時間



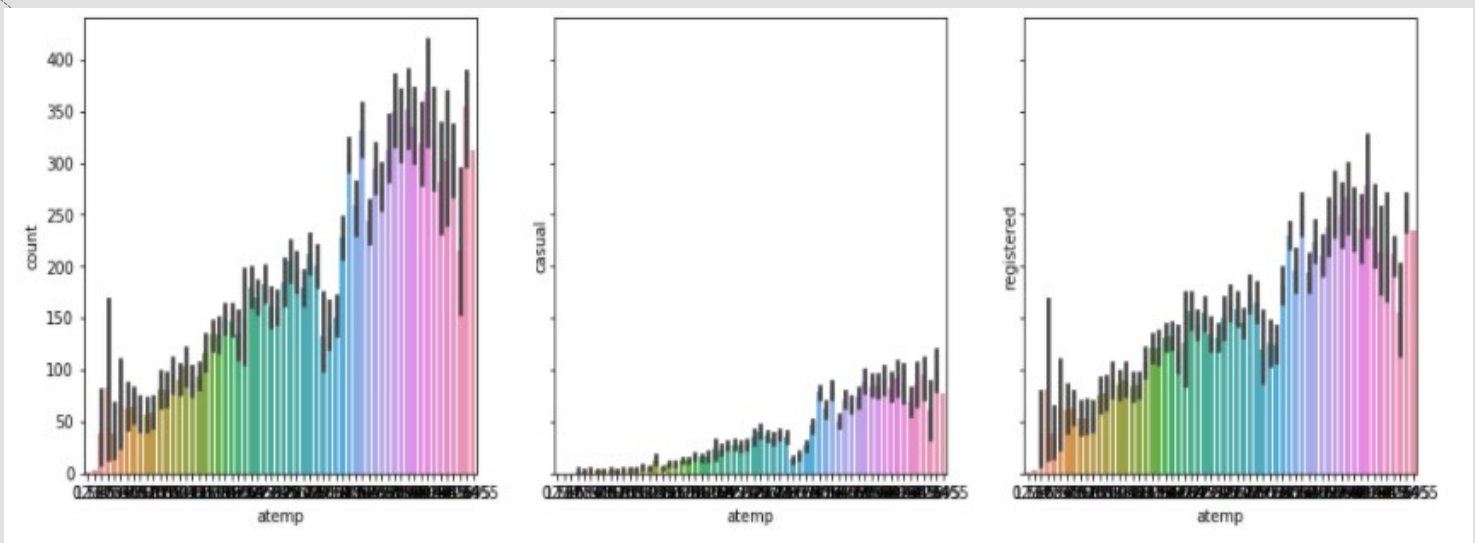
資料前處理-觀察各因子作用

- 未註冊使用者傾向在好天氣使用
- 已註冊使用者除了3(小雪)外，使用量差不多



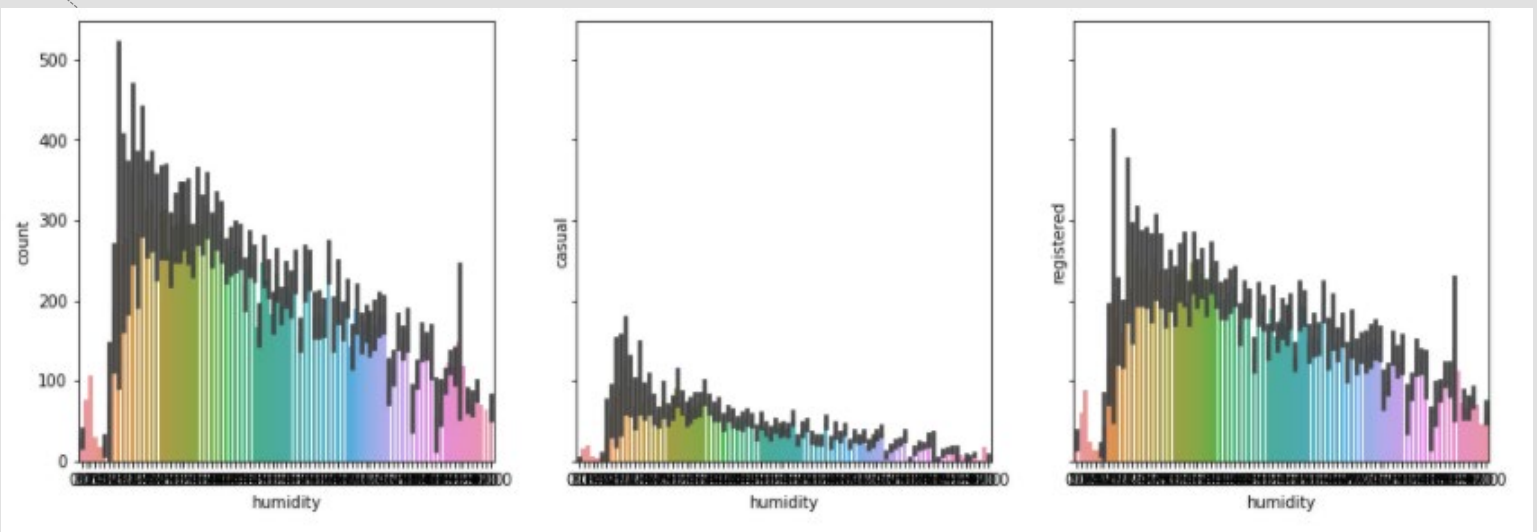
資料前處理-觀察各因子作用

- 趨勢與溫度相符



資料前處理-觀察各因子作用

- 傾向在低濕度時使用，但過於乾燥時使用量會下降



模型建構-KNN演算法

```
from sklearn.preprocessing import StandardScaler
#轉換資料
scaler = StandardScaler()
general_data = data[["casual_log", "count_log", "registered_log"]]
scaler.fit(general_data)
scaled = scaler.transform(general_data)
scaled = pd.DataFrame(data=scaled, columns = general_data.columns)

#轉換Log資料(KNN不支援)
scaled["casual_log"] = (scaled["casual_log"] * 10) .apply(np.floor)
scaled["registered_log"] = (scaled["registered_log"] * 10) .apply(np.floor)
scaled["count_log"] = (scaled["count_log"] * 10) .apply(np.floor)


general_data = general_data.join(data.drop(["casual_log", "count_log", "registered_log"],axis=1))
```

```
def build_kmodel(data, tuning_parameter):

    X_train, X_test, y_train, y_test = train_test_split(data.drop(["count_log", "registered_log","casu

    kmodel = KNeighborsRegressor(n_neighbors = tuning_parameter)
    kmodel.fit(X_train,y_train)
    pred=kmodel.predict(X_test)

    MSE = mean_squared_error(pred, y_test)

    return kmodel, MSE
```

模型建構-KNN演算法

```
#finding the best k value
k_values = [3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 30]

MSEs =[]
models = []

for k_value in k_values:
    model, MSE = build_kmodel(general_data, tuning_parameter = k_value)
    MSEs.append(MSE)
    models.append(model)
```

其中K value以迴圈測試最佳解(k=6)

```
kmodel = KNeighborsRegressor(n_neighbors = 6)
kmodel.fit(general_data.drop(["count_log", "registered_log","casual_log"],axis=1), general_data["count_
pred=kmodel.predict(data_test)

submission = pd.DataFrame({
        "datetime": dt_test["datetime"],
        "count": [max(0, x-1) for x in np.exp(pred)]
    })
submission.to_csv('KNN.csv', index=False)
```

在Kaggle所得的分數(RMSLE)為0.78，在排行榜上約前20%。

模型建構-Random Forest

```
X_train, X_test, y_train, y_test = train_test_split(data.drop(["count_log", "registered_log","casual_l
rfmodel = RandomForestRegressor(n_estimators=3000, random_state=42, max_depth=30)
rfmodel.fit(X_train,y_train)
pred = rfmodel.predict(X_test)
MSE = mean_squared_error(pred, y_test)
MSLE = mean_squared_log_error(pred, y_test)
#測試集分數
print("RMSE =", MSE**0.5)
print("RMSLE =", MSLE**0.5)
```

```
RMSE = 0.3493234181071481
RMSLE = 0.09199942633926979
```

使用訓練集測試

RMSE為0.35

RMSLE為0.09

```
rfmodel = RandomForestRegressor(n_estimators=3000, random_state=42, max_depth=30)
rfmodel.fit(data.drop(["count_log", "registered_log","casual_log"],axis=1),data["count_log"])

pred = rfmodel.predict(data_test)
submission = pd.DataFrame({
        "datetime": dt_test["datetime"],
        "count": [max(0, x-1) for x in np.exp(pred)]
    })
submission.to_csv('RF.csv', index=False)
```

參數對模型表現影響不大

模型建構-Random Forest

```
rfmodel = RandomForestRegressor(n_estimators=3000, random_state=42, max_depth=30)
rfmodel.fit(data.drop(["count_log", "registered_log","casual_log"],axis=1),data["count_log"])

pred = rfmodel.predict(data_test)
submission = pd.DataFrame({
        "datetime": dt_test["datetime"],
        "count": [max(0, x-1) for x in np.exp(pred)]
    })
submission.to_csv('RF.csv', index=False)
```

使用測試集資料進行預測，並提交到Kaggle

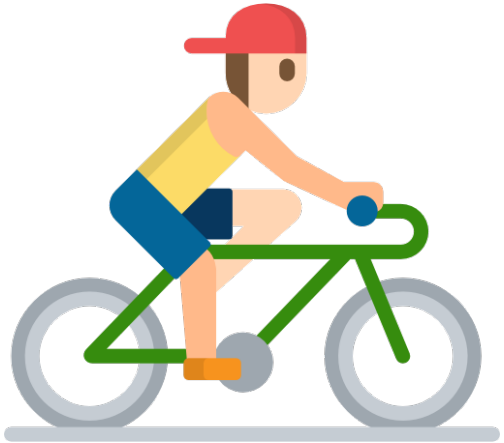
所得的分數(RMSLE)為0.41，在排行榜上約前10%

結論與展望

資料來源
- 資料完整
- 進一步分析?

專案可行性
- RMSLE<0.5
- 可協助處裡需求

資料更新
- 資料為2011~2012
  - 需求是否有不同?
    - 各地區差異



THANK YOU