Reducing Commercial Aviation Fatalities

組員：
10446009 戴孟涓
10446026 鄒佩欣
10446029 吳筱筑

摘要、目的與動機：

大多數與飛行相關的死亡事故源於“飛機狀態意識”的喪失。也就是說，飛行員可能會分心，困倦或處於其他危險的認知狀態時，注意管理效率低下。

建立一個模型來檢測機組人員生理數據中的變化。利用在測試情況下從實際飛行員獲取的數據並計算，以監控飛行員的認知狀態。

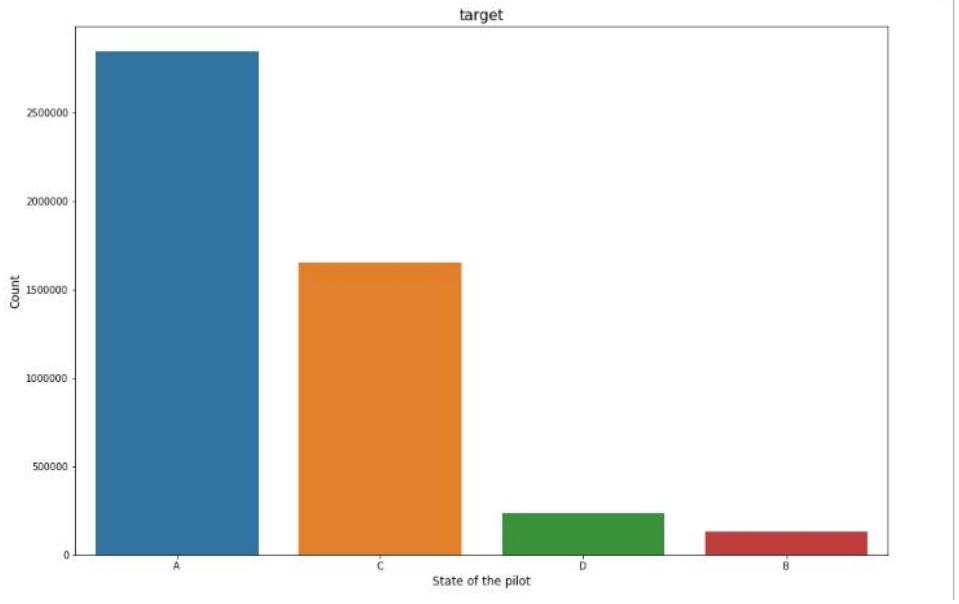
資料集介紹：

1. id (只有 test 跟最後文件才有)：一個 crew 加上 time
2. crew ：一對空服員，總共有 9 個 crew
3. experiment： CA, DA, SS , LOFT 其中一個.
4. time ： 進入 experiment 的時間(秒)
5. seat ：左邊空服員(0) 、右邊空服員 (1)
6. egg 開頭的變數為腦電圖的紀錄(分別代表位置)：
   eeg_fp1、eeg_f7、eeg_f8、eeg_t4、eeg_t6、eeg_t5、eeg_t3、eeg_fp2、eeg_o1、eeg_p3、eeg_pz、eeg_f3、eeg_fz、eeg_f4、eeg_c4、eeg_p4、eeg_poz、eeg_c3、eeg_cz、eeg_o2
7. ecg ：三點心電圖（3-point Electrocardiogram signal）
8. r ：Respiration,測試胸腔起伏
9. gsr ：皮膚電阻的感應（Galvanic Skin Response）
10. event : The state of the pilot at the given time: one of A = baseline, B = SS, C = CA, D = DA

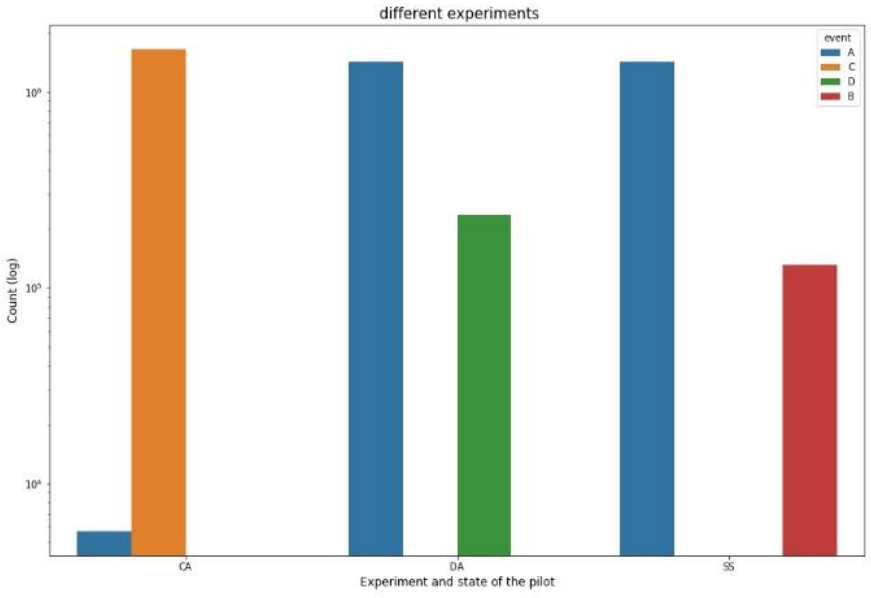
資料預處理：

1. 先行判斷資料中有無空值
2. 各特徵分析

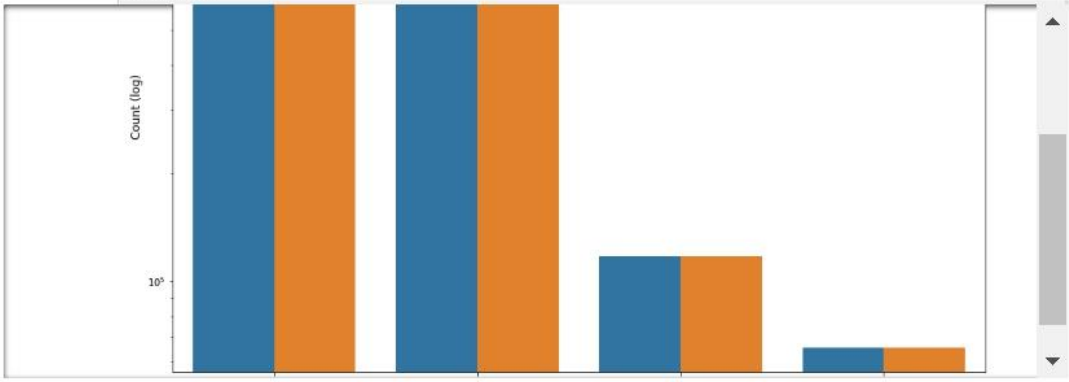
```
In [3]: plt.figure(figsize=(15,10)) #count A,B,C,D個數
        sns.countplot(train['event'])
        plt.xlabel("State of the pilot", fontsize=12)
        plt.ylabel("Count", fontsize=12)
        plt.title("target", fontsize=15)
        plt.show()
```




```
In [4]: #區分DA,CA,SS
        plt.figure(figsize=(15,10))
        sns.countplot('experiment', hue='event', data=train)
        plt.xlabel("Experiment and state of the pilot", fontsize=12)
        plt.ylabel("Count (log)", fontsize=12)
        plt.yscale('log')
        plt.title(" different experiments", fontsize=15)
        plt.show()
```



```
In [5]: #0,1區分正副駕駛
        plt.figure(figsize=(15,10))
        sns.countplot('event', hue='seat', data=train)
        plt.xlabel("Seat and state of the plot", fontsize=12)
        plt.ylabel("Count (log)", fontsize=12)
        plt.yscale('log')
        plt.title("Left seat 、 light seat ?", fontsize=15)
        plt.show()
```



3. 將資料分成測試及訓練，將 event 中的文字轉成數字以便後續分析

```
In [4]: from sklearn.model_selection import train_test_split #分訓練及測試

        train, val_df = train_test_split(train, test_size=0.2, random_state=420)
        print(f"Training on {train.shape[0]} samples.")
```

```
Training on 3893936 samples.
```

```
In [5]: train.loc[train['event']=="A",'event']=0
        train.loc[train['event']=="B",'event']=1
        train.loc[train['event']=="C",'event']=2
        train.loc[train['event']=="D",'event']=3
```

```
In [6]: #print(train["event"])
```

```
In [7]: val_df.loc[val_df['event']=="A",'event']=0
        val_df.loc[val_df['event']=="B",'event']=1
        val_df.loc[val_df['event']=="C",'event']=2
        val_df.loc[val_df['event']=="D",'event']=3
```

研究方法：

預測 Model 使用 LightGBM，並使用 Feature importance 及 Confusion Matrix

- 產生預測 Model

In [9]:
```
import lightgbm as lgb

features = ["crew", "seat","ecg","r","gsr"]

def run_lgb (train, val_df):


    params = {"objective" : "multiclass",
              "num_class": 4,
              "metric" : "multi_error",
              "num_leaves" : 30,
              "min_child_weight" : 50,
              "learning_rate" : 0.1,
              "bagging_fraction" : 0.7,
              "feature_fraction" : 0.7,
              "bagging_seed" : 420,
              "verbosity" : -1
             }

    lg_train = lgb.Dataset(train[features], label=(train["event"]))

    lg_test = lgb.Dataset(val_df[features], label=(val_df["event"]))

    model = lgb.train(params, lg_train, 1000, valid_sets=[lg_test], early_stopping_rounds=50, verbose_eval=100)

    return model

#print(train["event"])

model = run_lgb(train, val_df)
```

```
Training until validation scores don't improve for 50 rounds.
[100]   valid_0's multi_error: 0.0882366
[200]   valid_0's multi_error: 0.076034
[300]   valid_0's multi_error: 0.0708937
[400]   valid_0's multi_error: 0.0677935
[500]   valid_0's multi_error: 0.065245
[600]   valid_0's multi_error: 0.0631042
[700]   valid_0's multi_error: 0.0616661
[800]   valid_0's multi_error: 0.0604848
[900]   valid_0's multi_error: 0.0595859
[1000]  valid_0's multi_error: 0.0587867
Did not meet early stopping. Best iteration is:
[1000]  valid_0's multi_error: 0.0587867
```

- 最終結果預測及輸出

In [34]:
```
pred_test = model.predict(test[features], num_iteration=model.best_iteration)    #預測
```

In [ ]:
```

```

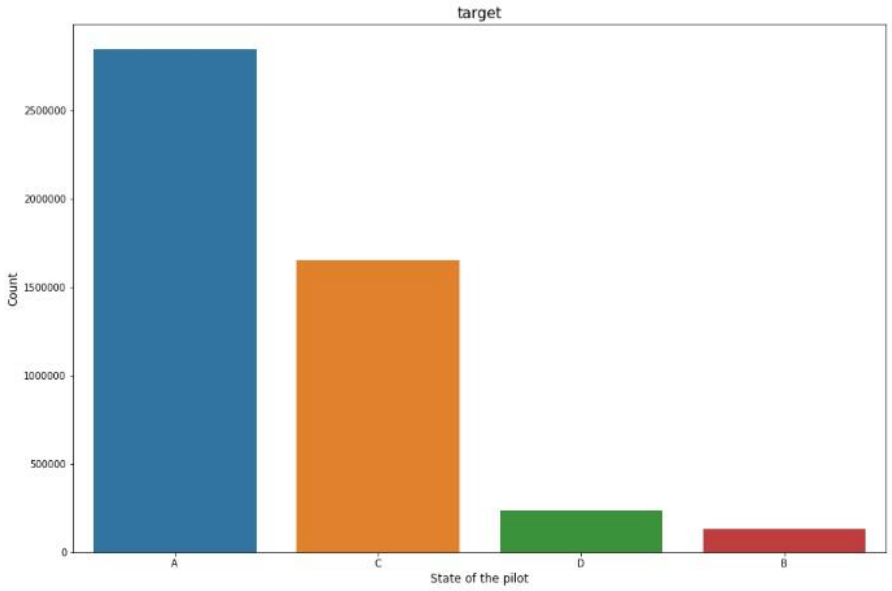
In [ ]:
```
#輸出檔案
submission = pd.DataFrame(np.concatenate((np.arange(len(test))[:, np.newaxis], pred_test), axis=1), columns=['id', 'A', 'B', 'C',
submission['id'] = submission['id'].astype(int)
```

In [ ]:
```
submission.head()
```

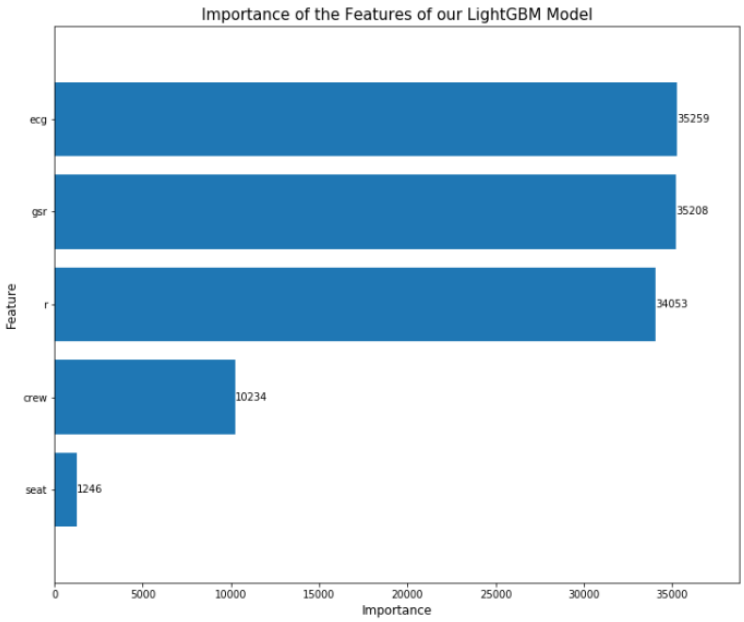
In [ ]:
```
submission.to_csv("R_submission.csv", index=False)
```

- Feature importance

In [3]:
```
plt.figure(figsize=(15,10)) #count A,B,C,D個數
sns.countplot(train['event'])
plt.xlabel("State of the pilot", fontsize=12)
plt.ylabel("Count", fontsize=12)
plt.title("target", fontsize=15)
plt.show()
```

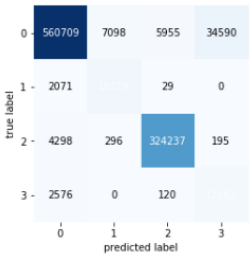


n [33]:
```
#看特徵重要重度
fig, ax = plt.subplots(figsize=(12,10))
lgb.plot_importance(model, height=0.8, ax=ax)
ax.grid(False)
plt.ylabel('Feature', size=12)
plt.xlabel('Importance', size=12)
plt.title("Importance of the Features of our LightGBM Model", fontsize=15)
plt.show()
```



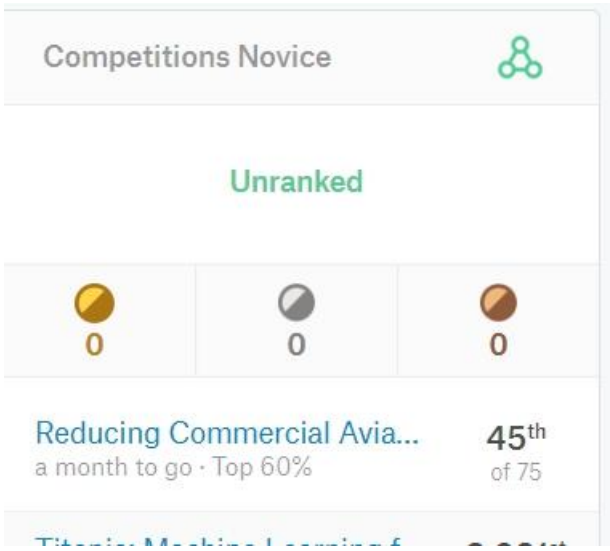
- Confusion Matrix

```
In [48]: from sklearn.metrics import confusion_matrix

         pred_val = model.predict(val_df[features], num_iteration=model.best_iteration)

         from mlxtend.plotting import plot_confusion_matrix

         conf_mat_val = confusion_matrix(np.argmax(pred_val, axis=1), val_df["event"].values)
         plot_confusion_matrix(conf_mat_val, ["0", "1", "2", "3"])

Out[48]: (<Figure size 432x288 with 1 Axes>,
          <matplotlib.axes._subplots.AxesSubplot at 0x2cc8291b160>)
```



結論：

在飛行員的飛行測試中，我們透過各項測試數據，如：皮膚的擴張、心電圖等相關數據，可得知在飛行狀態中的突發狀況會影響飛行員當下的危機處理。所以，我們應透過加以訓練的方式，讓飛行員能夠在過程中適應突發狀況，降低慌張並做出快速且正確反應，做最佳的處理，以降低空難的發生率。



參考文獻：

- [資料分析&機器學習] 第 4.1 講 : Kaggle 競賽-鐵達尼號生存預測-(前 16% 排名)
- https://ithelp.ithome.com.tw/users/20103074/ironman/1869
- https://www.youtube.com/channel/UC_MQXAzHP7E1YB5K49ZCqRQ
- https://scikit-learn.org/stable/
- https://www.csdn.net/