

Présentation 3

- Analyse des corrélations
- Mise en place d'une grille pour l'optimisation des paramètres du Random Forest
 - Comparaison selon différents critères
 - Problèmes et à venir

Rappels :

Travail sur PPMI_Gait_ArmSwing :

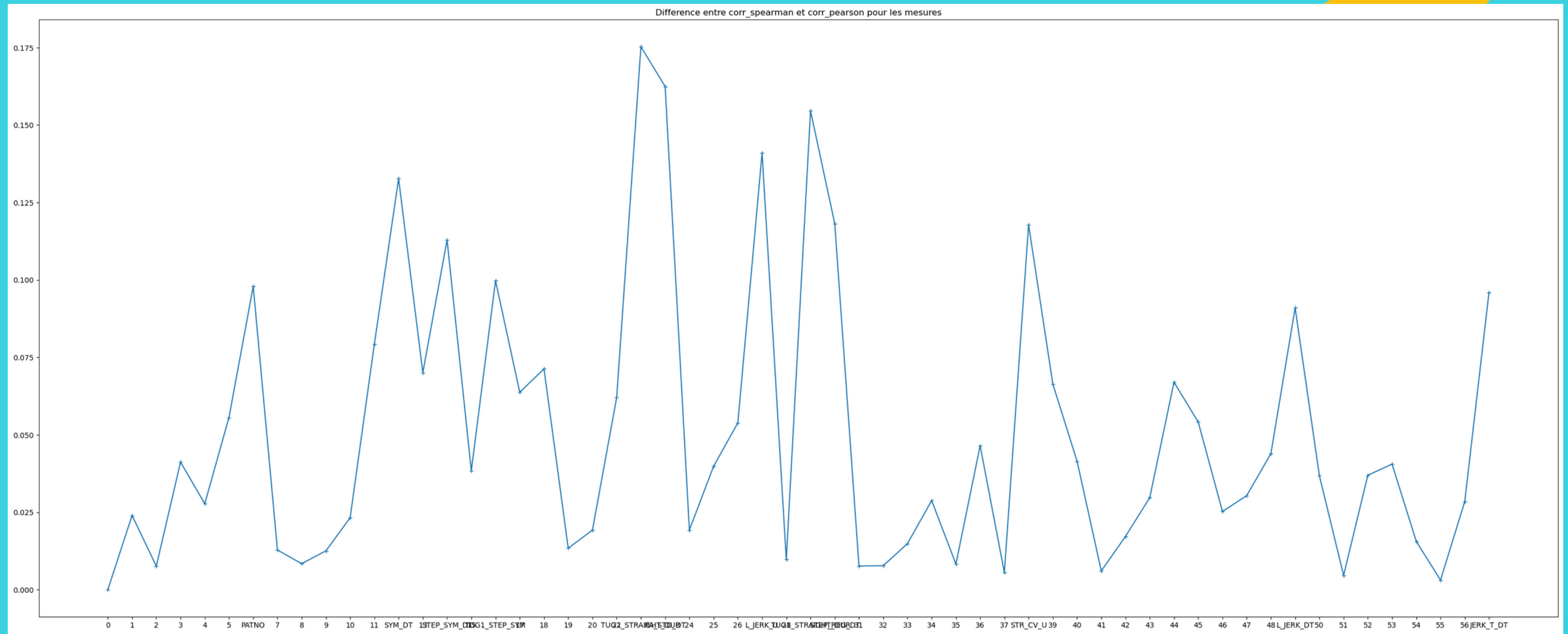


- The gait system used includes three lightweight wireless wearable sensors containing three axial accelerometers, gyroscopes and magnetometers
- 6 tests made : SWAY eyes open/closed + TUG 1 + TUG 2 + Usual walk + Dual task walk (voir pdf)
- Motor features extracted from the raw accelerometer and gyroscope signals
- Data processing with 54 measures calculated

Analyse des corrélations:



Calcul des écarts pour les 2 correlations (Pearson et Spearman) des 54 mesures



Analyse des corrélations:

COHORT	0.000000	LA_STD_DT	0.036945
LA_AMP_DT	0.003117	TRA_DT	0.038399
RA_STD_U	0.004624	STEP_REG_U	0.039837
SW_PATH_CL	0.005537	JERK_T_U	0.040540
SW_VEL_CL	0.006062	TUG2_DUR	0.041184
TUG2_TURNS_DUR	0.007607	SW_JERK_OP	0.041485
TUG1_STEP_REG	0.007658	R_JERK_U	0.043953
T_AMP_DT	0.007744	R_JERK_DT	0.046495
T_AMP_U	0.008264	CAD_DT	0.053854
ASYM_IND_DT	0.008392	RA_AMP_U	0.054126
SW_FREQ_OP	0.009716	STR_T_U	0.055612
TUG1_STEP_NUM	0.012540	SYM_U	0.062045
ASA_DT	0.012829	CAD_U	0.063704
ASA_U	0.013405	SW_PATH_OP	0.066371
TUG2_STEP_REG	0.014878	SW_JERK_CL	0.066993
RA_AMP_DT	0.015573	STEP_SYM_U	0.070070
SW_VEL_OP	0.017218	TUG2_STEP_SYM	0.071288
ASYM_IND_U	0.019233	STR_T_DT	0.079294
TRA_U	0.019277	L_JERK_DT	0.090994
TUG2_STEP_NUM	0.023242	JERK_T_DT	0.095951
TUG1_TURNS_DUR	0.023980	PATNO	0.097929
LA_AMP_U	0.025256	TUG1_STEP_SYM	0.099733
TUG1_DUR	0.027759	STEP_SYM_DT	0.112899
SP__DT	0.028463	STR_CV_U	0.117811
SW_FREQ_CL	0.028761	STEP_REG_DT	0.118126
STR_CV_DT	0.029723	SYM_DT	0.132666
SP_U	0.030346	L_JERK_U	0.140969
LA_STD_U	0.036818	TUG1_STRAIGHT_DUR	0.154564
		RA_STD_DT	0.162406
		TUG2_STRAIGHT_DUR	0.175274

Correlations max : +0.5 pour TUG1_TURNS_DUR (Average step duration during turns (sec))

et -0.53 pour JERK_T_DT (Jerk de l'accélération mouvement des jambes pendant la double tâche marche (m/sec³) => fluidité de la marche)



Les valeurs de coefficient de corrélation inférieures à +0,8 ou supérieures à -0,8 ne sont pas considérées comme significatives.

RandomForest Classifier

Procédure pour le meilleur set d'hyperparamètre mais sans faire d'overfitting :

- CrossValidation

Demande en calcul enorme car pour chaque set d'hyperparamètre il faut tester K-Fold CV => Nombre d'execution de l'algo = (nbre_set*K):

- Faire un premier balayage Random généralisé sur les set d'hyperparamètre les plus performant + comparer avec set de base
- Puis effectuer un recherche plus précise avec une grille sur le set random le plus précis précédent + comparer avec set de base
- Enfin, tableau récapitulatif des performances

RandomForest Classifieur

Etablissement d'une grille de set à tester puis test avec k-fold(k=5)

```
classif = RandomForestClassifier(n_estimators = 10)
```

Parameters	Base Model	Random Search Model
n_estimators	10	160
min_samples_split	2	10
min_samples_leaf	1	2
max_features	auto	sqrt
max_depth	None	50.0
bootstrap	True	True

```
# 1er balayage sur large éventail de sets d'hyperparamètres avec RandomizedSearchCV
from sklearn.model_selection import RandomizedSearchCV

# Combinaison des classifications d'ensembles d'entraînement générés aléatoirement
bootstrap = [True, False]

# Profondeur max de l'arbre
max_depth = np.linspace(10, 100, num = 10).tolist()

# Number of features to consider when looking for the best split, sqrt=auto, essayer None
max_features = ['auto', 'sqrt', 'log2']

# the minimum number of samples required to split an internal node,
min_samples_split = [2, 5, 10, 20]

# min_samples_leaf specifies the minimum number of samples required to be at a leaf node
min_samples_leaf = [1, 2, 4]

# Nombre d'arbres dans forêt aléatoire
n_estimators = np.linspace(10, 200, num = 20, dtype = "int64").tolist()

grille = {'bootstrap':bootstrap,
          'max_depth':max_depth,
          'max_features':max_features,
          'min_samples_leaf':min_samples_leaf,
          'min_samples_split':min_samples_split,
          'n_estimators':n_estimators}

# Choix aléatoire parmi 2*10*3*4*3*20=14400 combinaisons de paramètres possibles
```

Test sur 100 set choisis random et sors meilleur set sur les 100 testés

RandomForest Classifier

Etablissement d'une grille de set à tester puis test avec k-fold(k=5)

```
{'bootstrap': True,  
 'max_depth': 60,  
 'max_features': 'sqrt',  
 'min_samples_leaf': 2,  
 'min_samples_split': 10,  
 'n_estimators': 160}
```

```
# Plus précis autour des meilleurs paramètres random -> 576 sets  
  
best_grille = {  
    'bootstrap' : [True],  
    'max_depth' : [40, 50, 60, 70, 80, 90],  
    'max_features': ['sqrt', 'log2'],  
    'min_samples_leaf': [1, 2],  
    'min_samples_split': [1, 5, 10, 20],  
    'n_estimators' : [150, 160, 170, 180, 190, 200]  
}
```

Test 576 set avec Cross Validation 5-folds

RandomForest Classifier

Base rf :
Accuracy : 0.6666666666666666

Rapport :

	precision	recall	f1-score	support
1.0	0.59	0.76	0.67	17
3.0	0.76	0.59	0.67	22
accuracy			0.67	39
macro avg	0.68	0.68	0.67	39
weighted avg	0.69	0.67	0.67	39

Meilleur random rf :
Accuracy : 0.7948717948717948

Rapport :

	precision	recall	f1-score	support
1.0	0.76	0.76	0.76	17
3.0	0.82	0.82	0.82	22
accuracy			0.79	39
macro avg	0.79	0.79	0.79	39
weighted avg	0.79	0.79	0.79	39

Best rf :
Accuracy : 0.8461538461538461
Matrice de confusion :
[[15 2]
[4 18]]
Rapport :

	precision	recall	f1-score	support
1.0	0.79	0.88	0.83	17
3.0	0.90	0.82	0.86	22
accuracy			0.85	39
macro avg	0.84	0.85	0.85	39
weighted avg	0.85	0.85	0.85	39

A venir:

- Tester XGBoost sur l'ensemble des données avec paramètres opti
- Comparer avec Random Forest