Université du Québec à Montréal (UQAM)
Faculté des sciences

ACT3035– Examen Final (SAS-Python)
Laboratoire d'actuariat

Enseignant: Noureddine Meraihi

2019/12/11

Nom: _____

Code permanent: _____

Signature: _____

Cet examen contient 7 pages (incluant la page couverture) et 4 questions sur un total de 18 points. Bon succès à tous et joyeux temps des fêtes!

Distribution des points

Question	Points	Score
1	1	
2	2	
3	9	
4	6	
Total:	18	

Instructions

- Les notes du cours se trouvent ici: https://nour.me/act3035/
- ou là : https://nbviewer.jupyter.org/github/nmeraihi/ACT3035/tree/master/AUT_2018/
- L'examen commence à 9:00 pour une durée de 180 minutes;
- Prévoyez un 5 minutes pour la remise de votre examen;
- Chaque minute de retard vous coûtera 5% de cet examen;
- Vous avez le droit de compléter votre examen sur SAS 9.4 du laboratoire, ou SAS *University*
- Vous avez le droit de consulter vos notes de cours personnelles;
- Vous avez le droit de consulter l'aide de SAS *help*;
- Il est strictement interdit de faire des recherches sur le web (tous vos logs sur les ordis de l'UQAM sont sauvegardés);
- Il est strictement interdit d'utiliser un quelconque moyen de communication pendant l'examen (logs sauvegardés);
- Pour toutes les questions, le terme **df** désigne *data frame*
- N'oubliez pas de sauvegarder aussi souvent que possible (Ctrl+s)!
- Le nom de votre fichier doit contenir votre code permanent, par exemple: BRUH123456.sas pour la partie SAS;
- L'examen compte pour 50% de la note finale;

QUESTIONS SAS

Pour la question (1) et (2), écrivez votre réponse dans votre script de réponse `.sas` à l'endroit désigné sous forme de commentaire. Écrivez une courte phrase afin d'argumenter votre réponse (`/* votre réponse ....*/`).

1. (1 point) Comment peut-on limiter les variables écrites dans un jeu de données de sortie dans `DATA STEP`?

```
A. DROP
B. KEEP
C. RETAIN
D. VAR
E. A ou B
F. A, B ou C
```

2. (2 points) Lorsque l'on exécute le code SAS ci-dessous sur la base de données `EMP` présentée à la figure (1), combien d'observations seront affichées?

```
proc print data = emp;
     where Name like '_R%';
run;
```

Obs	DOB	Employee_id	Gender	Name	Location	Salary	Manager_Emp_ID
1	12/01/1995	101	M	John	Delhi	350000	101
2	07/04/1980	102	F	Sangeeta	Delhi	450000	103
3	03/05/1973	103	F	Mary	Mumbai	500000	101
4	06/25/1975	104	M	Richard	Mumbai	750000	101
5	08/20/1990	105	M	Fredrick	Delhi	320000	101

Figure 1: Base de données `EMP`

NB: Cette base de données n'est pas dans la trousse d'examen, vous pouvez la créer manuellement (si vous avez le temps) s'assurer.

PROC SQL

Misen en context:
En surfant sur le net, vous mettez la main sur une base de données d'un magasin en ligne du type Amazon. Comme l'idée de démarrer, votre propre entreprise est toujours présente dans votre tête, vous décidez alors de répliquer le même modèle d'affaires à petite échelle, mais vous voulez utiliser votre talent de modélisateur que vous avez acquis durant votre baccalauréat en actuariat. Ainsi, votre modèle d'affaires consiste à vous concentrer sur les produits les plus vendus, vous aurez alors un avantage par rapport à vos concurrents, car votre marge de bénéfice est plus petite à cause de la quantité astronomique que vous vendez chaque jour.

Les données:
Afin de résoudre les prochaines questions, vous devez travailler avec une base de données contenant quatre fichiers `.csv`. Voici la description des variables dont vous aurez besoin;

- ID - est l'identifiant d'une variable quelconque
- shop_id - un identifiant unique d'un magasin (*shop*)
- item_id - un identifiant unique d'un produit
- item_category_id - un identifiant unique d'une catégorie
- item_cnt_day - Nombre de produit vendus.
- item_price - le prix courant d'un item
- date - date en format format dd/mm/yyyy
- item_name - nom de l'item
- shop_name - nom du magasin
- item_category_name - catégorie de l'item

3. (a) (2 points) Importez les quatre fichiez `.csv` suivants dans la librairie `work`
   - `sales_train.csv` appelé `salesdata`
   - `shops.csv` appelé `shops`
   - `items.csv` appelé `items`
   - `item_categories.csv` appelé `item_categories`

   (b) (3 points) Créer une table appelée `salesdata1` qui joint les deux tables `salesdata` et `shops`, où l'on trouve toutes les variables de `salesdata` et seulement la variable *shop_name* de la table `shops`

(c) (1 point) Créez une nouvelle table appelée `salesdata2` qui joint la table `salesdata1` (de la question précédente) et la table `items`.

(d) (3 points) Créer une table appelée `salesdataFinal` qui joint la table `salesdata2` et la table `item_categories`. Si votre code est correct, vous devriez obtenir une table telle qu'illustrée à la figure (2)[1].

Columns

Select all

- date
- date_block_num
- shop_id
- item_id
- item_price
- item_cnt_day
- shop_name
- item_name
- item_category_id
- item_category_name

Total rows: 2935849 Total columns: 10

	date	date_block_num	shop_id	item_id	item_price
1	02/04/2013	3	38	16256	22.4
2	10/01/2013	0	25	16257	148
3	28/02/2013	1	0	16255	93
4	20/02/2013	1	0	5740	283
5	10/01/2013	0	45	5606	148
6	06/09/2013	8	10	5572	1322
7	27/12/2014	23	27	5643	3290
8	20/12/2013	11	31	5573	449
9	23/01/2014	12	2	5637	2490
10	16/01/2013	0	1	5575	806
11	24/04/2015	27	52	5643	2990
12	11/04/2015	27	56	5638	3290
13	12/01/2014	12	5	5573	449
14	23/02/2013	1	41	5575	1090

Figure 2: Résultat de la table `salesdataFinal`

[1]Il est possible que votre tableau soit différent que celui à la figure (2) si vous avez un tri sur vos données

QUESTIONS `Python`

Pour cette question, vous pouvez tester votre code dans Jupyter notebook, un script python ou n'importe quel autre interface de développement. Une fois que vous avez terminé votre code, vous pouvez simplement le coller dans votre script de réponse `.sas` à l'endroit désigné, ou déposer votre script python (ou votre notebook jupyter).

NB Les points ne sont pas accordées qu'au résultat final, mais l'ensemble de vos démarches

4. La performance prédictive de ces modèles est évaluée à l'aide de règles de scores pour les données de comptage. Les règles de scores évaluent la qualité des prédictions probabilistes à l'aide d'un score numérique $s(P, n)$ basé sur la distribution prédictive P et le nombre n observé. Des scores plus faibles indiquent une meilleure qualité des prédictions.

   Une des règles est appelée *Dawid–Sebastiani* définie par la fonction ci-dessous où la moyenne et l'écart-type de P sont écrits comme μ_p et σ_p respectivement.

$$\text{dss}(P, n) = \left(\frac{n - \mu_p}{\sigma_p}\right)^2 + 2\log(\sigma_p)$$

   (a) (1 point) Insérez les données `ObsPred.csv` à l'intérieur d'un *data frame* appelé `df` en utilisant la bibliothèque `Pandas`.

   (b) (2 points) Calculez ce score pour chaque observation du jeu de données `ObsPred.csv`

   (c) (3 points) Écrivez une fonction appelée `dss` qui permet de calculer la somme des scores *Dawid–Sebastiani* sur l'ensemble des données. Votre fonction doit prendre deux arguments; les observations (n) ainsi que les prédictions obtenues par votre modèle μ_p du jeu de données `ObsPred.csv`.

Fin de l'examen

- N'oubliez pas de renommer le gabarit de réponse BRUH123456.sas
- téléchargez votre fichier de réponses comme à la figure (3).

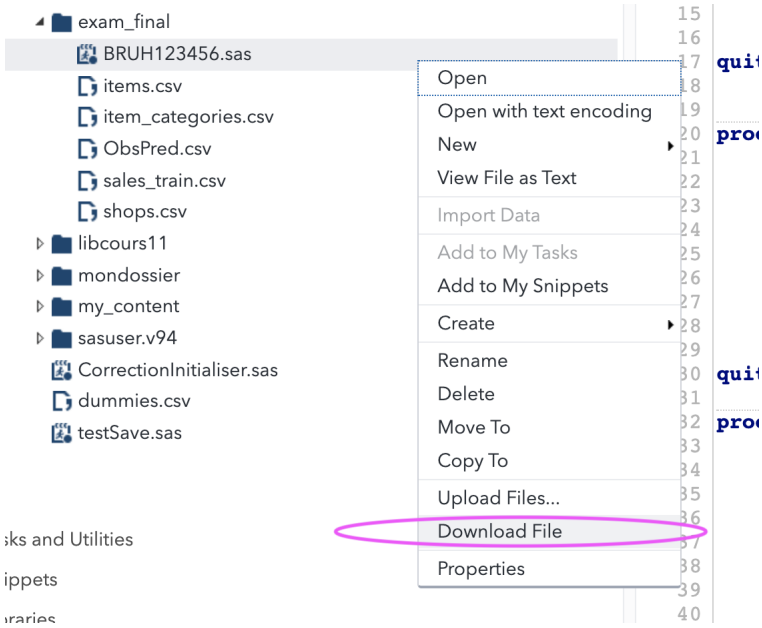

Figure 3: Résultat de la table `totalS`

- Déposez votre examen final à l'adresse: https://bit.ly/36k2v6X pas plus tard que la minute indiquée. Le dépôt de l'examen sera fermé à l'heure:minute indiquée.
- Vous serez avisé par courriel quand les notes seront disponibles.