

L'UTILISATION DE PYTHON DANS L'ACTUARIAT

MOTS CLES : ACTUARIAT - MODELISATION - LANGAGE DE PROGRAMMATION - MACHINE LEARNING - DATA SCIENCE - IA- PYTHON

Selon les indices TIOBE¹ et PYPL², Python est aujourd'hui le langage de programmation le plus populaire au monde. Son essor, y compris dans le secteur assurantiel, illustre la transformation numérique en cours : gestion des données, modélisation, automatisation des processus et intelligence artificielle deviennent des composantes centrales du métier d'actuaire et font de Python un levier stratégique pour l'actuariat. De fait, de nombreux acteurs de l'assurance migrent ou envisagent de migrer leurs solutions propriétaires vers Python et des plateformes data modernes, ce qui impacte les processus actuariels et redéfinit les pratiques de place.

LES FORCES DE PYTHON

UNE ACCESSIBILITÉ REMARQUABLE

Python est reconnu pour sa simplicité d'apprentissage. Enseigné dès le lycée et largement utilisé dans l'enseignement supérieur, il est déjà familier à de nombreux jeunes professionnels.

En effet, guidé par le *Zen of Python*³ qui prône la simplicité et la clarté, son caractère interprété (exécuté à la volée) et à typage dynamique (souplesse sur le type de variable) le rend très flexible et facilite le développement rapide.

UNE COMMUNAUTÉ MONDIALE ACTIVE

Python est le langage le plus utilisé sur GitHub. Il bénéficie d'une communauté internationale extrêmement dynamique qui enrichit en permanence son écosystème open source. Des milliers de bibliothèques et de solutions sont disponibles pour répondre à des besoins variés. Pour presque chaque problème rencontré, une solution existe déjà en ligne.

UN AVANTAGE ÉCONOMIQUE NON NÉGLIGEABLE

Open source et entièrement gratuit, Python réduit significativement les coûts liés aux licences logicielles. Pour les entreprises, cela représente un levier d'optimisation budgétaire important, en

particulier face à des solutions propriétaires coûteuses.

UN ÉCOSYSTÈME RICHE ET ADAPTÉ AUX BESOINS ACTUELS

Au-delà de sa simplicité, Python séduit par la richesse de son écosystème. Ses milliers de bibliothèques couvrent tous les champs de la donnée : manipulation (*Numpy*, *Pandas*), visualisation (*Matplotlib*, *Seaborn*), calcul scientifique (*SciPy*), *machine learning* (*scikit-learn*) ou *deep learning* (*TensorFlow*, *Pytorch*). Cette diversité permet d'adresser des besoins très différents avec un seul langage, qu'il s'agisse d'analyser des bases de données volumineuses, de déployer des modèles prédictifs ou de concevoir des applications opérationnelles.

UNE INTEROPÉRABILITÉ FACILITÉE

Grâce à ses packages spécialisés, Python s'intègre facilement avec les bases de données (SQL, NoSQL), les environnements cloud et les outils historiques. Les modèles développés peuvent aussi être interfacés avec Excel, qui demeure une interface privilégiée pour le reporting actuariel. Cette compatibilité facilite l'adoption et le déploiement en production.

¹ <https://www.tiobe.com/tiobe-index/>

² <https://pypl.github.io/PYPL.html>

³ [PEP 20 – The Zen of Python | peps.python.org](https://peps.python.org/PEP-20/)

UNE ADOPTION MASSIVE QUI EN FAIT UNE NORME TECHNOLOGIQUE

La flexibilité et la polyvalence de Python expliquent son adoption massive dans de nombreux secteurs, dont l'assurance. Les plus grands acteurs du numérique l'utilisent en production au cœur de leurs systèmes et services. Cette diffusion mondiale contribue à en faire une norme technologique, rassurante pour les entreprises qui choisissent de l'adopter à leur tour.

PYTHON ET L'IA GÉNÉRATIVE

Python occupe une place centrale dans l'IA moderne. Il est au cœur des modèles génératifs (comme ChatGPT) qui transforment les métiers de la donnée et du chiffre. Cette dimension attire de nouveaux talents et stimule l'innovation dans le secteur de l'assurance.

Enfin, les performances des IA génératives actuelles permettent désormais d'accélérer le codage à l'aide de l'assistance de l'IA et de réduire la barrière technique pour les débutants.

Python allie à la fois puissance, richesse d'écosystème et souplesse. En contrepartie, son utilisation exige des vérifications rigoureuses du code et une bonne maîtrise de ses limites, notamment en calcul intensif.

L'UTILISATION DE PYTHON EN ACTUARIAT

GESTION DES DONNÉES

En actuariat, Python s'impose comme un allié naturel de la gestion de données. Sa flexibilité permet de traiter une grande variété de formats et de construire des pipelines robustes pour automatiser les tâches répétitives et ETL (Extract-Transform-Load). Les actuaires s'appuient sur des bibliothèques spécialisées pour manipuler efficacement des volumes massifs, connecter Python aux bases de données SQL ou NoSQL et intégrer leurs traitements dans des environnements cloud.

Mais cette puissance implique une vigilance : un mauvais usage des ressources mémoire, une dépendance mal gérée entre bibliothèques ou un environnement mal configuré peuvent

rapidement devenir des freins de performance. La clé réside dans l'adoption de bonnes pratiques de codage et dans une gouvernance technique solide.

MODÉLISATION ACTUARIELLE

Python se distingue également dans la modélisation actuarielle par sa capacité à produire des outils sur mesure. Qu'il s'agisse de générer des scénarios économiques, de réaliser des projections ALM, de calculer des provisions ou d'automatiser le reporting, il permet de concevoir des solutions adaptées à chaque besoin. La possibilité de connecter directement les modèles à des outils comme Excel facilite la diffusion des résultats vers les utilisateurs finaux.

Les actuaires bénéficient aussi de la puissance du calcul parallèle et du calcul sur GPU pour réduire les temps de traitement. Les notebooks, enfin, offrent un cadre de travail interactif idéal pour tester et suivre pas à pas la construction des modèles et le prototypage.

Cependant, cette flexibilité a une contrepartie : elle exige une rigueur accrue. Pour garantir la pérennité d'un modèle, il faut structurer le code avec rigueur, mettre en place un versioning, documenter précisément et appliquer des standards de qualité. Faute de discipline, le modèle risque de rester une "boîte noire" inutilisable à long terme.

DATA SCIENCE ET INNOVATION

Python est aujourd'hui le langage de référence pour la data science et l'intelligence artificielle. Il permet d'intégrer des modèles prédictifs dans les processus actuariels, qu'il s'agisse de tarification, de détection de fraude ou de prédiction des résiliations. Ses bibliothèques de pointe en *machine learning* et *deep learning* s'accompagnent d'outils collaboratifs qui facilitent le partage de code, la gestion des environnements et la reproductibilité.

Au-delà de la performance, Python donne accès à des solutions d'explicabilité et de monitoring qui renforcent la transparence, l'explicabilité et l'audit des modèles, un enjeu essentiel dans le secteur de l'assurance soumis à de fortes contraintes réglementaires (IA Act, RGPD). Ces

capacités stimulent l'innovation et rendent la profession plus attractive pour les talents. Mais elles impliquent un effort constant de formation et une gouvernance exigeante, sans quoi le risque est d'accumuler une dette technique qui fragilise les projets.

Python permet de traiter des problématiques directement liées au cœur du métier actuariel et ouvre également la voie à une nouvelle génération de travaux actuariels intégrant les enjeux de durabilité et d'équité : modélisation des risques climatiques et ESG, évaluation de la double matérialité, contrôle des biais dans les modèles de tarification et développement d'algorithmes explicables conformes à l'AI Act.

LA MIGRATION VERS UN ÉCOSYSTÈME PYTHON

Le secteur de l'assurance connaît une transition progressive de logiciels propriétaires tels que SAS vers Python et des plateformes data modernes comme Databricks, Dataiku, Snowflake ou JupyterHub. Ce mouvement est accéléré par l'essor des solutions cloud (AWS, Azure), qui apportent puissance de calcul, ouverture multi-langages et de nouvelles possibilités de collaboration.

La réussite d'une migration repose sur trois grandes étapes.

LA PHASE PRÉPARATOIRE

Avant toute chose, il est essentiel de poser les bases du projet. Cela passe par l'identification des cas d'usage les plus pertinents et la cartographie des programmes et des données existantes, pour comprendre leur structure et leur complexité. C'est aussi le moment de définir la gouvernance (gestion des accès, responsabilités) et de mettre en place les environnements de développement, test et production, ainsi que la gestion des bibliothèques. Cette phase permet d'anticiper les obstacles techniques et organisationnels, en impliquant dès le départ les équipes IT.

LA PHASE DE MIGRATION

Une fois les fondations posées, vient le cœur de la transformation : réécrire la logique des traitements et calculs existants et développer les nouvelles solutions en Python. Ce code est ensuite structuré en pipelines automatisés pour garantir sa robustesse et sa maintenabilité. Les tâches doivent être hiérarchisées pour faciliter le passage progressif d'un monde à l'autre.

Dans ce processus, les outils collaboratifs jouent un rôle clé : ils assurent la traçabilité, l'industrialisation et la qualité des développements. Les résultats doivent ensuite être validés grâce à des critères rigoureux, par exemple via des phases de "double run", où les anciens codes et leurs équivalents Python sont exécutés en parallèle pour valider les outputs.

LA PHASE POST-MIGRATION

Le travail ne s'arrête pas une fois la migration technique réalisée. Il faut industrialiser les opérations, orchestrer les workflows, assurer la maintenance et le monitoring des traitements. La gestion des dépendances devient un enjeu crucial : bibliothèques, compatibilités et mises à jour doivent être contrôlées pour garantir la stabilité du nouvel environnement.

Enfin, la réussite passe par la conduite du changement : former activement les équipes, documenter les pratiques, diffuser les standards et accompagner l'adoption. C'est cette appropriation par les utilisateurs qui permet de capitaliser pleinement sur le nouvel écosystème et d'en tirer tous les bénéfices.

La migration vers un environnement Python doit aussi répondre aux exigences de sécurité et de conformité du secteur : traçabilité des calculs, auditabilité des modèles, gestion rigoureuse des accès et maîtrise des dépendances logicielles.

CONCLUSION

En adoptant Python, les assureurs ne se dotent pas seulement d'un outil moderne, mais d'un véritable langage commun entre data scientists, développeurs et actuaires — une passerelle essentielle pour accélérer l'innovation et renforcer la résilience du secteur.

Une migration structurée, qui associe des compétences en développement à l'expertise métier des actuaires, et qui s'accompagne d'une conduite du changement rigoureuse, permet de tirer pleinement parti des atouts de Python tout en maîtrisant ses limites.

Les actuaires occupent une place centrale dans cette transformation : leur connaissance des modèles et des données garantit la cohérence entre les développements Python, les exigences métier et les besoins de pilotage des risques

Les consultants du DataLab de GALEA sont à votre disposition pour plus de précisions sur le contenu de cette note et pour vous accompagner dans la migration vers un écosystème Python, la mise en place de projets développés en Python ou la formation à l'utilisation de cet outil puissant.



<https://www.galea-associes.eu/>