

# Hackathon 主题五：报销异常检测与防欺诈系统

## 核心需求

构建一个报销管理系统，能够自动审查报销申请，检测异常和潜在的欺诈行为，并为财务人员提供风险提示。系统需要能够学习正常的报销模式，识别偏离模式的异常申请，并根据配置的规则自动批准、拒绝或标记为需要人工审核。

## 使用场景示例

### 报销数据分析：

- 系统接收员工的报销申请（包括金额、类别、日期、部门等信息）。
- 系统分析员工的历史报销模式。
- 系统识别该申请是否符合员工的正常行为。

### 异常检测：

- 金额异常：**检测报销金额是否异常高或异常低。
- 频率异常：**检测报销频率是否异常高。
- 时间异常：**检测报销是否在异常的时间提交（如深夜、周末）。
- 类别异常：**检测报销类别是否与员工的职位或部门相符。
- 地点异常：**检测报销地点是否与员工的工作地点相符。

### 欺诈风险检测：

- 检测是否存在重复报销（同一笔费用被多次报销）。
- 检测是否存在虚假发票（如发票号码格式不对、发票日期不合理等）。
- 检测是否存在团伙欺诈（如多个员工提交相似的异常报销）。
- 检测是否存在金额篡改迹象。

### 风险评分与处理：

- 系统为每份报销申请生成一个风险评分。
- 系统根据风险等级自动执行相应的处理（如自动批准、自动拒绝、标记为需要审核等）。
- 系统生成审核报告，列出所有异常点和建议。

## 功能需求

- 系统能够接收和处理报销申请数据（可以是CSV、JSON、数据库等格式）。
- 系统能够学习和建立每个员工的正常报销模式。
- 系统能够实时检测异常的报销申请。

- 系统能够识别多种类型的欺诈风险。
- 系统能够根据配置的规则自动执行相应的处理。
- 系统能够生成详细的审核报告和统计数据。
- 系统应该提供一个管理界面，允许财务人员查看报销申请、异常点和处理建议。
- 系统应该能够记录所有处理过程，便于审计和追踪。

## 数据源

- 模拟报销数据：**参赛者可以生成模拟的报销申请数据，包括正常申请和欺诈申请。
- 历史数据：**可以使用公开的或模拟的员工报销历史数据。
- 规则配置：**参赛者需要设计和实现规则定义机制。

## 评分标准

评分维度	权重	关键考察点
异常检测准确性	40%	- 系统能否准确检测异常报销? - 误报率是否在可接受范围内? - 能否识别多种类型的欺诈?
风险评分合理性	30%	- 风险评分是否能有效区分高风险和低风险申请? - 评分逻辑是否清晰合理? - 是否能根据反馈改进评分模型?
系统可用性	20%	- 管理界面是否清晰易用? - 报告和建议是否有帮助? - 规则配置是否简洁易懂?
创新与效率	10%	- 是否有创新的检测算法或风险评分方式? - 系统性能如何?

## 提交要求

- 完整的源代码。
- 部署和运行的说明文档。
- 一份演示报告，展示系统在检测异常和欺诈报销时的表现。

## 附录：技术参考

### 数据处理

- **Pandas**: 用于数据处理和分析。
- **NumPy**: 用于数值计算。

### 异常检测算法

- **统计方法**: 如Z-score、IQR等。
- **机器学习方法**: 如Isolation Forest、Local Outlier Factor (LOF)、One-Class SVM等。
- **聚类方法**: 如K-means、DBSCAN等，用于识别异常的报销模式。

### 规则引擎

- **Python-Rule-Engine** 或自己设计简单的规则定义和执行机制。

### 可视化

- **Streamlit / Dash**: 用于构建管理界面。
- **Plotly**: 用于数据可视化和报表生成。

### 数据存储

- **SQLite / PostgreSQL**: 用于存储报销数据和处理历史。