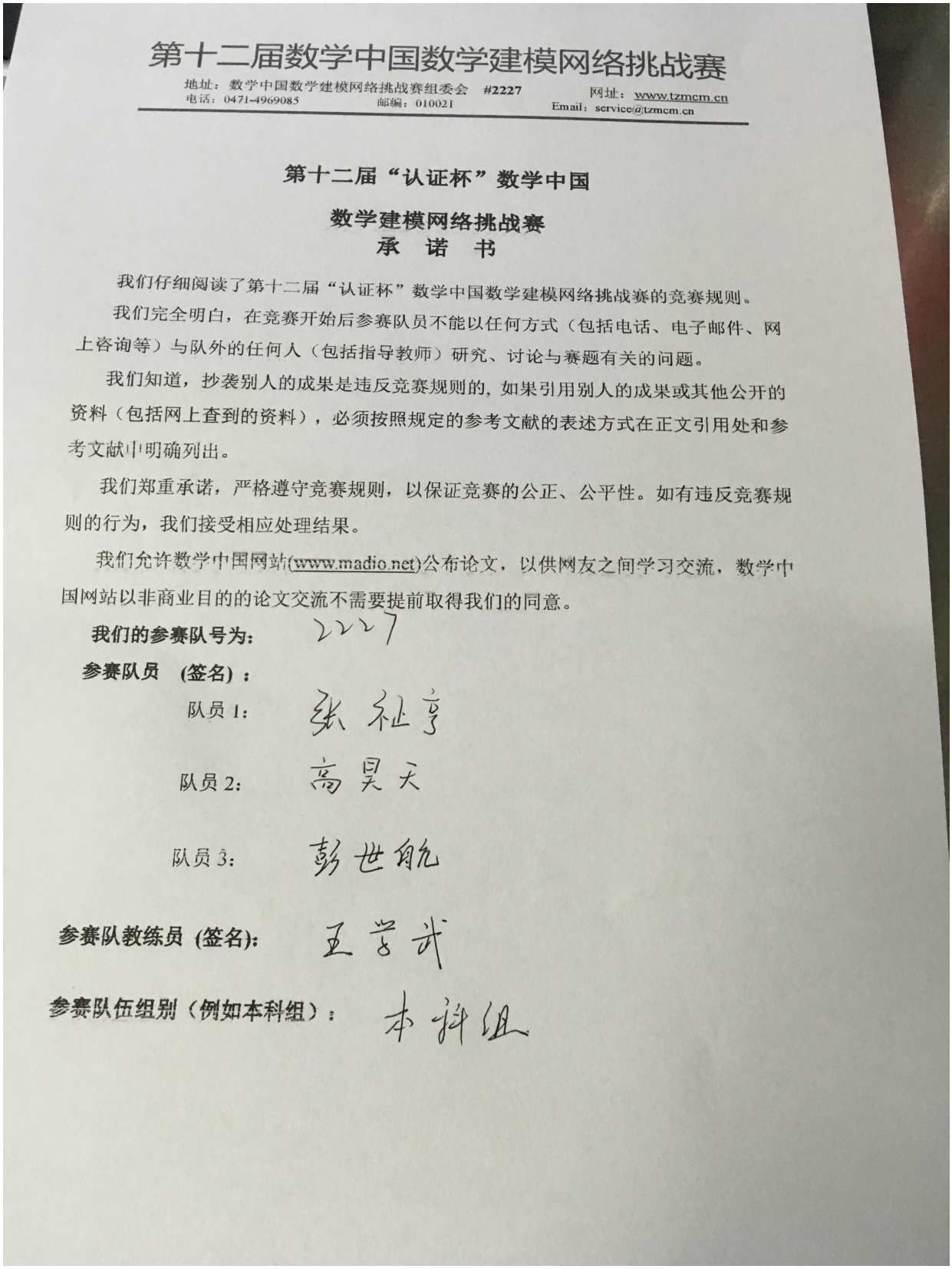

第十二届数学中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会 #2227 网址：www.tzmcm.cn
电话：0471-4969085 邮编：010021 Email：service@tzmcm.cn

第十二届“认证杯”数学中国

数学建模网络挑战赛

承 诺 书

我们仔细阅读了第十二届“认证杯”数学中国数学建模网络挑战赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们接受相应处理结果。

我们允许数学中国网站(www.madio.net)公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

我们的参赛队号为： 2227

参赛队员 (签名)：

队员 1：张社亨

队员 2：高昊天

队员 3：彭世航

参赛队教练员 (签名)： 王学武

参赛队伍组别（例如本科组）： 本科组

第十二届数学中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会 #2227 网址：www.tzmcm.cn
电话：0471-4969085 邮编：010021 Email：service@tzmcm.cn

第十二届“认证杯”数学中国

数学建模网络挑战赛
编 号 专 用 页

参赛队伍的参赛队号：（请各个参赛队提前填写好）：

2227

竞赛统一编号（由竞赛组委会送至评委团前编号）：

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

第十二届数学中国数学建模网络挑战赛

地址：数学中国数学建模网络挑战赛组委会 #2227 网址：www.tzmcm.cn
电话：0471-4969085 邮编：010021 Email：service@tzmcm.cn

2019 年第十二届“认证杯”数学中国数学建模网络挑战赛第二阶段论文

题　　目　保险业的数字化变革

关 键 词　GLM 模型、驾驶习惯评分模型、SAS 数据分析

摘　　要:

车险，即机动车辆保险。可以通过数字化技术来更加精准地了解客户，制定营销和服务方案。本文运用 **GLM 模型、 驾驶习惯评分模型、UBI 车险费率、SAS 数据分析**等方法成功解决了题目所给的问题，建立了合适的、完整的、系统的续保率模型，采用定量、定性分析相结合的方法标定模型参数，以此对车险续保率问题进行处理分析。

针对问题一，为了搜集客户三个方面的驾驶习惯信息。首先基于影响驾驶安全的因素进行指标筛选后得到三个方面的信息：1.车辆驾驶、2.车辆速度、3.车辆操作；接着根据三个方面信息指标建立**驾驶习惯评分模型**；然后厘定了各个指标的相对风险系数并进行了**拟合值-残差检验**，得出了不同驾驶习惯的评分值如下公式；最后通过实际 **APP 软件**测试评分值，使模型更具有真实性和可靠性。

$$S=\frac{S_0}{\prod i}$$

（ S:最终评分，S0:基准评分值，i:各指标变量某一风险水平的相对系数值。）

针对问题二，为了而通过设计驾驶习惯调查问卷来提高续保概率。首先我们对问题一的各项指标作为调查问卷的内容；然后基于问卷形式引入某财险公司的真实数据并进行处理；接着基于各指标作为解释变量建立 **GLM 模型**对每个指标进行建模求解；然后基于 **SAS 软件**对求解后的数据进行结果分析和检验；最后对 **UBI 车险费率**进行计算，根据保费公式进行各指标的调整以提高续保概率，使结果更具有可行性和可行性。

保费=基准纯风险保费/（1-附加费用率）*折扣系数

参赛队号：2227

参赛密码 _____
（由组委会填写）

所选题目：C 题

英文摘要（选填）

（此摘要非论文必须部分，选填可加分，加分不超过论文总分的 5%）

Vehicle insurance is motor vehicle insurance. Through digital technology, we can understand customers more accurately and make marketing and service plans. In this paper, **GLM model, driving habits scoring model, UBI vehicle insurance premium rate, SAS data analysis** and other methods are used to successfully solve the problem. A suitable, complete and systematic renewal rate model is established, and the model parameters are calibrated by combining quantitative and qualitative analysis to deal with the renewal rate of vehicle insurance.

To solve the first problem, in order to collect three aspects of customer driving habits information. Firstly, based on the factors affecting driving safety, three aspects of information are obtained: 1. Vehicle driving, 2. Vehicle speed and 3. Vehicle operation; secondly, **driving habits scoring model** is established according to three aspects of information indicators; secondly, relative risk coefficients of each index are **determined and fitted value-residual test** is carried out, and the scoring values of different driving habits are obtained as follows; Finally, the model is more authentic and reliable by testing the score value of the actual **APP software.**

$$S=\frac{S_0}{\prod i}$$

(S: Final score, S0: Benchmark score, i: Relative coefficient value of each index variable at a risk level.)

To solve the second problem, In order to improve the probability of renewal of insurance, a questionnaire on driving habits was designed. Firstly, we use the indicators of Question 1 as the content of the questionnaire, then introduce the real data of a property insurance company and process them based on the form of the questionnaire, then build a **GLM model** based on the indicators as explanatory variables to solve each index, then analyze and test the data after solving based on **SAS software**, and finally calculate the **UBI vehicle insurance premium rate.** According to the premium formula to adjust the indicators to improve the probability of renewal, so that the results are more feasible and feasible.

Premium = Benchmark pure risk premium/(1-surcharge rate)*Discount factor

一、问题重述

1.1 引言

车险，即机动车辆保险。车险即为分散机动车辆在行驶过程中可能发作的未知风险和损失的一种保障机制信息时代的到来，为车险企业提供了一个更加有力的武器，可以通过数字化技术来更加精准地了解客户，制定营销和服务方案。

1.2 问题的提出

围绕对这些数据，请你们团队完成下列三项任务：

任务 1：为了能够提高销售效率，降低客户等待时间，每次销售只能搜集客户三个方面的驾驶习惯信息。请结合第一阶段问题建立数学模型，讨论我们应该搜集哪三个方面的信息。

任务 2：请结合前面的问题设计调查问卷，并建立数学模型阐述如何利用这个问卷的数据来提高续保概率。

任务 3 给保险公司的 CEO 写一封 1-2 页长短的信，陈述你对车险业在大数据环境下应如何发展的建议。

二、模型的假设

- 假设题目所给的数据真实可靠;
- 只考虑目标客户的影响;
- 分析不同驾驶习惯因素对续保率的影响时，假设选取的各因素之间没有显著相关性;
- 分析不同驾驶习惯因素对续保率的影响时，假定其他因素不变，以确保能准确计算单一因素对续保率的影响程度;

三、符号说明

符号	意义
$V(\mu_i)$	方差函数
ω	先验权重
r_i	Pearson 残差
S_0	基准评分值
$Y_{abcde}^{`}$	平均索赔额
m_{abcde}	索赔次数
H	二阶导数矩阵
β	指数化的值
y_i	概率密度函数

注：这里只列出了部分符号，未列出的符号及重复的符号以出现处为准

四、问题分析

问题一中为了能够提高销售效率，搜集客户三个方面的驾驶习惯信息，从车辆驾驶方面、车辆速度方面、车辆操作方面进行分析，然后通过对驾驶习惯评分指标体系构建广义线性模型以便对问题二做进一步分析具体车险的计算。

问题二中结合前面的问题设计驾驶习惯调查问卷，并建立数学模型阐述如何利用这个问卷的数据来提高续保概率，我们将续保概率转换为另一种方式————UBI 车险费率厘定。利用广义线性模型建立的评分模型对 UBI 车险的风险进行厘定分级。所建立的驾驶习惯评分模型进而厘定得到 UBI 车险的保费计算来进行对提高续保概率的的预测。

本文的研究思路如下图所示：

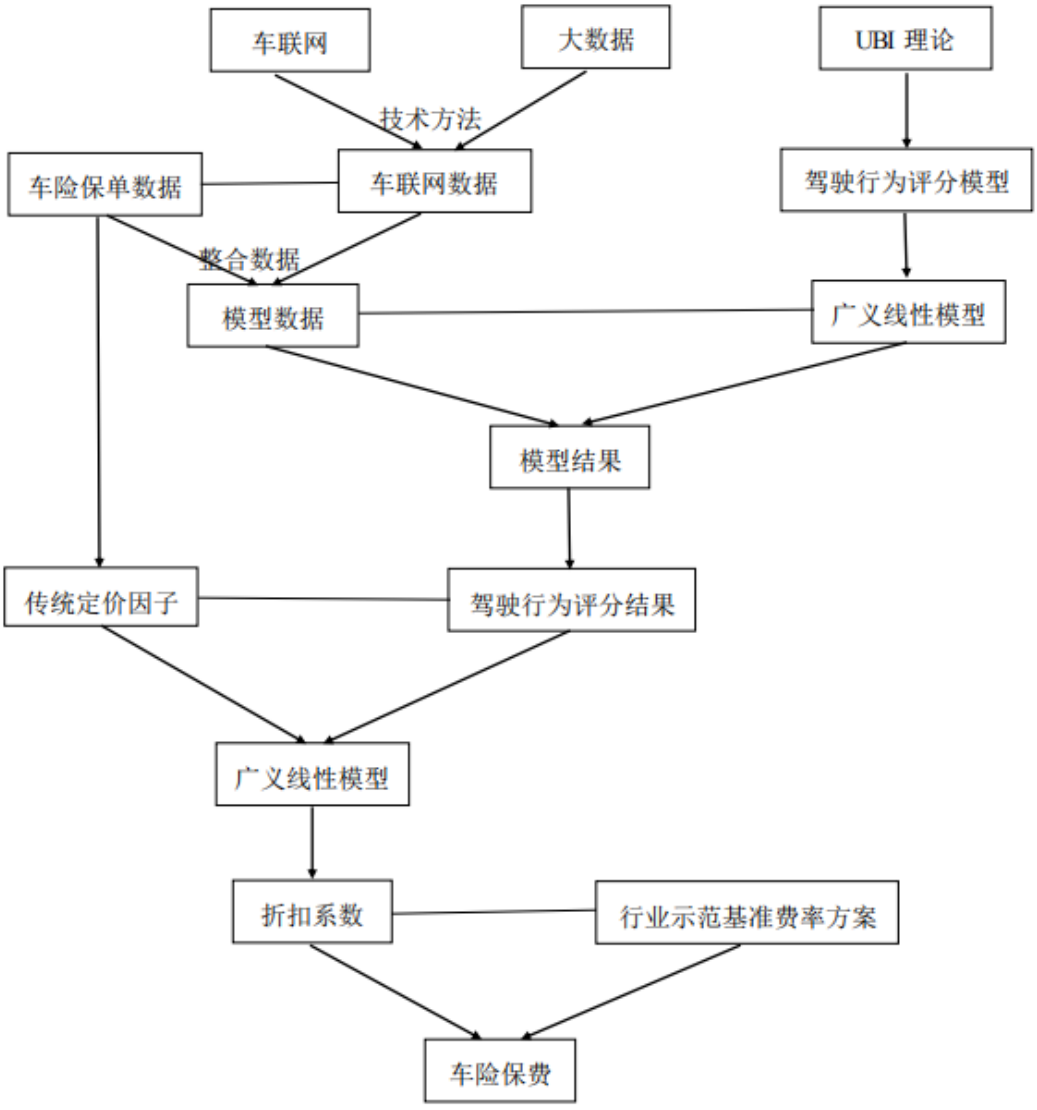

图 **1**　研究思路

五、模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 影响驾驶安全的基本因素

[1] 本文我们主要从行驶里程、出行时间（夜间驾驶和驾驶时长等）、速度（超速行驶和相对速度等）、加速度（急加速和急减速等）、刹车（急刹车等）和转弯（急转弯）等方面考察驾驶习惯对驾驶安全的影响并作分析。

上面的指标可从三个驾驶习惯方面的信息进行概括：

1、车辆驾驶方面: 行驶里程、出行时间（夜间驾驶和驾驶时长等）。

2、车辆速度方面: 速度（超速行驶和相对速度等）、加速度（急加速和急减速等）。

3、车辆操作方面：刹车（急刹车等）和转弯（急转弯）。

以下我们对这几种行为做具体分析：

（1）行驶里程

行驶里程是国际上很多知名的 UBI 保险公司都会纳入驾驶习惯考察范围内的因素，而且不论是第一代的 PAYD（Paid As You Drive）还是第二代的 PHYD（Paid How You Drive），行驶里程都是他们着重考虑的因素。当然其中一个主要的原因是行驶里程是比较容易获得的一项车联网数据且精度较高。但另一方面，由于行驶里程的增加，机动车辆的老化磨损程度会越来越严重，整个机动车辆的状况就比较差，那么就会更加容易造成交通事故。最近有不少的研究表明，行驶里程与交通事故的强关联性，大概行驶里程每增加 1%，交通事故的损失率就会增加 50% 以上。

（2）出行时间

出行时间，尤其是其中的夜间行驶时间是诸多 UBI 保险公司所考察的范围，一定程度上也能说明出行时间对风险的影响的显著性。夜间驾驶和驾驶时长都是导致驾驶员疲劳驾驶的主要因素。其中，夜间驾驶的时候由于光线较差，而且夜间驾驶员难以保持高度的注意力，因此夜间的事故发生率也是相对较大；驾驶时长是导致驾驶疲劳的最主要因素，连续长时间驾驶会导致精神困乏、注意力不集中的状态，这种状态下人的反应能力很差，甚至有危险的驾驶习惯出现，因此驾驶时长也是造成事故发生的重要因素。综上可见出行时间是重要的风险因素。

（3）速度

速度是机动车辆行驶中的重要指标，它同样也是影响安全驾驶的重要因素。其中，超速行驶（Over Speeding）是我们常识中所知道的一种危险的驾驶习惯，超速行驶是指行驶过程中行驶的速度超过路段规定的时速的行驶时速。一旦机动车辆进入超速行驶，那么车辆的稳定性和驾驶的把握性都很差，故而事故的发生率及事故的危险程度也会随之成倍增大。诸多研究表明，随着车速的提高，驾驶员的视野会变得狭窄，辨别事物及分析事物的能力随之下降，导致反应能力变差，从而极易引发交通事故。另外，相对速度是指在同一路段同一时段行驶的机动车辆的速度相对大小的差异，而超速行驶是其中的特殊情况，相对平均速度过快过慢都是相对速度有较大差异的情况，该种情况也是导致交通事故发生的重要因素。

（4）加速度

加速度同样是机动车辆行驶中的重要性能指标，当加速度绝对值大于某一临界值时，往往意味着车辆正在进入急加速（Sudden Acceleration）或者急减速（Sudden Deceleration）的阶段。首先，当车辆间的间距较小的时候，急加速或者急减速极易造成前后车辆的追

尾事故；其次，急加速急减速这种不良的驾驶习惯容易导致机动车辆的故障，甚至造成车辆不能正常的行驶，这在很多情况下是极其危险的。

（5）刹车

急刹车（Sharp Braking）是一种很糟糕的驾驶习惯习惯，造成事故的原理和急减速类似，但是急刹车相对急减速往往造成的危害程度更大。一般出现急刹车的交通事故，不管是否是有急刹车引起的都很大概率上是重大交通事故。

（6）转弯

转弯往往实在路口进行，那么路口是相比直行更多更易发生交通事故的路段，尤其是当驾驶员进行急转弯（Risky Steering）的时候，对整个交通路段的影响是非常恶劣的。

5.1.2 驾驶习惯评分指标体系构建

（1）驾驶习惯评分指标的引入

首先，不论是智能手机 APP 还是车载终端 OBD 盒子，其对各项驾驶习惯指标的感知和识别都是我们用来分级驾驶习惯风险数据的基础，只有通过相关车险网技术和大数据技术做到感知和识别的指标才能够为我们所用。根据笔者所参与项目所采用的数据采集技术，采集到的指标有行驶里程、行程时间和行驶时间、怠速时间、平均时速及最高时速、急加速次数、急刹车次数、急转弯次数等。在这些所采集到的指标的基础上，我们还要依据交通运输相关的法律法规进行判断某项指标处于何种档次或水平，进而为厘定驾驶习惯风险做好准备。

（2）各指标的解释

1、行驶里程和行驶时间

驾驶员的行驶里程和行驶时间指标包括月行驶里程、月行驶时长、单次行程行驶时长以及夜间行车时间等。首先，月行驶里程指的是机动车辆该月总的行驶里程数。其次，月行驶时长指的是机动车辆在该月总的行驶时间长度，而单次行程行驶时长指的是一次完整的行程从启动到熄火的时间间隔，这两项指标综合起来可以判断一个驾驶员是否有疲劳驾驶以及疲劳驾驶的程度，而疲劳驾驶是威胁交通安全的重要因素。夜间行使时间指的是机动车辆每天夜间时段（18：00 至次日 6：00）的行驶时间。夜间行使时间指标从两个方面反映了驾驶员疲劳驾驶的情况，包括其会一定程度上影响驾驶员的视线以及其精神状况也相对较差。

2、超速行驶

根据我们目前所能采集到的数据，其中包括不同速度区间下的行驶时间，因此我们可以超速行驶划分几个不同的档次，比如车辆时速在 60-80km/h、时速在 80-100km/h、时速在 100-120km/h 以及时速在 120km/h 以上，同时考察这些区间车辆的行驶时间以及其占比情况。

3、急加速急刹车急转弯

急加速急刹车急转弯是考察驾驶员驾驶习惯的重要指标，因为它们反映了驾驶员驾驶技术的高低，进而反映其驾驶风险的高低，急加速急刹车急转弯不但会给车辆本身带来一定程度的磨损，时间一长就会出险一些风险，而且这些不良的行为会影响其他驾驶员的判断，进而造成整个运行中的交通处于一个高风险的状况下。而对急加速急刹车急转弯的判断是通过正向加速度、反向加速度和侧向加速度高于或低于某一临界值时，我们则认为该机动车辆处于急加速急刹车急转弯的状态下，因此我们认为急加速急刹车急转弯的次数是重要的参考维度。

5.1.3 驾驶习惯评分模型建立

前文中已经提到的UBI车险定价一般有两种模式，一种是PAYD（Paid As You Drive）表示的是驾驶员使用机动车辆的时间越长频率越高，那么其风险也就越大，相应的保费就越高；反之，如果使用的量越小那么保费就相应的越低，同网络流量的道理相类似，通过这种方式实现保费厘定的公平性的。另外一种则是PHYD（Paid How You Drive），这种模式指的是根据驾驶员如何驾驶机动车辆来判断其风险的大小，也就是根据驾驶员的驾驶习惯来判断他的风险大小，进而判断他的保费高低，对驾驶习惯良好的驾驶员给予相应的保费优惠以激励，同时使得保费因人不同而个性化。驾驶习惯的评分机制或者评分模型是UBI车险定价的关键。为了对驾驶习惯进行合理公平的评分，根据前文的介绍，我们采用了月行驶里程、月行驶时长、单次行程行驶时长、夜间行驶时间、超速行驶以及急加速急刹车急转弯的次数等指标对驾驶员的驾驶习惯的风险进行分级和评分。这些指标以及各指标中包含的各个水平构成了该评分模型的多个维度，因此为了更加合理的得到各指标的权重，本文提出基于广义线性模型（GLM）来构建各指标的权重并进行评分，即通过广义线性模型来量化驾驶员驾驶习惯的风险等级，进而获得每种类型的驾驶习惯的评分，做到费率厘定的差异化和个性化。

5.1.3.1 广义线性模型

[2] 经典线性回归模型往往假设被解释变量Y服从正态分布，被解释变量Y的方差为常数，解释变量X通过线性变换影响因变量Y。而广义线性模型通常假设被解释变量Y服从指数型分布，被解释变量Y的方差随均值而变化，解释变量X通过非线性变换影响被解释变量Y期望值。广义线性模型包含随机成分、系统成分和连接函数等三个部分。各部分描述如下：第一，随机成分，即因变量或者误差项的概率分布。因变量的每个观察值相互独立且服从指数型分布族中的一个分布。指数型分布族包括很多常见的分布，如正态分布、泊松分布、逆高斯分布、二项分布、伽马分布等。可表示为：

$$f\left(y_i|\theta_i,\phi\right)=\exp\left\{\frac{y_i-b\left(\theta_i\right)}{a(\phi)}+c\left(y_i,\phi\right)\right\} \tag{1}$$

式中，i表示第i个观察值，$a(\varphi)$,$b(\theta_i)$,$c(y_i,\phi)$为已知函数。

第二，系统成分，即解释变量的线性组合，可以表示为：$\eta=X_i\beta=X_1\beta_1+\cdots X_p\beta_p$。系统成分与古典线性回归模型没有任何区别。在建立广义线性模型时，通常会包含截距项，此时 $x_1=1$。

第三，连接函数，连接函数单调且可导，它建立了随机成分与系统成分之间的关系可表示为：

$$E\left[Y_i\right]=\mu_i=g^{-1}\left(\eta_i\right) \tag{2}$$

可见，在广义线性模型中，对因变量的预测值并不直接等于解释变量的线性组合，而是该线性组合的一个函数变换。综上所述，广义线性模型的一般形式可表示为：

$$E\left[Y_i\right]=\mu_i=g^{-1}\left(\sum_j X_{ij}\beta_j+\varepsilon_i\right) \tag{3}$$

$$\mathrm{Var}\left(Y_i\right)=\frac{\phi V\left(\mu_i\right)}{\omega_i} \tag{4}$$

其中，$V(\mu_i)$ 表示方差函数，ω 表示先验权重，即观察值对应的一个权重或信度因子。

索赔强度一般是用伽马分布来拟合，伽马分布的密度函数是右偏函数且其尾部相对比较薄，而且其方差等于均值的平方。伽马分布的概率密度函数可以表示为

$$f\left(y_i\right)=\frac{1}{y_i\Gamma(v)}\left(\frac{y_iv}{\mu}\right)^v\exp\left(-\frac{y_iv}{\mu}\right) \tag{5}$$

索赔频率是非负离散性随机变量，泊松分布是其中比较常用的分布，其概率分布函数为：

$$\Pr\left(Y_i=y_i\right)=\frac{\exp\left(-\lambda_i\right)\lambda_i^{y_i}}{y_i!} \tag{6}$$

广义线性模型的参数估计一般采用极大似然估计法，由 y_i 的概率密度函数可得 y_i 的似然函数为

$$l\left(y_i|\theta_i,\phi\right)=\ln f\left(y_i|\theta_i,\phi\right)=\frac{y_i-b\left(\theta_i\right)}{a(\phi)}+c\left(y_i,\phi\right) \tag{7}$$

通常采用迭代的方法来估计广义线性模型的参数，如 Newton-Raphson 方法，该方法是三种极大似然估计近似值算法之一，其他两种方法分别为 Fisher 得分方法和迭代加权最小二乘法。本文采用 Newton-Raphson 方法，该方法的第 r 次迭代结果可表示为:

$$\tilde{\beta}^{(r)}=\left(X^TW^{(r-1)}X\right)^{-1}X^TW^{(r-1)}\left(\eta^{(r-1)}+g'\left(\mu^{(r-1)}\right)\left(Y-\mu^{(r-1)}\right)\right) \tag{8}$$

本文采用 $Pearson\chi^2$ 统计量，该统计量表示实际值 y_i 和模型估计值之间的偏差，其值越小，广义线性模型拟合度越高，该统计量可表示为：

$$P=\sum_{i=1}^{n}r_i^2\sim\chi^2(n-p) \tag{9}$$

其中 r_i 为 Pearson 残差

$$r_i=\frac{y_i-\mu_i}{\sqrt{\mathrm{Var}\left(\mu_i\right)}} \tag{10}$$

我们同时也考虑估计值和残差的散点图来体现模型的拟合情况，如果广义线性模型拟合程度较差，那么相应的估计值和残差的散点图中存在某种趋势，反之则没有明显趋势。

5.1.3.2 模型的选择及因子分析

[3] 评分机制的前后逻辑和主要步骤：

（1）首先将驾驶习惯各指标对应的变量作为广义线性模型的解释变量，并将该驾驶员对应的保单数据，包括承保数据和理赔数据得到的案均赔款和索赔频率作为被解释变量；

（2）然后通过建立 Poisson 模型和 Gamma 模型厘定各个指标变量各个不同水平的相对风险系数；

（3）以每个指标变量风险水平最低的系数为基准，调整其他风险水平的系数；

（4）设定评分的基准为满分 100 分，然后通过基准的分数除以各指标变量系数的乘积，从而得到各种交叉风险水平的评分值，即得到不同类型驾驶习惯的评分值。

对于利用广义线性模型进行风险的厘定，解释变量一般是单均赔款，而单均赔款是索赔频率和案均赔款的乘积，因此对索赔频率和案均赔款用两个子模型分别建模，最后将两模型结果整合起来得到对解释变量的预测模型。分拆建模理由的主要有三个：

1. 不同风险因子对两个指标的影响程度有所不同；

2. 赔案金额的变动性较大，若直接建模，可能会隐藏原本从索赔频率中可以发现的一些规律。

3. 可以更好地确认并剔除某些随机效应的影响。

表 4-1 列出了几个解释变量建模中最常用的模型结构：

响应性变量（索赔倾向、续保率、签约率）：通常采用二项分布、logit 函数作为连结函数，此结构通常也被称为 logistic 模型。

在初步确定模型的基本结构后就要开始选择模型所需的因子。一般来说，GLM 要求模型要包含那些对数据有系统性影响的解释因子，而排除那些没有系统性影响的因子。区别一个因子的影响是系统性的还是随机的（故而在未来很难重复出现），通常可以根据以下几点进行判断：（1）参数估计值的标准差（2）偏差分析（类型 III 检验）（3）时间一致性检验（4）经验判断。

表 **1**　广义线性模型常用模型结构

解释变量	分布类型	连接函数	权重
索赔频率	Poisson 分布	log 连结函数	已赚车年作为权重
案均赔款	Gamma 分布	log 连结函数	赔案次数作为权重
单均赔款	Tweedie 分布	log 连结函数	已赚车年作为权重

GLM 除了可以给出极大拟似然法的参数估计值，还可以提供一些关于参数估计不确定性的信息。其中一个比较重要的信息就是估计的标准差。我们可以根据对数似然函数值的二阶导数矩阵 H（Hessian）来计算出该数值。尽管理论上有关标准差的检验可以在单个参数估计值上进行，但在实践中通常都会如图 2 这样将 GLM 因子拟合的各个参数估计值与其相对应的标准差放在一起来研究

实际上，模型中的每一个因子都可以画出类似于图 2 的折线图。图中展示的是过往出险记录因子的模型结果，它的因子水平分成了 13 组。中间的实线表示 GLM 模型拟合结果。上下两条虚线代表着在参数估计值两侧的两个标准差。蓝色柱状图代表的是过往出险记录因子在各个水平的暴露数分布情况。

可以近似认为，在模型合适准确的前提下，每一个水平的真实值存在于两条细实线之间的可能性约为 95%，即为 95% 的置信区间。当上下两条细实线相距非常远时，说明参数估计的精度不高、不确定性较大。此时就需要对该结果加以分析，究竟是数据量太小引起的，还是由于其他的相关因子的影响，又或者这个经验数据本身就具有很大的浮动性。倘若是因为数据量太小引起的、在某个因子水平的标准差非常大，可以考虑将这个因子水平和其他数据量充足且稳定的因子水平进行合并。倘若是因为经验数据本身就具有很大的浮动性，我们需要通过其他检验，例如类型 III 检验（Type3）和时间一致性检验，来判断是否要在 GLM 模型中使用该因子。尽管图上的标准差只代表了对比基础水平的参数估计值的确定性，这样的图表也对因子的显著性提供了很好的直觉判断。

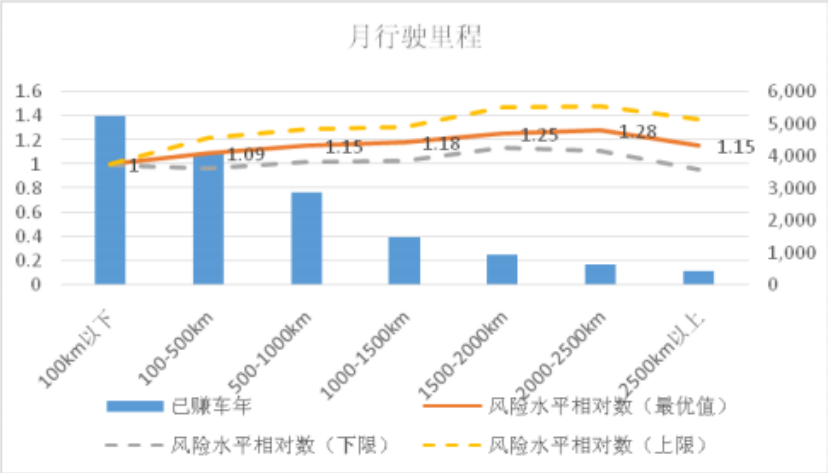

图 **2**　广义线性模型拟合结果和置信区间-月行驶里程

上图中过往出险记录这一因子的参数估计值的阶差还是比较明显的，这很说明图中分析的因子是显著的。

在基于偏差的诊断检验中，我们一般是对两个“嵌套”模型规模残差的差值实施检验（如果规模参数未知，可以经过适当调整后进行 F 检验），并将该过程称之为“类型 III 检验”。其中“嵌套”是指一个模型的变量集包含在另一个模型的变量集之中。如果或 F 检验所得 P 值小于 5%，那么就可以认为该相应因子是显著的。上述类型 III 检验除了可以计算出所需 P 值以外，还可以为参数估计值和标准差的图表演示提供一些有意义信息。例如当一个因子被剔除时，我们就可以通过类型 III 检验来看出其他相关因子为此所做出的调整或补偿。而且不像估计参数标准差，类型 III 检验结果并不会受到因子基础水平选择的影响。偏差分析的优点在于能够非常简单的应用统计数据来判断该因子的显著性。缺点在于，比较机械且缺少经验判断。

一般来说, 除了这些比较典型的统计检验以外, 我们还要考察各个因子的观测结果在时间上的一致性。例如，当我们在处理一年以上的经验数据时，我们就需要考虑特定因子在每个暴露年份的影响力。理论上，我们可以针对各个年份分别进行模型拟合，但由于某一个因子在随时间变化中所产生的影响往往会被另一个相关因子的变动所抵消，从而导致模型的可解释性大大降低。因此，一个更清晰有效的检验是先针对各个因子建立包含因子与时间变量相互作用的模型，然后再对这一系列的模型进行拟合。图 3 就展示了因子与暴露年份相互影响的一个例子。我们可以看出，途中的各条线基本保持平行，说明因子的相对影响程度在各年间保持一致，从而表明了这些因子很可能对未来数据具有较好的预测能力。

当驾驶习惯的指标变量进入模型，在车联网数据和保单数据的基础上，广义线性模型拟合出最终的结果，各项变量的各个水平都有其对应的风险系数或者说是风险水平相对数，而且在 95% 置信区间下还有风险水平相对数的上限和下限，用以表示该风险水平相对数最优估计的精确度。

5.1.4 模型验证

[4] 皮尔逊残差是一种常用的残差统计量，调整后的皮尔逊残差为:

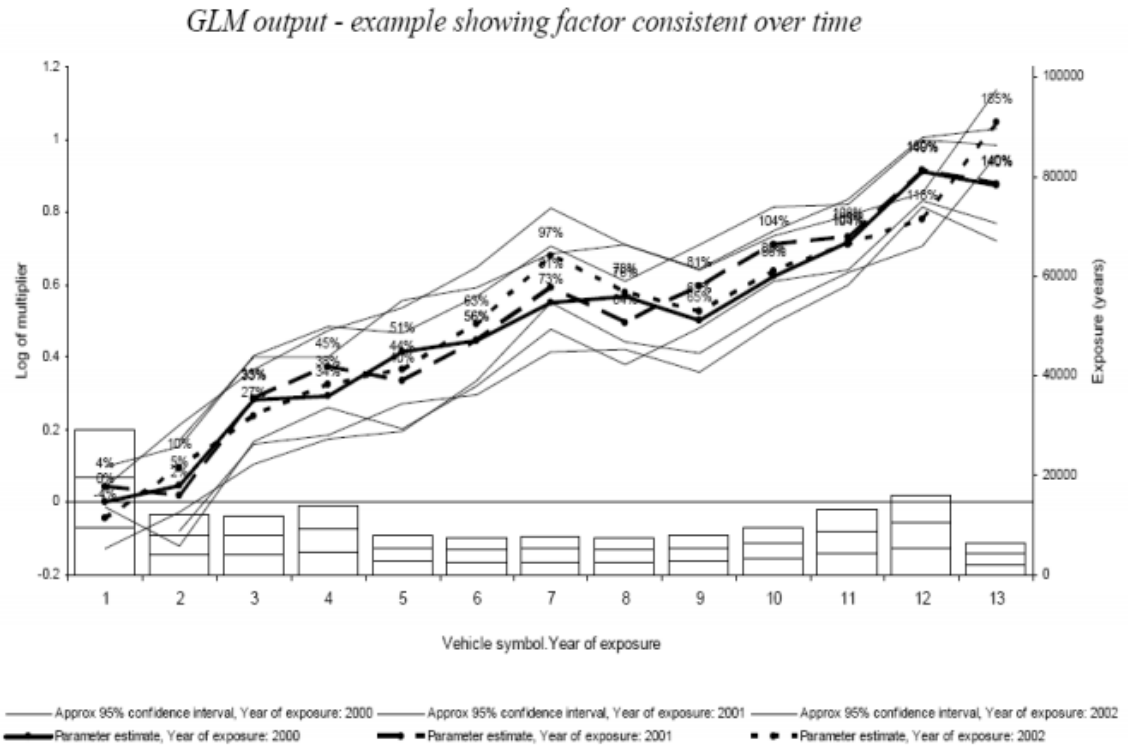

图 **3**　因子未来预测能力

$$r_i = \frac{y_i - \mu_i}{\sqrt{\mathrm{Var}\,(\mu_i)}} \tag{11}$$

经过这种调整并不能改变分布的形状，但却可以让均值不同的观测点变得具有可比性。通过观察拟合值—残差的散点图，我们可以对误差函数假设的合理性进行一定的评判。例如，如果假设成立，那么不管拟合值是多少，它的标准偏离残差值都应该服从标准正态分布。下图展示的是用一个带有伽玛方差函数的 GLM 对一组数据进行拟合的结果，该数据是基于伽玛分布随机产生的数据集。从图 4 中可以看出，从左到右，散点图的均值和方差基本保持一致，说明我们对该方差函数的假设是成立的。

在得到各个指标变量的风险相对系数后，可以利用下面的计算公式计算各种交叉风险水平下的评分，即不同类型驾驶习惯的评分值：

$$S = \frac{S_0}{\prod i} \tag{12}$$

其中 S 表示某一类型驾驶习惯的最终评分，S_0 表示基准评分值，一般为满分 100，i 表示各指标变量所对应的某一风险水平的相对系数值。这样我们可以以风险水平的相对系数的乘积作为权重获得不同驾驶习惯的评分。

在得到驾驶习惯的评分以后，因为每位驾驶员都有其对应的得分，那么也就是说保单数据中每一个保单或每一辆车都对应一个评分值，所以再次结合广义线性模型，将驾驶习惯评分作为车险定价的一个解释变量进入模型，和其他从人因素、从车因素和环境因素的指标变量一起作为解释变量进入广义线性模型进行车险的风险分级，并且在监管的范围内，得到承保折扣，进而利用保费计算公式计算出保费。

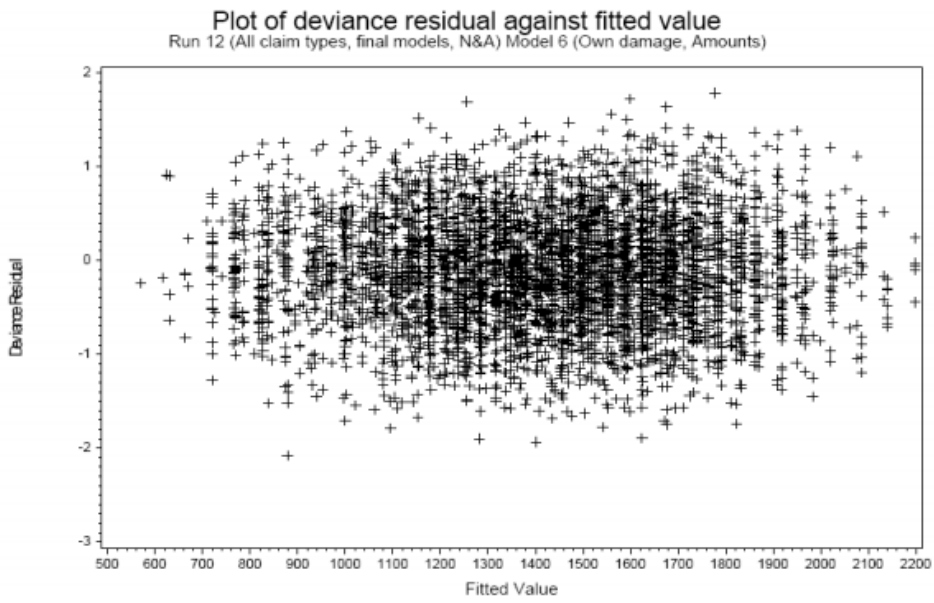

图 **4**　拟合值-残差散点图

5.1.5 软件测试

通过 APP 对我们之前提出的各方面驾驶习惯信息进行具体测试：

（1）行车数据模块测试

行车数据功能模块的实现运行效果如图 5 所示，从测试结果可以看出，用户能够比较清楚的浏览每日和每周出行的里程、行驶时长、平均车速，也可以了解夜间驾驶时长、夜间车速和连续驾驶时长等信息，将页面下拉后可以查看车辆超速和四急驾驶行为的出现次数，由此可对自己的日常驾驶情况形成一个直观的认识，也可以作为对驾驶行为得分的原始数据解读。

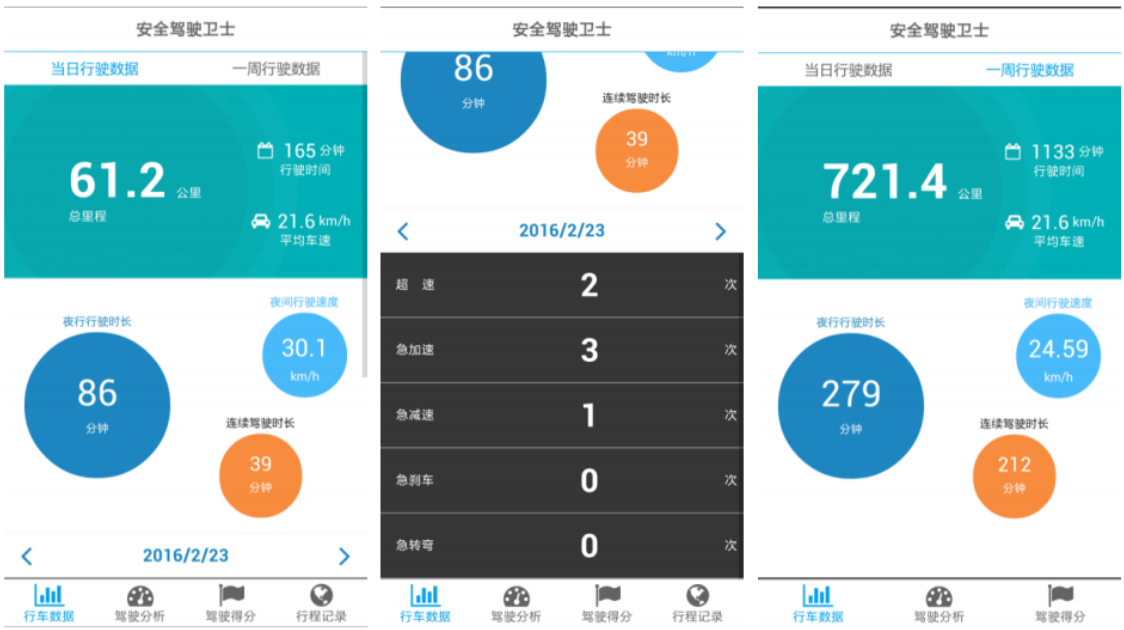

图 **5**　行车数据功能模块运行效果图

（2）驾驶分析模块测试

1、打开客户端，点击跳转到驾驶分析界面；2、查看一周四急数据统计折线图，点击图例；3、点击里程选项，查看本周行驶里程统计柱图；4、点击时长选项，查看本周行驶时长统计柱图；5、点击车速选项，查看本周平均车速统计柱图；6、点击昼夜比选项，查看当日和本周的行驶昼夜时长比饼图；7、点击评分选项，查看本周驾驶综合得分趋势。

图 **6**　车辆操作效果图

图 **7**　车辆行驶效果图

在四急数据统计图表中，用户可以看到本周内每天的四急行为次数，从横向上了解四急次数的一周变化，从纵向上比较四种不良驾驶行为出现的多少和比例。通过点击图例可关闭折线的显示，从而单独查看其中的一条或多条折线，此时折线纵轴的显示比例也会自动调整为合适的区间。驾驶得分的折线图主要展示了一周驾驶安全综合评分的发展变化趋势，增强用户对近期自身驾驶行为和驾驶安全性的认识，结合折线走势逐步改进自己的驾驶习惯。通过直方图上的数据展示，用户可以快速了解自己本周驾驶出行情况，按照日期查看一周内每天的行驶里程、行驶时长和行车速度信息，柱状图的呈现方式提供了更直观的纵向比较，能够表现某日的突出数据，也能反映出用户的整体驾驶出行规律。

行驶昼夜比反映了车主夜间和白天驾驶出行占总时长的比例情况，红色表示白天出行，深蓝表示夜间出行，用户通过饼图可以很方便的查看两部分对应的行驶时长和所占比例，点击扇形后这部分内容会突出显示，点击图例可选择任意一部分填满饼图单独进行查看。行驶昼夜比的数据展示可为车主调整驾驶出行时间提供参考。

（3）驾驶得分模块测试

驾驶行为得分功能模块的实际运行效果如图 5 所示，可以看出，雷达图使用不同维度展示了五个单项驾驶行为得分，综合得分显示在中心位置上，五个得分节点组成的区域大小可以一定程度上体现出用户的驾驶行为综合表现情况。根据用户的综合得分情况，对得分区间进行判断，可将安全等级划分为安全、较安全、提高警惕、危险驾驶和非常危险，在雷达图上方用不同颜色显示，点击后可以查看具体的驾驶得分说明，其页面运行效果如图 8 所示。得分说明页面对综合得分和个单项得分的评测内容进行了说明，同时解释了不同安全级别对应的驾驶得分区间，帮助用户更好地理解自己的每日驾驶得分和驾驶安全等级

（4）行程记录模块测试

行程记录模块的地图轨迹展示效果如图 9 所示，图中显示的是车主于 2016 年 2 月 21 日全天的行驶轨迹，地图上标明了驾驶起点和终点位置。图 6 是开启四急点标记显示后的轨迹展示效果，点击“显示四急”按钮后，系统将在地图上标记出一天中急加速、

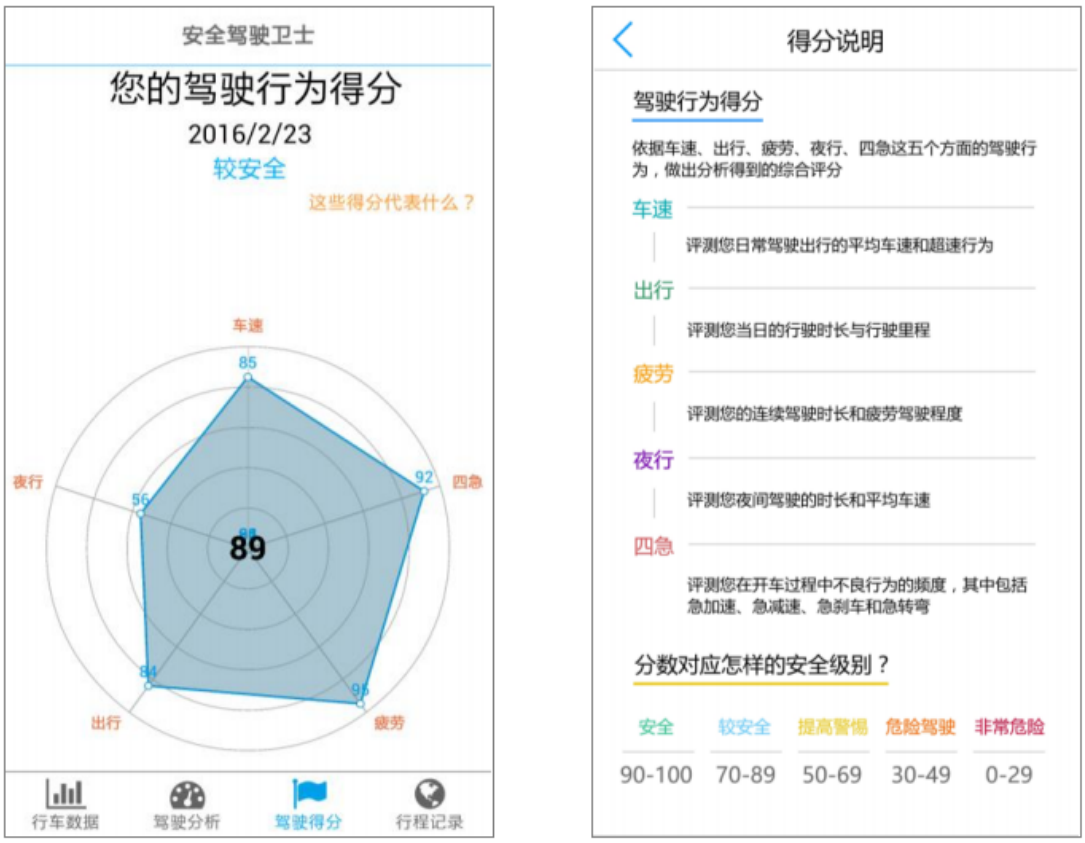

图 **8**　驾驶得分功能运行效果图

急减速、急刹车和急转弯行为出现的位置。底部显示的是四急驾驶行为对应的图标及出现次数。

5.2　问题二模型的建立与求解

5.2.1 调查问卷的设计

基于第一问的模型建立以及接下来要利用这个问卷的数据来提高续保概率的可行性来看，我们将调查问卷设计如下图：

5.2.2 数据的来源与处理

5.2.2.1 数据来源

基于以上提出的调查问卷，我们提取基础数据，包括车险的承保数据和理赔数据以及保单车辆所对应的车联网数据均是来自某财险公司的真实数据，其中车联网数据是作者参与的该公司的车联网项目所收集的数据。其中承保数据和理赔数据是 2016 年 1 月 1 日至 12 月 31 日起保和报案的数据，车联网数据为对应车辆的月度数据。该数据的统计指标前面已经做过陈述，此处不再赘述。而对车险的承保数据和理赔数据的主要维度稍作陈述。

从车信息：车架号、发动机号、新车购置价、厂牌车型、座位数、吨位数、使用性质、车辆类型等；

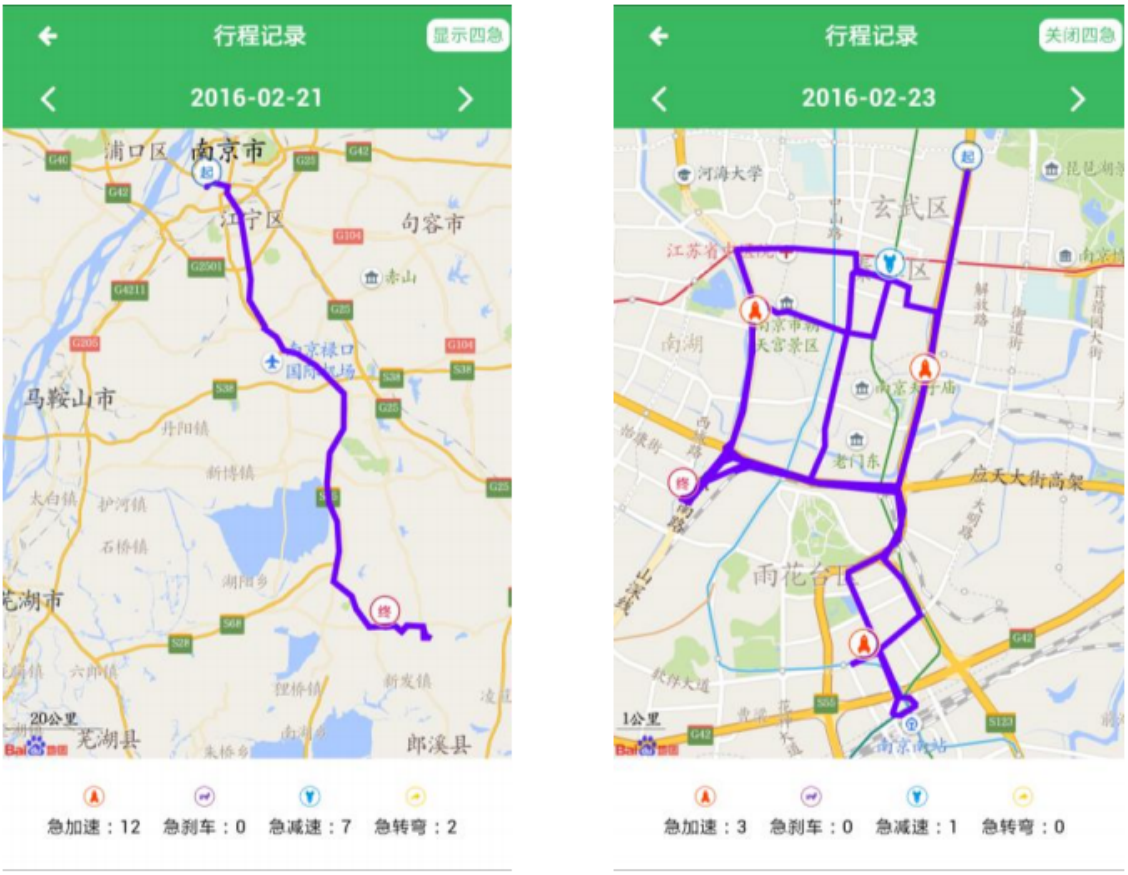

图 **9**　行驶轨迹地图界面

从人信息：性别、年龄等；地区信息：二级机构、三级机构等；险别；保单信息：保单号、保单起期、保单止期、基准保费、签单保费等；承保渠道；（中保信）平台信息：无赔款优待记录、上年出险次数、上年出险金额等；

赔款数据包含的信息包括：报案号、赔案号、事故发生时间、报案时间、立案时间、结案时间、理赔类型、事故原因、损失性质、已决赔款、未决赔款、直接理赔费用等

5.2.2.2 数据处理

根据提取的基础数据，对定价数据和准备金评估数据进行了核对，包括保费收入、已决赔款和未决赔款，由于业务数据的机密性，因此不再具体展示，但从数据校验的情况来看，公司商业车险精算定价数据与公司的准备金评估数据之间的保费收入差异非常小，数据可靠性很好。其他需要考虑的因素还有：建模过程中首先要考虑数据量，理想的情况是每个分组的赔案件数都大于 1082 件，以使得分析测算得到的结果具备一定的稳定性。其次，实际建模过程中需要根据公司的实际承保数据和风险细分类别进行调整，在数据量较小的情况下，需要对不同的分组合并达到模型的收敛，合并时要根据单因子的结果尽可能满足同质性。对该财险公司 2016 年的理赔数据进行初步分析，通过 Excel 软件作图，得到的结果如图 12 和 13。从图中可以看出，不论是车损险还是第三者责任险都有厚重的右尾，整体来看赔案金额分布很符合伽马分布的特点:

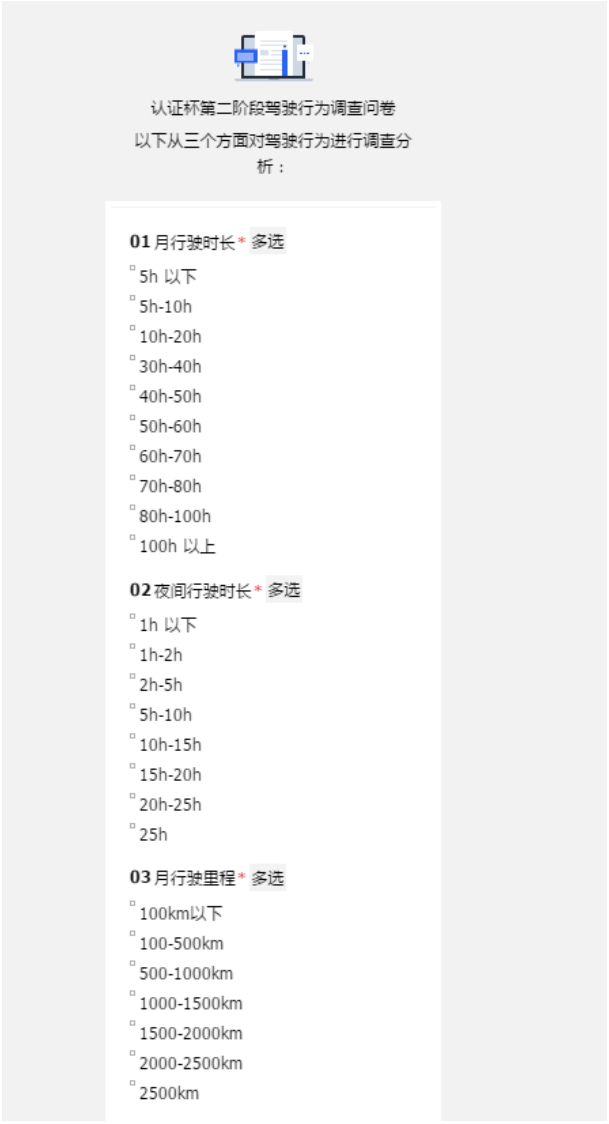

图 **10**　调查问卷部分（**1**）

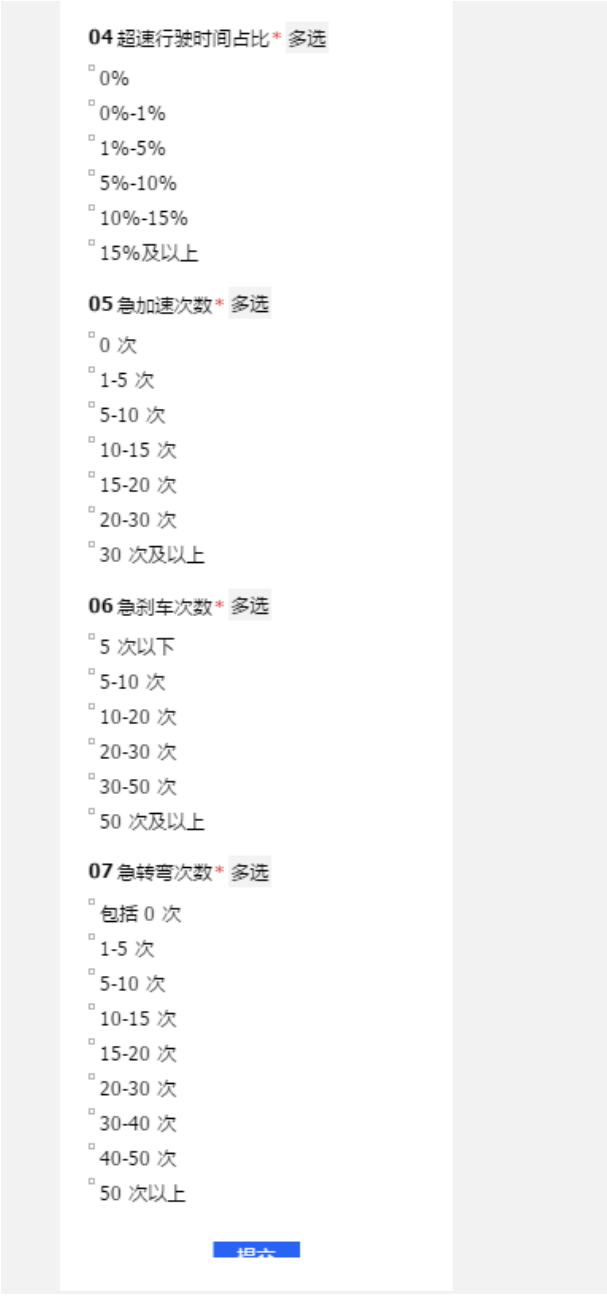

图 **11**　调查问卷部分（**2**）

5.2.3 驾驶习惯评分模型的运行与检验

5.2.3.1 选定分类变量

根据前文的介绍，本章的实证分析选定以下指标变量：

（1）月行驶时长，包括 5h 以下、5h-10h、10h-20h、···、80h-100h、100h 以上等 10 个变量水平；

（2）夜间行驶时长，包括 1h 以下、1h-2h、···、20h-25h、25h 及以上等 10 个变量水平；

（3）超速行驶时间占比，包括 0%、0%-1%、···、10%-15%、15% 及以上等 8 个变量水平；

（4）急加速次数，包括 0 次、1-5 次、···、20-30 次、30 次及以上等 6 个变量水平；

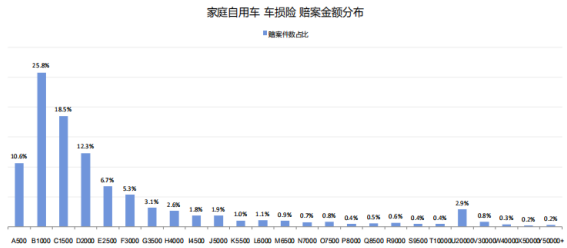

图 **12**　家用车车损险赔案金额分布

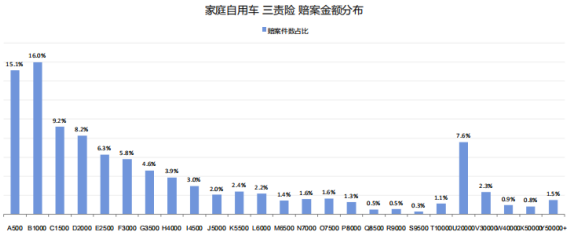

图 **13**　家用车第三者责任险赔案金额分布

（5）急刹车次数，包括 5 次以下、5-10 次、···、30-50 次、50 次及以上等 6 个变量水平；

（6）急转弯次数，包括 0 次、1-5 次、···、40-50 次、50 次以上等 8 个变量水平。

（7）月行驶里程，包括 100km 以下、100-500km、500-1000km、1000-1500km、1500-2000km、2000-2500km、2500km 以上 7 个变量水平；

因此，按照这种划分方法，风险类别的组合共有 10*10*8*6*6*8*7=1612800 种

5.2.3.2 广义线性模型的建立

[5] 建立广义线性模型：

$$\begin{cases} E(Y_{ijklmnu}) = u_{ijklmnu} \\ \varphi_{ijklmnu} = \mu_0 + \alpha_i + \beta_j + \gamma_k + \lambda_m + v_n + \omega_u \\ \varphi_{ijklmnu} = ln(u_{ijklmnu}) \\ Var(Y_{ijklmnu}) = \frac{\phi}{m_{ijklmnu}} V(\mu_{ijklmnu}) \\ (i = 1,2,...10; j = 1,2,...10; k = 1,2,...8; l = 1,2,...6; \\ m = 1,2,...10; n = 1,2,...8; u = 1,2,...7;) \end{cases} \tag{13}$$

其中 i，j，k，l，m，n，u 分别表示上面的 7 个指标变量，i=1,2,···,9,10，j=1,2,···,9,10，k=1,2,···,7,8，l=1,2,···,5,6，m=1,2,···,5,6，n=1,2,···,7,8，u=1,2,···,6,7。

该模型的矩阵可以表示为：

$$\begin{aligned} Y &= (Y_1, Y_2, \ldots, Y_{1612800})^T \\ \mu &= (\mu_1, \mu_2, \ldots, \mu_{1612800})^T \\ \varphi &= (\varphi_1, \varphi_2, \ldots, \varphi_{1612800})^T \\ m &= (m_1, m_2, \ldots, m_{1612800})^T \\ \beta &= (\mu_0, \alpha_1, \ldots, \alpha_{10}; \beta_1, \ldots, \beta_{10}; \gamma_1, \ldots, \gamma_8; \delta_1, \ldots, \delta_6; \lambda_1, \ldots, v_8; \omega_1, \ldots, \omega_7) \end{aligned} \tag{14}$$

所以，矩阵 X 表示为：$X^T = (L, A, B, C, D, E, F, G)$ 其中，$A = \begin{bmatrix} A_1 \\ A_2 \\ \cdots \\ A_{10} \end{bmatrix}, B = \begin{bmatrix} B_1 \\ B_2 \\ \cdots \\ B_{10} \end{bmatrix}, C = \begin{bmatrix} C_1 \\ C_2 \\ \cdots \\ C_{10} \end{bmatrix}, D = \begin{bmatrix} D_1 \\ D_2 \\ \cdots \\ D_{10} \end{bmatrix}, E = \begin{bmatrix} E_1 \\ E_2 \\ \cdots \\ E_{10} \end{bmatrix}, F = \begin{bmatrix} F_1 \\ F_2 \\ \cdots \\ F_{10} \end{bmatrix}, G = \begin{bmatrix} G_1 \\ G_2 \\ \cdots \\ G_{10} \end{bmatrix}$。

5.2.3.4 结果分析及模型验证

将投保客户（驾驶员）的驾驶习惯数据，即通过车联网技术收集并通过大数据技术处理后的车联网数据和保单承保理赔数据进行整合。整合的数据导入到广义线性模型中，由模型得到的直接结果是各个指标变量各个水平的相对风险系数（估计值 β 指数化的值），即通过广义线性模型量化了驾驶员驾驶习惯的风险相对大小，然后对风险相对系数做平滑处理，进而在此基础上计算得到各个驾驶员，即投保客户驾驶习惯的评分结果。需要说明的是处理驾驶习惯数据的软件是 SAS 软件，SAS 软件是由全球最大的软件公司之一的 SAS 软件公司开发的一款强大的数据存储和数据分析的软件。本文利用 SAS 软件对进入模型前的数据进行整理，然后利用 SAS 中的广义线性模型模块进行建模，并将整理后的数据导入模型中进行风险分级。结果如表 2：

表 **2**　月行驶里程建模结果

月行驶里程	风险水平相对数	风险水平相对数 (下限)	风险水平相对数 (上限)	选定系数
100km 以下	1.00	1.00	1.00	1.00
100-500km	1.09	0.97	1.22	1.10
500-1000km	1.15	1.02	1.29	1.15
1000-1500km	1.18	1.03	1.31	1.20
1500-2000km	1.25	1.14	1.47	1.25
2000-2500km	1.28	1.11	1.48	1.28
2500km 以上	1.15	0.96	1.37	1.30

从上表可以看到下面的个别水平由于赔案件数较少，风险水平相对数的上下限区间也较大，因此该最优估计值存在一定的不确定性。对于这种情况我们通常有两种方法来处理，一个是利用线性趋势或者指数线性趋势来拟合最优估计值的趋势，将偏离趋势的估计值进行相应的调整，另外一个就是根据经验将明显偏离规律的点调整回来。因此对于上表的指标变量的结果我们可以根据经验以及其规律将 2500km 以上的水平所对应的最优估计值上调到 1.30 即可。当然对于基准不是相对风险最小的水平的指标变量还应该进行基准的调整，然后再进行系数的平滑处理。

表 **3**　月行驶时间建模结果

月行驶时长	风险水平相对数	风险水平相对数 (下限)	风险水平相对数 (上限)	选定系数
5h 以下	0.42	0.35	0.50	1.00
5h-10h	0.58	0.50	0.67	1.03
10h-20h	0.58	0.45	0.76	1.05
20h-30h	0.71	0.65	0.78	1.15
30h-40h	0.87	0.79	0.97	1.20
40h-50h	0.92	0.64	1.30	1.35
50h-60h	1.00	1.00	1.00	1.45
60h-80h	1.14	0.99	1.32	1.70
80h-100h	1.17	1.04	1.32	1.95
100h 及以上	1.01	0.86	1.18	2.10

对于这种基准不在风险相对最低的水平上的应该调整其基准，各水平的最优值同除以 0.42，则调整后的风险水平相对数为 1.00、1.38、1.38、1.69、2.07、2.19、2.38、2.71、2.78 和 2.40，然后经过平滑处理我们得到最终的风险水平相对数，即表中的选定系数。

表 **4**　夜间行驶时长建模结果

夜间行驶时长	风险水平相对数	风险水平相对数 (下限)	风险水平相对数 (上限)	选定系数
1h 以下	0.40	0.05	2.99	1.00
1h-2h	0.68	0.57	0.83	1.05
2h-4h	0.83	0.73	0.94	1.10
4h-6h	0.99	0.88	1.11	1.25
6h-8h	1.03	0.93	1.15	1.30
8h-10h	1.00	1.00	1.00	1.40
10h-15h	1.07	0.96	1.19	1.50
15h-20h	1.17	1.04	1.32	1.80
20h-25h	1.27	1.12	1.44	2.00
25h 及以上	1.26	1.09	1.45	2.30

表 4 展示了夜间行驶时长的建模结果。风险水平的相对数与夜间行驶时长成正相关，符合规律。2h 到 25h 的分组有相对较多的数据量，结果相对稳定，但受到数据量的限制，其他分组的风险水平相对数的置信区间都很宽，模型结果的可靠性较差。

表 5 展示了超速行驶时间占比情况的建模结果。可以看出随着超速行驶时间的占比越高，其相对风险也越大，这是合乎常理的。

表 **5**　超速行驶时间占比建模结果

超速行驶时间	风险水平相对数	风险水平相对数 (下限)	风险水平相对数 (上限)	选定系数
0%	0.73	0.65	0.81	1.00
0%-1%	0.76	0.69	0.85	1.04
1%-2%	0.78	0.70	0.87	1.06
2%-4%	0.81	0.73	0.89	1.10
4%-6%	1.00	1.00	1.00	1.35
6%-10%	1.28	1.02	1.59	1.75
10%-15%	1.39	1.12	1.48	1.90
15% 及以上	1.51	1.28	1.79	2.06

表 6、表 7 和表 8 中分别展示了急加速急刹车急转弯建模的结果，大体上符合常理的规律，另外从中可以看出部分分组下的赔案件数较少，结果存在一定的不确定性；另外，选定系数是在最优值进行 offbalance 平滑之后的系数，再进行基准（base）的调整后选定的系数，其中有一定的主观因素，这些都是其局限性。

在确定最终的模型之后我们对建模中使用的因子的显著性和模型的拟合结果都进行了检验。因子的显著性分析在确定模型之后对出险频率和案均赔款模型中的所有因子的显著性进行排序，并结合经验加以判断，判断是否为对模型有系统性影响的因子。对模型的检验主要是通过 type1 和 type3 检验。通过 type1 检验附加到包括先前所有效应

表 **6**　急加速次数建模结果

急加速次数	风险水平相对数	风险水平相对数 (下限)	风险水平相对数 (上限)	选定系数
0 次	0.53	0.44	0.65	1.00
1 次-5 次	0.64	0.48	0.85	1.05
5 次-10 次	1.00	1.00	1.00	1.15
10 次-20 次	1.21	1.05	1.48	1.25
20 次-30 次	1.33	1.16	1.58	1.65
30 次及以上	1.56	1.25	1.71	1.95

表 **7**　急刹车次数建模结果

急刹车次数	风险水平相对数	风险水平相对数 (下限)	风险水平相对数 (上限)	选定系数
5 次以下	0.92	0.46	1.86	1.00
5 次-10 次	0.66	0.43	1.03	1.10
10 次-20 次	0.73	0.56	0.95	1.20
20 次-30 次	0.83	0.71	0.97	1.38
30 次-50 次	1.00	1.00	1.00	1.65
50 次及以上	1.44	1.23	1.94	2.40

的模型中的效应是否显著，通过 type3 检验从包含所有变量的模型中剔除某个变量对模型的拟合结果是否有显著影响。

下面的表 9 和表 10 展示了模型结果的 type1 和 type3 检验的情况：

通过 type1 和 type3 检验可以看出，和 P 值检验所得的 P 值都小于 5%，因此可以认为驾驶习惯的各指标变量都具有统计显著性。

5.2.4 车险费率计算

5.2.4.1 驾驶习惯得分计算

[6] 通过一个实例来分析驾驶习惯得分的计算过程和结果，根据车联网数据的统计结果，该财险公司某客户的驾驶习惯信息如下：月行驶里程 778km，月行驶时长 16.4h，

表 **8**　急转弯次数建模结果

急转弯次数	风险水平相对数	风险水平相对数 (下限)	风险水平相对数 (上限)	选定系数
0 次	1.89	0.74	4.81	1.00
1 次-5 次	0.70	0.51	0.95	1.10
5 次-10 次	0.93	0.54	1.62	1.40
10 次-20 次	1.00	1.00	1.00	1.53
20 次-30 次	1.01	0.79	1.31	1.55
30 次-40 次	1.30	1.03	1.62	2.00
40 次-50 次	1.38	0.98	1.95	2.12
50 次以上	1.05	0.70	1.59	230

表 **9**　出险频率模型 **type1** 检验

Source	Deviance	NumDF	DenDF	FValue	ProbF	ChiSq	ProbChiSq
Intercept	76,058	-	-	-	-	-	-
月行驶里程	75,838	6	152,990	229.27	<0.0001	458.53	<0.0001
月行驶时长	75,462	9	152,990	32.60	<0.0001	782.46	<0.0001
夜间行驶时长	75,211	9	152,990	74.72	<0.0001	523.01	<0.0001
超速行驶时间占比	75,156	7	152,990	57.21	<0.0001	114.43	<0.0001
急加速次数	74,131	5	152,990	177.75	<0.0001	2132.97	<0.0001
急刹车次数	74,027	5	152,990	8.37	<0.0001	217.58	<0.0001
急转弯次数	73,723	7	152,990	70.28	<0.0001	632.50	<0.0001

表 **10**　出险频率模型 **type3** 检验

Source	NumDF	DenDF	FValue	ProbF	ChiSq	ProbChiSq	Method
月行驶里程	6	152,990	32.31	<0.0001	64.62	<0.0001	LR
月行驶时长	9	152,990	27.25	<0.0001	653.99	<0.0001	LR
夜间行驶时长	9	152,990	69.59	<0.0001	487.16	<0.0001	LR
超速行驶时间占比	7	152,990	30.80	<0.0001	61.60	<0.0001	LR
急加速次数	5	152,990	178.05	<0.0001	1958.59	<0.0001	LR
急刹车次数	5	152,990	6.78	<0.0001	176.32	<0.0001	LR
急转弯次数	7	152990	7064	<0.0001	63572	<0.0001	LR

夜间行驶时长 3.2h，超速行驶时间占比 0.37%，急加速次数 5 次，急刹车 4 次，急转弯 2 次。那么其分别对应的系数为 1.15、1.05、1.10、1.04、1.05、1.00、1.05，所以，计算得到该用户的驾驶习惯得分为：100/（1.15*1.05*1.10*1.04*1.05*1.00*1.05）=68.9，即该用户的驾驶习惯得分为 68.9 分。

5.2.4.2 车险保费计算

通过上面的方法得到驾驶习惯的评分之后，将评分重新作为定价的解释变量进入广义线性模型进行风险分级，得到不同评分区间的相对风险系数，和其他解释变量：年

表 **11**　案均赔款模型 **type1** 检验

Source	Deviance	NumDF	DenDF	FValue	ProbF	ChiSq	ProbChiSq
Intercept	22,786	-	-	-	-	-	-
月行驶里程	22,718	6	15,673	28.56	<0.0001	57.12	<0.0001
月行驶时长	22,542	9	15,673	6.17	<0.0001	148.19	<0.0001
夜间行驶时长	22,405	9	15,673	16.40	<0.0001	114.77	<0.0001
超速行驶时间占比	22,394	7	15,673	4.56	0.0104	9.13	0.0104
急加速次数	22,021	5	15,673	26.18	<0.0001	314.19	<0.0001
急刹车次数	18,802	5	15,673	104.09	<0.0001	2706.27	<0.0001
急转弯次数	18,688	7	15,673	10.63	<0.0001	95.64	<0.0001

表 **12**　案均赔款模型 **type3** 检验

Source	NumDF	DenDF	FValue	ProbF	ChiSq	ProbChiSq	Method
月行驶里程	6	15,673	13.61	<0.0001	27.23	<0.0001	LR
月行驶时长	9	15,673	4.90	<0.0001	117.54	<0.0001	LR
夜间行驶时长	9	15,673	1.80	0.0831	12.58	0.083	LR
超速行驶时间占比	7	15,673	2.17	0.1139	4.35	0.1139	LR
急加速次数	5	15,673	3.32	0.0001	36.53	0.0001	LR
急刹车次数	5	15,673	101.56	<0.0001	2640.58	<0.0001	LR
急转弯次数	7	15673	1100	<0.0001	9896	<0.0001	LR

龄性别（10 个水平）、行驶范围（7 个水平）、无赔款优待（7 个水平）、车型（9 个水平）等的相对风险系数相乘后的结果再进行监管范围（0.7225-1.3225 的系数范围）下的调整，就得到用以计算保费公式中的折扣值。根据上面的四个传统定价因子以及驾驶习惯评分因子（6 个水平）相结合得到 26460（10*7*7*9*6=26460）个风险单元。设随机变量 $Y_{abcde}^{`}$ 表示单元（a,d,c,d,e）的平均索赔额，权重 m_{abcde} 为索赔次数。这里的 a，b，c，d，e 表示上述的五个变量的各个水平，以此建立广义线性模型:

$$\begin{cases} E\left(Y'_{abcde}\right)=\mu'_{abcde} \\ \varphi'_{abcde}=\mu'_0+\alpha'_a+\beta'_b+\gamma'_c+\delta'_d+\lambda'_e \\ \varphi'_{abcde}=\ln\left(\mu'_{abcde}\right) \\ \operatorname{Var}\left(Y_{abcde}\right)=\frac{\phi}{m_{abcde}}V\left(\mu_{abcde}\right) \\ a=1,2,\ldots,10; b=1,2,\ldots,7; c=1,2,\ldots,7; \\ d=1,2,\ldots,9; e=1,2,\ldots,6) \end{cases} \tag{15}$$

模型的矩阵表现形式如下：

$$\begin{aligned} Y' &= \left(Y'_1, Y'_2, \ldots, Y'_{26460}\right)^T \\ \mu' &= \left(\mu'_1, \mu'_2, \ldots, \mu'_{26460}\right)^T \\ \varphi' &= \left(\varphi'_1, \varphi'_2, \ldots, \varphi'_{26460}\right)^T \\ m' &= \left(m'_1, m'_2, \ldots, m'_{26460}\right)^T \\ \beta' &= \left(\mu'_0, \alpha'_1, \ldots, \alpha'_{10}; \beta'_1, \ldots, \beta'_7; \gamma'_1, \ldots, \gamma'_7; \delta'_1, \ldots, \delta'_9; \lambda'_1, \ldots, \lambda'_6\right) \end{aligned} \tag{16}$$

所以，矩阵 X 表示为：$X^T=(L,A,B,C,D,E,F,G)$ 其中，$A=\begin{bmatrix}A_1\\A_2\\\cdots\\A_{10}\end{bmatrix}$，$B=\begin{bmatrix}B_1\\B_2\\\cdots\\B_{10}\end{bmatrix}$，$C=\begin{bmatrix}C_1\\C_2\\\cdots\\C_{10}\end{bmatrix}$，$D=\begin{bmatrix}D_1\\D_2\\\cdots\\D_{10}\end{bmatrix}$，$E=\begin{bmatrix}E_1\\E_2\\\cdots\\E_{10}\end{bmatrix}$，$F=\begin{bmatrix}F_1\\F_2\\\cdots\\F_{10}\end{bmatrix}$，$G=\begin{bmatrix}G_1\\G_2\\\cdots\\G_{10}\end{bmatrix}$。

同理驾驶习惯评分模型，估计得到参数的值，然后指数化并调整得到各个变量的风险相对系数。篇幅所限这里只列出了驾驶习惯评分的结果（经过调整）如表 13 所示：

表 **13**　驾驶习惯风险水平相对系数

驾驶习惯得分（S）	风险水平相对系数
0-40	1.20
40-60	1.15
60-70	1.10
70-80	1.00
80-90	0.90
90+	0.85

对模型的检验主要是通过 type1 和 type3 检验，由于模型结果及模型的检验结果很多，篇幅所限此处就不再一一列举。下面列举的分别是出现频率及案均赔款的 type1 检验结果。

表 **14**　出险频率模型 **type1** 检验

Source	Deviance	NumDF	DenDF	FValue	ProbF	ChiSq	ProbChiSq
Intercept	45614.4	-	-	-	-	-	-
sex_age	45013.53	9	128777	38.62259	0	926.9422	2.8E-180
area	44879.26	6	128777	27.87849	0	390.2988	1.4E-74
ncd	44502.67	6	128777	103.9732	0	1039.732	5.2E-217
models	44431.77	8	128777	17.17438	0	206.0925	1.8E-37
score	44301.68	5	128777	372.3847	0	372.3847	5.66E-83

表 **15**　案均赔款模型 **type1** 检验

Source	Deviance	NumDF	DenDF	FValue	ProbF	ChiSq	ProbChiSq
Intercept	19586.76	-	-	-	-	-	-
sex_age	18953.3	9	8174	7.410586	0	177.8541	1.88E-25
area	18781.92	6	8174	5.520306	1E-10	77.28428	8.99E-11
ncd	18618.63	6	8174	6.565916	3E-10	65.65916	3.03E-10
models	18393.87	8	8174	8.446464	4E-16	101.3576	3.02E-16
score	18376.77	5	8174	7.712554	0.005	7.712554	0.003484

通过模型检验结果可见模型拟合结果较好，且 P 值均小于 5%，故各变量都具有较好的统计显著性。

因此在模型结果的基础上，假设其他因子的风险水平相对系数都为基准，即为 1.00，而驾驶习惯的评分为 85 分，即其风险水平相对系数为 0.90，沿用之前的例子，那么根据保费计算公式得；保费 = 基准纯风险保费/（1-附加费用率）* 折扣系数 =666.9/（1-30%）*0.90=857.4 元，即可以理解为通过 UBI 车险定价将驾驶员的驾驶习惯纳入保费计算中，此类驾驶习惯良好的驾驶员在保费上得到了 10% 的优惠，反之对驾驶习惯较差，评分较低的驾驶员会相应的抬高保费，这样有利于促进整个车险市的业务品质的改善。同时也使得车险费率更加差异化和个性化，对于车险费率市场化改革具有重要意义，即提高续保概率。

5.3 车险业在大数据环境下应如何发展的建议

尊敬的 CEO：

您好：

基于我们对车险公司的续保率及车险率的分析，以下为我们对车险业在大数据环境下应如何发展的建议：

保险经营是建立在大量数据基础上的，这样的经营特点赋予了保险行业天然的数据优势。而占居财产保险业务主要地位的车险自始已经积累了大量的内外部数据，充分利用车险大数据对于提高车险经营效率，创新其服务模式具有重要作用，具体表现在：

（一）改善经营模式，提高产品销售效率

保险业在根本上是服务业，其服务宗旨应以客户为中心。对比大数据营销，当前普遍采用的车险营销模式主要以产品为导向，依靠电话、网络、4s 店、保险代理人、广告等渠道去宣传车险产品，以此拉动保费增长，这种营销模式并没有考虑客户是否真正需求车险产品，只是盲目性的“广撒网”，虽然能争取一些有需求的客户，但效果很一般。长此以往不仅会产生很高的经营成本，而且容易引起客户反感，降低客户对保险公司的信任度，给保险公司的声誉造成恶劣影响。随着大数据技术的快速发展，通过对大量相关数据的分析去锁定目标客户，进行针对性的营销将变得可能。例如对用户浏览的汽车品牌、价格、网站、点击量甚至点击时间等信息进行数据统计，区分出不同的客户群体，根据群体不同意向需求推送相关车险产品，实施精准营销，不仅能节省大量成本，还能提高销售效率。

（二）创新产品开发，提升承保定价能力

保险产品开发是保险经营活动的重要内容，对于增强保险公司竞争力、增加保险公司收益、满足保险消费者的需求有着重要意义。产品开发需要基于风险标的相关数据来预测损失，只有在大量数据的基础上才能对风险事故进行科学的评定，进而制定产品费率。目前，我国的车险费率还是统一制定的，按照车辆用途、使用年限、座位数、新车购置价等确定基础保费。这种费率厘定模式虽然简单，但风险划分不够细致，往往是车险费率与风险并不匹配。比如同样的车，由于驾驶员驾驶习惯不同，风险差异很大。近年我国交通意外事故发生率逐渐攀升，给车险经营带来了很大困难，主要原因在于保险公司无法掌握驾驶人的实时驾驶习惯，实行差异化定价。如今通过收集车辆驾驶人的驾驶时间、车速、急刹车频率等信息，利用数据处理技术，对驾驶人驾驶情况做出“个性化”评定，进而精算定价，设计出符合驾驶人驾驶习惯的保险产品。如人保财产对投保其网上车险的用户，从人、车、地域等多个角度综合评定，利用大数据分析技术对其车险费率进行适当调整，建立了浮动费率体系。

（三）改善保险理赔服务质量

传统车险经营中，当投保车辆发生保险事故时，保险公司总是被动的接收出险通知，然后登记立案、现场勘查、审核保险责任、确定损失赔偿限额，最后赔付保险金。这一系列理赔手续不仅费时耗力，而且在很多情况下无法及时处理，造成车险理赔低效率，引起保险消费者的不满。在大数据时代，保险公司可以利用数据平台及时获取客户的出险信息，如出险时间、地点等，在客户未提出理赔申请之前，主动与客户联系，提供理赔服务，不仅能提高理赔效率，还能为保险公司树立良好的企业形象。此外，保险公司还可以利用保险用户反馈的数据信息分析他们对保险服务的满意度，如调查出险报案电话接通的方便性、查勘人员到达事故现场的及时性、理赔手续的便利性、赔款到账的准时性、理赔结果满意性等内容，针对出现的特定问题制定相关解决办法，以改进服务质量。传统上，反馈信息的收集往往通过电话、客服、实际调查等方式，这些方式不仅收

集的数据量很有限，而且收集效率也不高。如今，随着互联网的普及使用，微博、微信等新兴社交软件的兴起，保险公司可以轻易获取大量的用户反馈信息，利用反馈信息去改善服务质量。

以上为我们的建议，谢谢！

六、模型的评价与推广

6.1 模型的优点

1、问题一中基于广义线性模型进行驾驶习惯风险分级并建立评分模型的主要方法和过程以及验证方法。UBI 车险定价的评分是其中关键的环节，而评分机制或模型中最重要的是各个指标变量的权重以及权重的赋予方法，本章主要通过广义线性模型解决了这个问题。

2、问题二以某财险公司的真实数据为基础，利用广义线性模型建立的评分模型对 UBI 车险的风险进行厘定分级。结果表明，所建立的驾驶习惯评分模型进而厘定得到 UBI 车险的保费是合理的也是可行的，即对提高续保率有一定的影响。

6.2 模型的缺点

1、驾驶习惯的指标变量的局限性。由于车联网技术和数据采集技术的限制，当前可供选择使用的驾驶习惯指标变量并不丰富，故而会导致某些对风险影响更为显著的指标无法被采用，而只能退而求其次，所以会影响模型的估计精度。

2、驾驶习惯评分模型的局限性。由于该模型对数据量有一定的要求，当数据量不足时会造成模型不收敛，或者即使收敛，然而由于某些风险水平的数据量很少导致该风险水平的精确度较差。

3、车联网数据的局限性。当前的车联网项目进展较晚，数据的积累尚有不足

6.3 模型的推广

随着我国经济的快速发展，人们的生活水平不断提高，汽车的保有量不断增长。然而现有的车险费率厘定模式仍然保守单调，难以为客户提供有差异化和个性化的车险保费，也因此难以为客户提供更加合理的车险保费。当前我国正面临车险费率的市场化改革，保费更加自主化对保险公司来说是一大机遇。另外，伴随着车联网技术和大数据技术的日益发展和成熟，也为实现 UBI 车险定价提供了技术上的可能。综上可见，利用车联网技术实现车险费率的进一步差异化和个性化是当前大的趋势。

七、参考文献

[1] 刘瑞刚. 广义线性模型在 UBI 车险费率厘定中的应用 [D]. 天津商业大学，2017 年。

[2] 牛睿尧. 基于广义线性模型的机动车险分类费率厘定方法研究 [D]. 吉林大学，2011 年。

[3] 张连增，段白鸽. 行驶里程数对车险净保费的影响研究 [J]. 保险研究，2012(6)：29-38，2012 年。

[4] 段冀阳，李志忠. 风险驾驶习惯影响因素的研究综述 [J]. 人类工效学，2013，19(2)：86-91，2013 年。

[5] 孟生旺. 广义线性模型在汽车保险定价的应用 [J]. 数理统计与管理，2007，26(1)：24-29，2007 年。

[6] 王亚娟. 基于广义线性模型的车险分类费率厘定研究 [D]. 山东大学，2013 年。

附录

问卷调查数据分析：SAS Code：

```
%macro consistn(where,part);
%let
var_list=dpt\sex_age\model\channel\ncd_renew\PL\seat1\accyr\combination
1;
data var(drop=va i);
va="&var_list.";
i=1;
do while (scan(va , i , "\")^="");
var=scan(va , i , "\");
call symputx(catt('var',i),var);
output;
i+1;
end;
i=i-1;
call symputx('var_num',i);
run;

proc summary data=temp&where. nway missing;
var EE ult_loss1 ep eg claimcount1;
output out=result(drop=_type_ _freq_)
sum=;
run;

data total;
set result;
format class $50.;
class= "Total";
format line_name $30. ;
line_name = "&line_list.";
call symputx("RP_TOTAL", ult_loss1/EE);
call symputx("LR_TOTAL", ult_loss1/ep);
call symputx("GR_TOTAL", ult_loss1/eg);
run;

%macro line(var_number);
data &line_list.;
set _null_;
```

```
run;

%do j=1%to&var_number.;
proc summary data=temp&where. nway missing;
class &&var&j;
var EE eg claimcount1 ult_loss1 ep;
output out=&&var&j (drop=_type_ _freq_)
sum=;
run;

data &&var&j;
format class $50.;
format class1 $50.;
set &&var&j;
class1=&&var&j;
class= "&&var&j";
drop &&var&j;
run;

data &line_list.;
set &line_list.&&var&j;
run;

%end;
proc sort data=&line_list. out=allfactor;
by class class1;
quit;

%mend line;
%line(&var_num.)
procgenmoddata=temp&where.ORDER=FREQ;
class&var_list_fre./desc;
model frequency = &var_list_fre. /dist=poisson link=log Type1Type3
SCALE=deviance ;
odsoutput parameterEstimates=frequency
 ModelFit=aic_fre
type1 = type1_fre
type3 = type3_fre;
weight EE;
run;

procgenmoddata=temp&where.ORDER=FREQ;
class&var_list_sev./desc;
model severity = &var_list_sev. /dist=Gamma link=log Type1Type3
```

```
SCALE=deviance ;
odsoutput parameterEstimates=severity
 ModelFit=aic_sev
type1 = type1_sev
type3 = type3_sev;
weight claimcount1 ;
run;

procexport data=frequency
dbms=excel2000 replace
OUTFILE="&path.\1117glm_result_&Vehicle_type..xls ";
sheet=" &line_list.fre_est&part. ";
run;

procexport data=severity
dbms=excel2000 replace
OUTFILE="&path.\1117glm_result_&Vehicle_type..xls ";
sheet=" &line_list.sev_est&part. ";
run;

data frequency(keep=class class1 frequency_opt frequency_low
frequency_high);
set frequency;
format class1 $50.;
format class $50.;
class=parameter;
class1=level1;
frequency_opt=exp(Estimate);
frequency_low=exp(LowerWaldCL);
frequency_high=exp(UpperWaldCL);
drop parameter level1;
if class="Intercept"then class="AA_Intercept";
run;

data severity(keep=class class1 severity_opt severity_low severity_high);
set severity;
format class1 $50.;
format class $50.;
class=parameter;
class1=level1;
severity_opt=exp(Estimate);
severity_low=exp(LowerWaldCL);
severity_high=exp(UpperWaldCL);
drop parameter level1;
```

```
if class="Intercept"then class="AA_Intercept";
run;

procsortdata=frequency;
by class class1;
quit;

procsortdata=severity;
by class class1;
quit;
```

GLM 模型——Python Code

```
import numpy as np
import scipy.stats as sts
import patsy as pt

from .utils import (check_types, check_commensurate, check_intercept,
                    check_offset, check_sample_weights, has_converged,
                    default_X_names, default_y_name)
from .families import Gaussian


class GLM:
    """A generalized linear model.

    GLMs are a generalization of the classical linear and logistic models
to
    other conditional distributions of response y.  A GLM is specified by
a
    *link function* G and a family of *conditional distributions* dist, with
    the model specification given by

        y | X ~ dist(theta = G(X * beta))

    Here beta are the parameters fit in the model, with X * beta a matrix
    multiplication just like in linear regression.  Above, theta is a
    *parameter* of the one parameter family of distributions dist.

    In this implementation, a specific GLM is specified with a *family* object
    of ExponentialFamily type, which contains the information about the
```

```
    conditional distribution of y, and its connection to X, needed to
construct
    the model. See the documentation for ExponentialFamily for
    details.

    The model is fit to data using the well known Fisher Scoring algorithm,
    which is a version of Newton's method where the hessian is replaced with
    its expectation with respect to the assumed distribution of Y.

    Parameters
    ----------
    family: ExponentialFamily object
        The exponential family used in the model.

    alpha: float, non-negative
        The ridge regularization strength. If non-zero, the loss function
        minimized is a penalized deviance, where the penalty is alpha *
        np.sum(model.coef_**2).

    Attributes
    ----------
    family: ExponentialFamily object
        The exponential family used in the model.

    alpha: float, non-negative
        The regularization strength.

    formula: str
        An (optional) formula specifying the model.  Used in the case that
X is
        passed as a DataFrame.  For documentation on model formulas, please
see
        the patsy library documentation.

    X_info: patsy.design_info.DesignInfo object.
        Contains information about the model formula used to process the
        training data frame into a design matrix.

    X_names: List[str]
        Names for the predictors.

    y_names: str
        Name for the target varaible.
```

```
    coef_: array, shape (n_features, )
        The fit parameter estimates.  None if the model has not yet been fit.

    deviance_: float
        The final deviance of the fit model on the training data.

    information_matrix_: array, shape (n_features, n_features)
        The estimated information matrix. This information matrix is
evaluated
        at the fit parameters.

    n: integer, positive
        The number of samples used to fit the model, or the sum of the sample
        weights.

    p: integer, positive
        The number of fit parameters in the model.

    Notes
    -----
    Instead of supplying a `fit_intercept` argument, we have instead assumed
    the programmer has included a column of ones as the *first* column X.
The
    fit method will throw an exception if this is not the case.
    """
    def __init__(self, family, alpha=0.0):
        self.family = family
        self.alpha = alpha
        self.formula = None
        self.X_info = None
        self.X_names = None
        self.y_name = None
        self.coef_ = None
        self.deviance_ = None
        self.n = None
        self.p = None
        self.information_matrix_ = None

    def fit(self, X, y=None, formula=None, *,
            X_names=None,
            y_name=None,
            **kwargs):
        """Fit the GLM model to training data.
```

```
        Fitting the model uses the well known Fisher scoring algorithm.

        Parameters
        ----------
        X: array, shape (n_samples, n_features) or pd.DataFrame
            Training data.

        y: array, shape (n_samples, )
            Target values.

        formula: str
            A formula specifying the model.  Used in the case that X is passed
            as a DataFrame.  For documentation on model formulas, please see
            the patsy library documentation.

        warm_start: array, shape (n_features, )
            Initial values to use for the parameter estimates in the
            optimization, useful when fitting an entire regulatization path
of
            models.  If not supplies, the initial intercept estimate will be
            the mean of the target array, and all other parameter estimates
            will be initialized to zero.

        offset: array, shape (n_samples, )
            Offsets for samples.  If provided, the model fit is
                E[Y|X] = family.inv_link(np.dot(X, coef) + offset)
            This is specially useful in models with exposures, as in Poisson
            regression.

        sample_weights: array, shape (n_sample, )
            Sample weights used in the deviance minimized by the model. If
            provided, each term in the deviance being minimized is multiplied
            by its corrosponding weight.

        max_iter: positive integer
            The maximum number of iterations for the fitting algorithm.

        tol: float, non-negative and less than one
            The convergence tolerance for the fitting algorithm.  The
relative
            change in deviance is compared to this tolerance to check for
            convergence.

        Returns
```

```
        -------
        self: GLM object
            The fit model.
        """
        check_types(X, y, formula)
        if formula:
            self.formula = formula
            y_array, X_array = pt.dmatrices(formula, X)
            self.X_info = X_array.design_info
            self.X_names = X_array.design_info.column_names
            self.y_name = y_array.design_info.term_names[0]
            y_array = y_array.squeeze()
            return self._fit(X_array, y_array, **kwargs)
        else:
            if X_names:
                self.X_names = X_names
            else:
                self.X_names = default_X_names(X)
            if y_name:
                self.y_name = y_names
            else:
                self.y_name = default_y_name()
            return self._fit(X, y, **kwargs)

    def _fit(self, X, y, *,
             warm_start=None,
             offset=None,
             sample_weights=None,
             max_iter=100,
             tol=0.1**5):
        """Fit the GLM model to some training data.

        This method expects X and y to be numpy arrays.
        """
        check_commensurate(X, y)
        check_intercept(X)
        if warm_start is None:
            initial_intercept = np.mean(y)
            warm_start = np.zeros(X.shape[1])
            warm_start[0] = initial_intercept
        coef = warm_start
        if offset is None:
            offset = np.zeros(X.shape[0])
        check_offset(y, offset)
```

```
        if sample_weights is None:
            sample_weights = np.ones(X.shape[0])
        check_sample_weights(y, sample_weights)

        family = self.family
        penalized_deviance = np.inf
        is_converged = False
        n_iter = 0
        while n_iter < max_iter and not is_converged:
            nu = np.dot(X, coef) + offset
            mu = family.inv_link(nu)
            dmu = family.d_inv_link(nu, mu)
            var = family.variance(mu)
            dbeta = self._compute_dbeta(X, y, mu, dmu, var, sample_weights)
            ddbeta = self._compute_ddbeta(X, dmu, var, sample_weights)
            if self._is_regularized():
                dbeta = dbeta + self._d_penalty(coef)
                ddbeta = self._dd_penalty(ddbeta, X)
            coef = coef - np.linalg.solve(ddbeta, dbeta)
            penalized_deviance_previous = penalized_deviance
            penalized_deviance = family.penalized_deviance(
                y, mu, self.alpha, coef)
            is_converged = has_converged(
                penalized_deviance, penalized_deviance_previous, tol)
            n_iter += 1

        self.coef_ = coef
        self.deviance_ = family.deviance(y, mu)
        self.n = np.sum(sample_weights)
        self.p = X.shape[1]
        self.information_matrix_ = self._compute_ddbeta(X, dmu, var,
sample_weights)
        return self

    def predict(self, X, offset=None):
        """Return predictions from a fit model.

        Predictions are computed using the inverse link function in the family
        used to fit the model:

            preds = family.inv_link(np.dot(X, self.coef_)

        Note that in the case of binary models, predict *does not* make class
        assignmnets, it returns a probability of class membership.
```

```
        Parameters
        ----------
        X: array, shape (n_samples, n_features)
            Data set.

        offset: array, shape (n_samples, )
            Offsets to add on the linear scale when making predictions.

        Returns
        -------
        preds: array, shape (n_samples, )
            Model predictions.
        """
        if not self._is_fit():
            raise ValueError(
                "Model is not fit, and cannot be used to make predictions.")
        if self.formula:
            rhs_formula = '+'.join(self.X_info.term_names[1:])
            X = pt.dmatrix(rhs_formula, X)
        if offset is None:
            return self.family.inv_link(np.dot(X, self.coef_))
        else:
            return self.family.inv_link(np.dot(X, self.coef_) + offset)

    def score(self, X, y):
        """Return the deviance of a fit model on a given dataset.

        Parameters
        ----------
        X: array, shape (n_samples, n_features)
            Data set.

        y: array, shape (n_samples, )
            Labels for X.

        Returns
        -------
        deviance: array, shape (n_samples, )
            Model deviance scored using supplied data and labels.
        """
        return self.family.deviance(y, self.predict(X))

    @property
```

```
    def dispersion_(self):
        """Return an estimate of the dispersion parameter phi."""
        if not self._is_fit():
            raise ValueError("Dispersion parameter can only be estimated for
a"
                             "fit model.")
        if self.family.has_dispersion:
            return self.deviance_ / (self.n - self.p)
        else:
            return np.ones(shape=self.deviance_.shape)

    @property
    def coef_covariance_matrix_(self):
        if not self._is_fit():
            raise ValueError("Parameter covariances can only be estimated
for a"
                             "fit model.")
        return self.dispersion_ * np.linalg.inv(self.information_matrix_)

    @property
    def coef_standard_error_(self):
        return np.sqrt(np.diag(self.coef_covariance_matrix_))

    @property
    def p_values_(self):
        """Return an array of p-values for the fit coefficients.  These
        p-values test the hypothesis that the given parameter is zero.

        Note: We use the asymptotic normal approximation to the p-values for
        all models.
        """
        if self.alpha != 0:
            raise ValueError("P-values are not available for "
                             "regularized models.")
        p_values = []
        null_dist = sts.norm(loc=0.0, scale=1.0)
        for coef, std_err in zip(self.coef_, self.coef_standard_error_):
            z = abs(coef) / std_err
            p_value = null_dist.cdf(-z) + (1 - null_dist.cdf(z))
            p_values.append(p_value)
        return np.asarray(p_values)

    def summary(self):
        """Print a summary of the model."""
```

```
        variable_names = self.X_names
        parameter_estimates = self.coef_
        standard_errors = self.coef_standard_error_
        header_string = "{:<10} {:>20} {:>15}".format(
            "Name", "Parameter Estimate", "Standard Error")
        print(f"{self.family.__class__.__name__} GLM Model Summary.")
        print('='*len(header_string))
        print(header_string)
        print('-'*len(header_string))
        format_string = "{:<20} {:>10.2f} {:>15.2f}"
        for name, est, se in zip(variable_names, parameter_estimates,
standard_errors):
            print(format_string.format(name, est, se))

    def clone(self):
        return self.__class__(self.family, self.alpha)

    def _is_fit(self):
        return self.coef_ is not None

    def _is_regularized(self):
        return self.alpha > 0.0

    def _compute_dbeta(self, X, y, mu, dmu, var, sample_weights):
        working_residuals = sample_weights * (y - mu) * (dmu / var)
        return - np.sum(X * working_residuals.reshape(-1, 1), axis=0)

    def _compute_ddbeta(self, X, dmu, var, sample_weights):
        working_h_weights = (sample_weights * dmu**2 / var).reshape(-1, 1)
        return np.dot(X.T, X * working_h_weights)

    def _d_penalty(self, coef):
        dbeta_penalty = coef.copy()
        dbeta_penalty[0] = 0.0
        return dbeta_penalty

    def _dd_penalty(self, ddbeta, X):
        diag_idxs = list(range(1, X.shape[1]))
        ddbeta[diag_idxs, diag_idxs] += self.alpha
        return ddbeta
```